Notes on Visual SLAM 14 lectures – Chapter 7

1 Feature method

A SLAM system can be divided into:

Visual odometry (VO): estimates the rough camera movement based on consecutive images and provides a good initial value for the backend

mainstream method, performs well due to stability and insensitivity to lighting and dynamic objects
Requirements for features:

- Repeatability: the same feature can be found in different images
- Distinctiveness: different features have different expressions
- Efficiency: in the same image, the number of feature points should be far smaller than the number of pixels
- Locality: the feature is only related to a small image area

1.0.1 Feature points

- key: 2D position of the feature point, orientation or size
- descriptors: usually a vector, describing the information of the pixels around the key point

1.0.2 SIFT

Scale-invariant feature transform

- cannot be done in real-time

1.1 ORB

Oriented FAST and rotated BRIEF

- real-time image feature extraction
- solves the problem that FAST detector does not have descriptors
- uses fast binary descriptor BRIEF

A. FAST corner point extraction: find the corner point in the image, main direction of feature points is calculated in ORB, making descriptors rotation-invariant

B. BRIEF descriptor: describe the surrounding image area where the feature points were extracted in the previous step

1.1.1 FAST key point

Main idea: if the pixel is very different from the neighboring pixels, it is more likely to be a corner point:

A. select pixel p in the image assuming its brightness as I_p

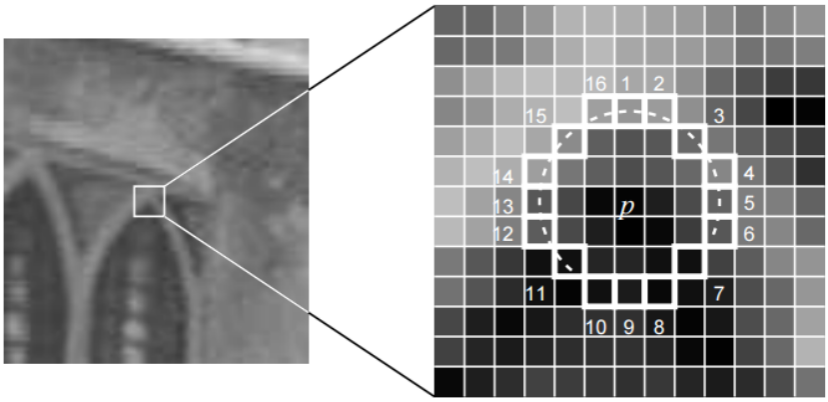
B. set a threshold T (for example 20% of I_p)

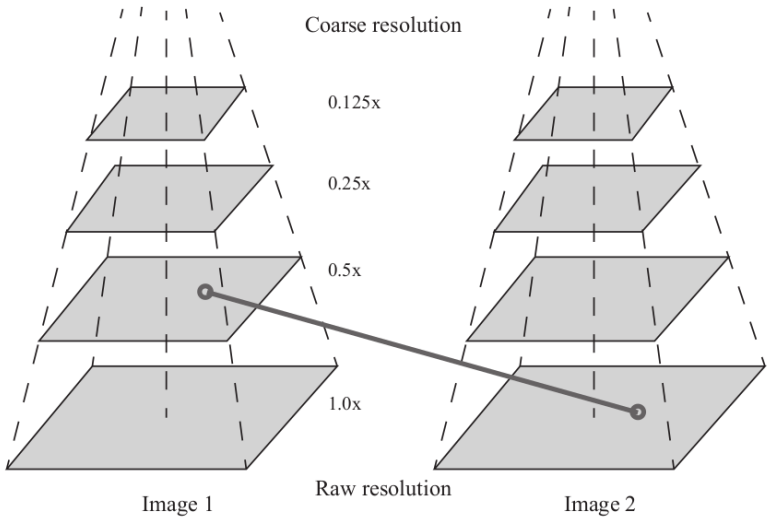
C. Take the pixel p as the center, and select the 16 pixels on a circle with a radius of 3

D. if there are consecutive N points on the selected circle whose brightness is greater than $I_p + T$ or less than $I_p - T$ then the central pixel p can be considered a feature point. N usually takes 12, which is FAST-12, there are also FAST-9 or FAST-11

E. iterate the above 4 steps on each pixel

Speedup: we can check the brightness of the 1, 5, 9, 13 pixels on the circle to quickly exclude many pixels that are not corner points. Only when 3 out of the 4 pixels are greater than $I_p + T$ or less than $I_p - T$, the current pixel may potentially be a corner point. Otherwise, we can proceed with the next pixel.





cons:

- suffer from lousy repeatability and uneven distribution
- do not include direction information
- scaling problem

