Notes on Visual SLAM 14 lectures – Chapter 9

1 State estimation from probabilistic perspective

SLAM process: motion and observation equations:

$$\begin{cases} x_k = f(x_{k-1}, u_k) + w_k \\ z_{k,j} = h(y_j, x_k) + v_{k,j}, \quad k = 1, ..., N, j = 1, ..., M \end{cases} \tag{1}$$

A. in the observation equation, only when x_k sees y_j will generate a real observation equation, but due to a large number of visual SLAM feature points, the number of observation equations will be much larger

B. we may not have a device to measure the motion

(a) assume there is no motion

(b) assume the camera does not move

(c) assume the camera is moving at a constant speed

in the absence of motion equations, this is similar to SfM (structure from motion) denote x_k as all the unknowns at the moment of k (camera pose and all landmarks):

$$x_k \triangleq \{x_k, y_1, ..., y_m\} \tag{2}$$

and we would have the re-written equations:

$$\begin{cases} x_k = f(x_{k-1}, u_k) + w_k \\ z_{k,j} = h(x_k) + v_k, \quad k = 1, ..., N \end{cases} \tag{3}$$

we hope to use the data from 0 to k to estimate the current state distribution:

$$p(x_k | x_0, u_{1:k}, z_{1:k}) \tag{4}$$

according to Bayes' rule, and the fact that $P(A|BC) = \frac{P(A|C)P(B|C)}{P(A)}$, denote $A = x_k, B = z_k, C = x_0, u_{1:k}, z_{1:k-1}$:

$$\begin{aligned}
P(x_k|x_0, u_{1:k}, z_{1:k}) &= P(x_k|x_0, u_{1:k}, z_k, z_{1:k-1}) \\
&= \frac{P(z_k|x_k, x_0, u_{1:k}, z_{1:k-1})P(x_k, x_0, u_{1:k}, z_{1:k-1})}{P(x_0, u_{1:k}, z_k, z_{1:k-1})} \\
&= \frac{P(z_k|x_k)P(x_k|x_0, u_{1:k}, z_{1:k-1})P(x_0, u_{1:k}, z_{1:k-1}))}{P(x_0, u_{1:k}, z_k, z_{1:k-1})} \\
&= \frac{P(z_k|x_k)P(x_k|x_0, u_{1:k}, z_{1:k-1})}{\frac{P(x_0, u_{1:k}, z_k, z_{1:k-1})}{P(x_0, u_{1:k}, z_{1:k-1})}} \\
&= \frac{P(z_k|x_k)P(x_k|x_0, u_{1:k}, z_{1:k-1})}{P(z_k|x_0, u_{1:k}, z_{1:k-1})} \\
&= \frac{P(z_k|x_k)P(x_k|x_0, u_{1:k}, z_{1:k-1})}{P(z_k)} \quad \text{since } z_k \text{ only depends on } x_k \\
&\propto P(z_k|x_k, x_