Solution: 1. add the description of scale and rotation with image pyramid
2. compute the gray centroid of the image near the feature point:

■ in a small image block B, define the moment of image block as:

$$m_{pq} = \Sigma_{x,y \in B} x^p y^q I(x,y), p, q = \{0, 1\} \tag{1}$$

■ compute the centroid of the image block by the moment:

$$C = (\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}) \tag{2}$$

A. Connect the geometric center O and the centroid C of the image block to get a direction vector $\vec{OC}$, so the direction of the feature point can be defined as:

$$\theta = \arctan(m_{01}/m_{10}) \tag{3}$$

1.1.2 BRIEF descriptor

Binary descriptor that contains ones and zeros that encode the size relationship between two random pixels near the key point p and q, and if p is greater than q, take 1, otherwise take 0. Direction information from FAST can be used to calculate the Steer BRIEF feature after the rotation.

1.2 Feature Matching

Match features $x_t^m, m = 1, 2, ..., M$ in image I_t and features $x_{t+1}^n, n = 1, 2, ..., N$ in image I_{t+1}. Use hamming distance (number of different digits) as a metric for binary descriptors.
Fast library approximate nearest neighbor (FLANN) is more suitable in this case. We can also limit the range of brute-force method to achieve real time performance.

1.3 Calculate camera motion

- monocular – epipolar geometry
- binocular/RGB-D – ICP
- 3D-2D – PnP

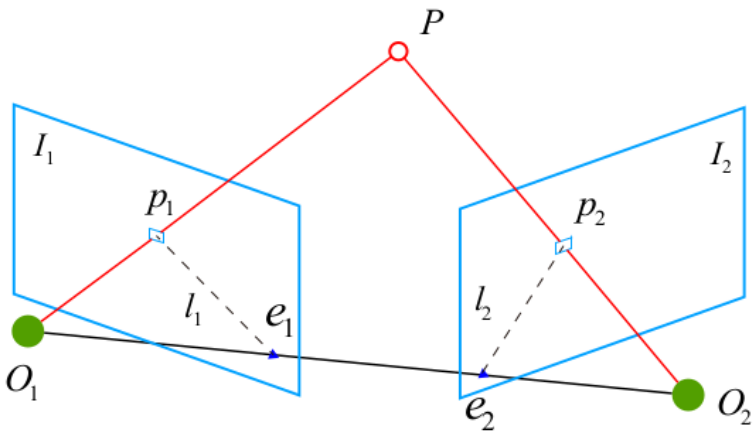
1.3.1 Epipolar constraints

Find the motion between two frames I_1, I_2. Define the motion from the first frame to the second frame as R, t and the centers of the two cameras as O_1, O_2. Suppose there is a feature point p_1 in image I_1 and a corresponding feature point p_2 in image I_2. The line $\vec{O_1 p_1}$ and $\vec{O_2 p_2}$ will intersect at the point P in the 3D space. The 3 points form a plane (epipolar plane). The intersection of the line of $O_1 O_2$ and the image plane is e_1, e_2 (epipoles). The line connection p_1 and e_1 and p_2 and e_2 are l_1, l_2 (epipolar lines).
Define the spatial position of P in the first frame to be:

$$P = [X, Y, Z]^T \tag{4}$$

we also have:

$$s_1 p_1 = KP, \quad s_2 P_2 = K(RP + t) \tag{5}$$



in which K is the intrinsics matrix amd R, t are the camera motions between two frames (transformation from the first frame to the second frame). Write them in homogeneous coordiantes (equal up to a scale):

$$p_1 \simeq KP, \quad P_2 \simeq K(RP + t) \tag{6}$$

Let $x_1 = K^{-1}p_1, \quad x_2 = K^{-1}p_2$ then we have:

$$x_2 \simeq Rx_1 + t \tag{7}$$

multiply both sides by $t^\wedge$:

$$t^\wedge x_2 \simeq t^\wedge R x_1 \tag{8}$$

left multiply x_2^T on both sides:

$$x_2^T t^\wedge x_2 \simeq x_2^T t^\wedge R x_1 \tag{9}$$

since the left side has $t^\wedge x_2$ is orthogonal to both $t^\wedge$ and x_2, its inner product with x_2 will get 0. We then get:

$$x_2^T t^\wedge R x_1 = 0 \tag{10}$$

substituting the x_1, x_2 we have:

$$p_2^T K^{-1} t^\wedge R K^{-1} p_1 = 0 \tag{11}$$

Define:

$$E = t^\wedge R, \quad F = K^{-T} E K^{-1}, \quad x_2^T E x_1 = p_2^T F p_1 = 0 \tag{12}$$

Therefore, the camera pose estimation can be summarized as:

A. find E or F based on the pixel positions of the matched points

B. find R, t based on E or F

simpler form E is often used in practice.

1.3.2 Essential matrix

A. Since epipolar constraint has 0 on the right side, E is invariant under different scales

B. according to $E = T^\wedge R$, it can be proved that the singular value of the essential matrix E must be in the form of $[\sigma, \sigma, 0]^T$ (internal properties of essential matrix)

C. $t^\wedge R$ should have 6 degrees of freedom, but due to the equivalence of scales, E actually has 5 degrees of freedom

According to the polar constraint, we have:

$$\begin{bmatrix} u_2 & v_2 & 1 \end{bmatrix} \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0 \tag{13}$$

we rewrite the matrix E in the vector form:

$$\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \end{bmatrix}^T \tag{14}$$

Then the epipolar constraint can be written as:

$$\begin{bmatrix} u_2u_1 & u_2v_1 & u_2 & v_2u_1 & v_2v_1 & v_2 & u_1 & v_1 & 1 \end{bmatrix} \cdot \mathbf{e} = 0 \tag{15}$$

stack all the points into one equation:

$$\begin{bmatrix} u_2^1u_1^1 & u_2^1v_1^1 & u_2^1 & v_2^1u_1^1 & v_2^1v_1^1 & v_2^1 & u_1^1 & v_1^1 & 1 \\ u_2^2u_1^2 & u_2^2v_1^2 & u_2^2 & v_2^2u_1^2 & v_2^2v_1^2 & v_2^2 & u_1^2 & v_1^2 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ u_2^8u_1^8 & u_2^8v_1^8 & u_2^8 & v_2^8u_1^8 & v_2^8v_1^8 & v_2^8 & u_1^8 & v_1^8 & 1 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \end{bmatrix} = 0 \tag{16}$$

which is eight-point algorithm. If the coefficient matrix is of full rank, then the null space dimension is 1, meaning that e forms a line (scale equivalence of e)

Perform SVD on E:

$$E = U\Sigma V^T \tag{17}$$

For any E, there are two possible t, R:

$$t_1^\wedge = UR_Z(\frac{\pi}{2})\Sigma U^T, \quad R_1 = UR_Z^T(\frac{\pi}{2})V^T$$
$$t_1^\wedge = UR_Z(-\frac{\pi}{2})\Sigma U^T, \quad R_1 = UR_Z^T(-\frac{\pi}{2})V^T$$

Since the negative of E is equivalent to E (does not change the epipolar constraint), there are in total 4 possible solutions, but only one solution will have positive depth for both cameras. Adjust the SVD results on E to make the sure $\Sigma = \mathbf{diag}(\sigma_1, \sigma_2, \sigma_3)$ and that $\sigma_1 \geq \sigma_2 \geq \sigma_3$:

$$E = U\mathbf{diag}(\frac{\sigma_1 + \sigma_2}{2}, \frac{\sigma_1 + sig_2}{2}, 0)V^T \tag{18}$$

which projects the essential matrix onto the manifold where E is located. We can also set the singular value matrix to be $\mathbf{diag}(1, 1, 0)$ due to E's scale equivalence.

1.3.3 Homography

mapping relationship between two planes – H For plane P:

$$\mathbf{n}^T\mathbf{P} + d = 0 \tag{19}$$

since $ax + by + cz + d = 0$ is the general equation for a plane. Rearrange it we have:

$$-\frac{\mathbf{n}^T\mathbf{P}}{d} = 1 \tag{20}$$

then we have:

$$\begin{aligned}\mathbf{p_2} &\simeq \mathbf{K}(\mathbf{RP}+t)\\ &\simeq \mathbf{K}(\mathbf{RP}+\mathbf{t}\cdot(-\frac{\mathbf{n}^T\mathbf{P}}{d}))\\ &\simeq \mathbf{K}(\mathbf{R}-\frac{\mathbf{tn}^T}{d})\mathbf{P}\\ &\simeq \mathbf{K}(\mathbf{R}-\frac{\mathbf{tn}^T}{d})\mathbf{K}^{-1}\mathbf{p}_1=\mathbf{Hp_1}\end{aligned}$$

so we get $\mathbf{p_2} \simeq \mathbf{Hp_1}$:

$$\begin{bmatrix}u_2\\v_2\\1\end{bmatrix} \simeq \begin{bmatrix}h_1 & h_2 & h_3\\h_4 & h_5 & h_6\\h_7 & h_8 & h_9\end{bmatrix}\begin{bmatrix}u_1\\v_1\\1\end{bmatrix} \tag{21}$$

due to its scale invariance, we can set h_9 to 1, and then:

$$\begin{aligned}u_2 &= \frac{h_1u_1+h_2v_1+h_3}{h_7u_1+h_8v_1+h_9}\\ v_2 &= \frac{h_4u_1+h_5v_1+h_6}{h_7u_1+h_8v_1+h_9}\end{aligned}$$

rearrange we have:

$$\begin{aligned}h_1u_1+h_2v_1+h_3-h_7u_1u_2-h_8v_1u_2=u_2\\ h_4u_1+h_5v_1+h_6-h_7u_1v_2-h_8v_1v_2=v_2\end{aligned}$$

Every pair of matching points can construct two constraints, so the homography matrix with 8 degrees of freedom can be estimated by 4 pairs of matched features:

$$\begin{bmatrix}u_1^1 & v_1^1 & 1 & 0 & 0 & 0 & -u_1^1u_2^1 & -v_1^1u_2^1\\ 0 & 0 & 0 & u_1^1 & v_1^1 & 1 & -u_1^1v_2^1 & -v_1^1v_2^1\\ u_1^2 & v_1^2 & 1 & 0 & 0 & 0 & -u_1^2u_2^2 & -v_1^2u_2^2\\ 0 & 0 & 0 & u_1^2 & v_1^2 & 1 & -u_1^2v_2^2 & -v_1^2v_2^2\\ u_1^3 & v_1^3 & 1 & 0 & 0 & 0 & -u_1^3u_2^3 & -v_1^3u_2^3\\ 0 & 0 & 0 & u_1^3 & v_1^3 & 1 & -u_1^3v_2^3 & -v_1^3v_2^3\\ u_1^4 & v_1^4 & 1 & 0 & 0 & 0 & -u_1^4u_2^4 & -v_1^4u_2^4\\ 0 & 0 & 0 & u_1^4 & v_1^4 & 1 & -u_1^4v_2^4 & -v_1^4v_2^4\end{bmatrix}\begin{bmatrix}h_1\\h_2\\h_3\\h_4\\h_5\\h_6\\h_7\\h_8\end{bmatrix} = \begin{bmatrix}u_2^1\\v_2^1\\u_2^2\\v_2^2\\u_2^3\\v_2^3\\u_2^4\\v_2^4\end{bmatrix} \tag{22}$$

which is DLT (direct linear transform) method to solve H. When points are coplanar, it causes degeneration which reduces the degree of freedom of the fundamental matrix. We usually estimate the fundamental matrix F and homography matrix H simultaneously and then choose one with a smaller reprojection error as the final result.

H requires the assumption that the points need to be on the same plane. Since E is scale equivalent, t, R are also scale equivalent. Since R is restricted by $\mathbf{SO}(3)$ constraint, if t is multiplied by any non-zero constant, the decomposition is still valid. Therefore, we normalize t to make its length equal to 1.

When there are more than 8 points, we can rewrite the overdetermined equation:

$$\mathbf{Ae}=\mathbf{0} \tag{23}$$

which can be solved by minimizing in a quadratic form:

$$\min_e ||\mathbf{Ae}||_2^2 = \min_e \mathbf{e}^T\mathbf{A}^T\mathbf{Ae} \tag{24}$$

which can be solved by minimizing least squares. However, RANSAC (random sample consensus) is normally used since it can handel potential mismatches.

1.3.4 Triangulation

let x_1, x_2 be the normalized coordinates of two feature points:

$$s_x x_2 = s_1 R x_1 + t \tag{25}$$

to get s_1, we multiply both sides by $x_2^\wedge$ and get:

$$s_2 x_2^\wedge x_2 = 0 = s_1 x_2^\wedge R x_1 + x_2^\wedge t \tag{26}$$

which can directly give us s_1, thus also giving us s_2. Due to noise, a least square solution is more used.

When the translation is small, the triangulation will have a larger depth uncertainty. Therefore, it is important to:

A. improve the accuracy of feature extraction (computationally more costly)

B. increase the amount of translation (will make feature extraction and matching more difficult)

Delayed triangulation: wait for the feature points to be tracked for a few frames before using triangulation.

1.4 PnP Matching

Perspective-n-Point:

- solve 3D to 2D motion estimation
- does not require epipolar constraints
- variants: P3P, EPnP (efficient PnP), UPnP

1.4.1 DLT method

consider a 3D spatial point P, its homogeneous coordinates are $P = (X, Y, Z)^T$ in image I_1, and is projected to the feature point $x_1 = (u_1, v_1, 1)^T$ (normalized homogeneous coordinates). The pose of camera R, t is unknown. We define 3×4 augmented matrix $[R|t]$, encoding rotation and translation information:

$$s \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{27}$$

eliminate s with the last row and get:

$$u_1 = \frac{t_1 X + t_2 Y + t_3 Z + t_4}{t_9 X + t_{10} Y + t_{11} Z + t_{12}}, v_1 = \frac{t_5 X + t_6 Y + t_7 Z + t_8}{t_9 X + t_{10} Y + t_{11} Z + t_{12}} \tag{28}$$

to simplify we define T as a row vector:

$$t_1 = (t_1, t_2, t_3, t_4)^T, t_2 = (t_5, t_6, t_7, t_8)^T, t_3 = (t_9, t_{10}, t_{11}, t_{12})^T \tag{29}$$

define T as a row vector:

$$t_1^T P - t_3^T P u_1 = 0$$
$$t_2^T P - t_3^T P v_1 = 0$$

Each feature point provides two linear constraints on t, assume there are a total of N feature points:

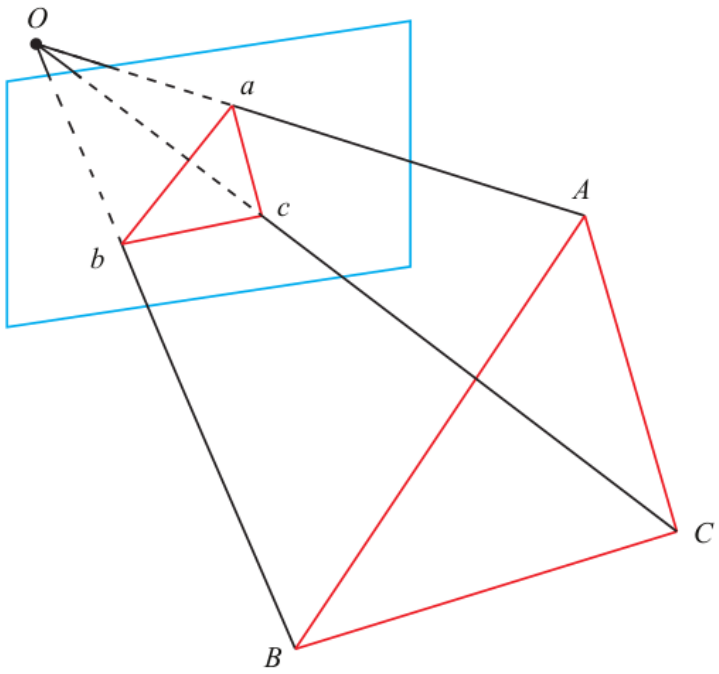
$$\begin{bmatrix} P_1^T & 0 & -u_1 P_1^T \\ 0 & P_1^T & -v_1 P_1^T \\ \vdots & \vdots & \vdots \\ P_N^T & 0 & -u_N P_N^T \\ 0 & P_N^T & -v_N P_N^T \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = 0 \tag{30}$$

since T has a total dimension of 12, the linear solution of the matrix T can be achieved by at least six pairs of matching points. To make sure that R satisfied $\mathbf{SO}(3)$ contraint, we perform QR decomposition on R:

$$R \longleftarrow (RR^T)^{-\frac{1}{2}}R \tag{31}$$

1.4.2 P3P

only uses 3 pairs of matching points, and an additional pair for verification (denoted as D - d) Once the coor-



dinates of the 3D point in the camera coordinate system can be calculated, we get the 3D-3D corresponding point and convert the PnP problem to the ICP problem. Using the law of similar triangles, we have:

$$OA^2 + OB^2 - 2OA \cdot OB \cdot \cos\langle a, b\rangle = AB^2 \tag{32}$$

The other two triangles have similar properties, so we get:

$$OB^2 + OC^2 - 2OB \cdot OC \cdot \cos\langle b, c\rangle = BC^2$$
$$OA^2 + OC^2 - 2OA \cdot OC \cdot \cos\langle a, c\rangle = AC^2$$

Denote $x = OA/OC, y = OB/OC$, and divide all the equations OC^2:

$$x^2 + y^2 - 2xy\cos\langle a, b\rangle = AB^2/OC^2$$
$$y^2 + 1^2 - 2y\cos\langle b, c\rangle = BC^2/OC^2$$
$$x^2 + 1^2 - 2x\cos\langle a, c\rangle = AC^2/OC^2$$

Let $v = AB^2/OC^2, uv = BC^2/OC^2, wv = AC^2/OC^2$, then:

$$x^2 + y^2 - 2xy\cos\langle a, b\rangle - v = 0$$
$$y^2 + 1^2 - 2y\cos\langle b, c\rangle - uv = 0$$
$$x^2 + 1^2 - 2x\cos\langle a, c\rangle - wv = 0$$

Move v in the first equation to the right side, and combine it with the other two equations, we have:

$$(1-u)y^2 - ux^2 - \cos\langle b,c\rangle y + 2uxy\cos\langle a,b\rangle + 1 = 0$$
$$(1-w)x^2 - wy^2 - \cos\langle a,c\rangle x + 2wxy\cos\langle a,b\rangle + 1 = 0$$

analytically solve x and y is complicated and requires Wu's elimination. cons:

A. only includes the information of 3 points, when given matched points are more than 3, it is difficult to use more information

B. if the 3D point or 2D point is affected by noise or a mismatch, the algorithm goes into trouble

common practice: estimate camera pose using P3P/EPnP and then construct a least-squares optimization problem to adjust the estimated values (bundle adjustment).

1.4.3 Solve PnP

suppose coordinates of a point are $\mathbf{P}_i = [X_i, Y_i, Z_i]^T$, and their projected pixel coordinates are $\mathbf{u}_i = [u_i, v_i]^T$:

$$s_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{KT} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix} \tag{33}$$

or $s_i\mathbf{u_i} = \mathbf{KTP}_i$ which includes a conversion from homogeneous coordinates to non-homogeneous coordinates. We then minimize the residue:

$$\mathbf{T}^* = \arg\min_T \frac{1}{2}\sum_{i=1}^{2} ||\mathbf{u}_i - \frac{1}{s_i}\mathbf{KTP}_i||_2^2 \tag{34}$$

which is a reprojection error. suppose $\mathbf{x}$ is the camera pose(6-d) and $\mathbf{e}$ is the pixel coordinate error (2-d), and $\mathbf{J}^T$ is a matrix of 2x6, we have:

$$\mathbf{e}(\mathbf{x}+\Delta\mathbf{x}) \approx \mathbf{e}(\mathbf{x}) + \mathbf{J}^T\Delta\mathbf{x} \tag{35}$$

Define the coordinates of the space point in the camera frame as $\mathbf{P'}$, take out the first 3 dimensions:

$$\mathbf{P'} = (\mathbf{TP})_{1:3} = [X', Y', Z']^T \tag{36}$$

the camera projection model with respect to $\mathbf{P'}$ is:

$$s\mathbf{u} = \mathbf{KP'} \tag{37}$$

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \tag{38}$$

use the third row to eliminate s:

$$u = f_x\frac{X'}{Z'} + c_x, v = f_y\frac{Y'}{Z'} + c_y \tag{39}$$

left multiply $\mathbf{T}$ by a disturbance quantity $\delta\xi$, then consider the derivative of the change of e with respect to the disturbance quantity:

$$\frac{\partial\mathbf{e}}{\partial\delta\xi} = \lim_{\delta\xi\longrightarrow 0}\frac{\mathbf{e}(\delta\xi\bigoplus\xi) - \mathbf{e}(\xi)}{\delta\xi} = \frac{\partial\mathbf{e}}{\partial\mathbf{P'}}\frac{\partial\mathbf{P'}}{\partial\delta\xi} \tag{40}$$

in which $\bigoplus$ denotes the left multiplication in Lie algebra:

$$\frac{\partial\mathbf{e}}{\partial\mathbf{P'}} = -\begin{bmatrix} \frac{\partial u}{\partial X'} & \frac{\partial u}{\partial Y'} & \frac{\partial u}{\partial Z'} \\ \frac{\partial v}{\partial X'} & \frac{\partial v}{\partial Y'} & \frac{\partial v}{\partial Z'} \end{bmatrix} = -\begin{bmatrix} \frac{f_x}{Z'} & 0 & \frac{-f_xX'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & \frac{-f_yY'}{Z'^2} \end{bmatrix} \tag{41}$$

The second term is the derivative of the transformed point with respect to the Lie algebra:

$$\frac{\partial \mathbf{TP}}{\partial \delta \xi} = (\mathbf{TP})^{\odot} = \begin{bmatrix} \mathbf{I} & -\mathbf{P'}^{\wedge} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \tag{42}$$

in the definition of $\mathbf{P'}$, we take out the first 3 dimensions:

$$\frac{\partial \mathbf{P'}}{\partial \delta \xi} = [\mathbf{I}, -\mathbf{P'}^{\wedge}] \tag{43}$$

as a result, we get:

$$\frac{\partial \mathbf{e}}{\partial \delta \xi} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X'^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix} \tag{44}$$

which optimizes the pose. To optimize the spatial position of the feature points:

$$\frac{\partial \mathbf{e}}{\partial \mathbf{P}} = \frac{\partial \mathbf{e}}{\partial \mathbf{P'}} \frac{\partial \mathbf{P'}}{\partial \mathbf{P}} \tag{45}$$

the second term is defined as $\mathbf{P'} = (\mathbf{TP})_{1:3} = \mathbf{RP} + \mathbf{t}$ so:

$$\frac{\partial \mathbf{e}}{\partial \mathbf{P}} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & \frac{-f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & \frac{-f_y Y'}{Z'^2} \end{bmatrix} \mathbf{R} \tag{46}$$

1.4.4 Optimization by G2O

A. Node: the pose fo the second camera $\mathbf{T} \in SE(3)$

B. Edge: the projection of each 3D point in the second camera, described by the observation equation: $\mathbf{z}_j = h(\mathbf{T}, \mathbf{P_j})$

fix pose of the first camera to be identity, so it can be excluded from the optimization variables.

1.5 3D-3D Iterative Closest Point(ICP)

$$\mathbf{P} = \{p_1, ..., p_n\}, \mathbf{P'} = \{p'_1, ..., p'_n\} \tag{47}$$

find an Euclidean transformation $\mathbf{R}, \mathbf{t}$:

$$\forall i, \mathbf{p}_i = \mathbf{R}\mathbf{P'}_i + \mathbf{t} \tag{48}$$

Two ways:

A. Linear algebra (SVD)

B. non-linear optimization (bundle adjustment)

construct a least-square problem to find the $\mathbf{R}, \mathbf{t}$:

$$\min_{\mathbf{R},\mathbf{t}} \frac{1}{2} \sum_{i=1}^{n} ||(\mathbf{p}_i - (\mathbf{R}\mathbf{p'}_i + \mathbf{t}))||_2^2 \tag{49}$$

define centroids of two sets of points:

$$\mathbf{p} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{p}_i), \quad \mathbf{p'} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{p'}_i) \tag{50}$$

we then have:

$$\begin{aligned}\frac{1}{2}\sum_{i=1}^{n}||(\mathbf{p}_i-(\mathbf{R}\mathbf{p'}_i+\mathbf{t}))||^2 &= \frac{1}{2}\sum_{i=1}^{n}||\mathbf{p}_i-\mathbf{R}\mathbf{p'}_i-\mathbf{t}-\mathbf{p}+\mathbf{R}\mathbf{p'}+\mathbf{p}-\mathbf{R}\mathbf{p'}||^2\\ &= \frac{1}{2}\sum_{i=1}^{n}||(\mathbf{p}_i-\mathbf{p}-\mathbf{R}(\mathbf{p'_i}-\mathbf{p'}))+(\mathbf{p}-\mathbf{R}\mathbf{p'}-\mathbf{t})||^2\\ &= \frac{1}{2}\sum_{i=1}^{n}(||\mathbf{p_i}-\mathbf{p}-\mathbf{R}(\mathbf{p'_i}-\mathbf{p'})||^2+\\ &||p-Rp'-t||^2+2(\mathbf{p_i}-\mathbf{p}-\mathbf{R}(\mathbf{p'_i}-\mathbf{p'}))^T(\mathbf{p}-\mathbf{R}\mathbf{p'}-\mathbf{t})||^2)\end{aligned}$$

since the third term is 0 after the summation, the optimization objective function can be simplified to:

$$\min_{\mathbf{R},\mathbf{t}} J=\frac{1}{2}\sum_{i=1}^{n}||\mathbf{p_i}-\mathbf{p}-\mathbf{R}(\mathbf{p'_i}-\mathbf{p'})||^2+||\mathbf{p}-\mathbf{R}\mathbf{p'}-\mathbf{t}||^2 \tag{51}$$

The first term is only related to the rotation matrix $\mathbf{R}$, while the second is related to both $\mathbf{R},\mathbf{t}$.

A. calculate centroids of the two groups of points $\mathbf{p},\mathbf{p'}$, and then calculate the de-centroid coordinates of each point:

$$\mathbf{q}_i=\mathbf{p}_i-\mathbf{p},\mathbf{q'}_i=\mathbf{p'}_i-\mathbf{p'} \tag{52}$$

B. the rotation matrix is calculated according to the following optimization problem:

$$\mathbf{R}^*=\arg\min_{\mathbf{R}}\frac{1}{2}\sum_{i=1}^{n}||\mathbf{q_i}-\mathbf{R}\mathbf{q'_i}||^2 \tag{53}$$

C. calculate $\mathbf{t}$ according to $\mathbf{R}$ in step 2:

$$\mathbf{t}^*=\mathbf{p}-\mathbf{R}\mathbf{p'} \tag{54}$$

expand the error term about $\mathbf{R}$:

$$\frac{1}{2}\sum_{i=1}^{n}||\mathbf{q_i}-\mathbf{R}\mathbf{q'_i}||^2=\frac{1}{2}\sum_{i=1}^{n}\mathbf{q_i^T}\mathbf{q_i}+\mathbf{q_i^{'T}}\mathbf{R^T}\mathbf{R}\mathbf{q'_i}-2\mathbf{q_i^T}\mathbf{R}\mathbf{q'_i} \tag{55}$$

the first term is not relevant to $\mathbf{R}$, the second term can also be ignored since $\mathbf{R^TR}=\mathbf{I}$, then:

$$\sum_{i=1}^{n}-\mathbf{q_i^T}\mathbf{R}\mathbf{q'_i}=\sum_{i=1}^{n}-\mathrm{tr}(\mathbf{R}\mathbf{q'_i}\mathbf{q_i^T})=-\mathrm{tr}(\mathbf{R}\sum_{i=1}^{n}\mathbf{q'_i}\mathbf{q_i^T}) \tag{56}$$

define $\mathbf{W}=\sum_{\mathbf{i=1}}^{\mathbf{n}}\mathbf{q_i}\mathbf{q_i^{'T}}$, perform SVD, we have:

$$\mathbf{W}=\mathbf{U}\mathbf{\Sigma}\mathbf{V^T} \tag{57}$$

Lemma: for arbitrary positive definite $\mathbf{AA^T}$, $\mathrm{tr}(\mathbf{AA^T})\geq\mathrm{tr}(\mathbf{BAA^T})$ for any orthogonal $\mathbf{B}$ proof:

$$\mathrm{tr}(BAA^T)=\mathrm{tr}(A^TBA)=\sum a_i^T(Ba_i) \tag{58}$$

in which a_i is the column vector of A, according to cauchy-schwartz inquality:

$$a_i^T(Ba_i)\leq\sqrt{(a_i^Ta_i)(a_i^TB^TBa_i)}=a_i^Ta_i \tag{59}$$

so $\mathrm{tr}(BAA^T)=\sum a_i^T(Ba_i)\leq\sum a_i^Ta_i=\mathrm{tr}(AA^T)$ in that case, we just have to find that positive semi-definite matrix A that could take our objective $\mathrm{tr}(\mathbf{R}\sum_{i=1}^{n}\mathbf{q'_i}\mathbf{q_i^T})$ to maximum, we let

$$\mathbf{R}=\mathbf{V}\mathbf{U^T} \tag{60}$$

then we will have:

$$\mathrm{tr}(\mathbf{RW})=\mathrm{tr}\mathbf{V}\mathbf{U^T}\mathbf{U}\mathbf{\Sigma}\mathbf{V^T}=\mathrm{tr}(\mathbf{V}\mathbf{U^T}\mathbf{U}\mathbf{\Sigma^{\frac{1}{2}}}\mathbf{\Sigma^{\frac{1}{2}}}\mathbf{V^T})=\mathrm{tr}(\mathbf{V}\mathbf{\Sigma^{\frac{1}{2}}}\mathbf{\Sigma^{\frac{1}{2}}}\mathbf{V^T})=\mathrm{tr}(\mathbf{V}\mathbf{\Sigma^{\frac{1}{2}}}(\mathbf{V}\mathbf{\Sigma^{\frac{1}{2}}})^\mathbf{T}) \tag{61}$$

which is maximum according to the lemma, since any orthogonal $\mathbf{R}$ could be left-multiply and it would not create a larger result. If the determinant of $\mathbf{R}$ is negative, then its negative is taken as the optimal value.

1.5.1 Solve ICP with non-linear optimization

when expressing the pose in Lie algebra, we can write the objective function as:

$$\min_{\xi} = \frac{1}{2}\sum_{i=1}^{n} ||(\mathbf{p}_i) - \exp(\xi^{\wedge})\mathbf{p}_\mathbf{i}'||_2^2 \tag{62}$$

the derivative of a single error term with respect to the pose can be written as with the Lie algebra perturbation model:

$$\frac{\partial \mathbf{e}}{\partial \delta\xi} = -(\exp(\xi^{\wedge})\mathbf{p}_\mathbf{i}')^{\odot} \tag{63}$$

for feature points with known depths, model their 3D-3D errors, for feature points with unknown depths, model 3D-2D reprojection errors.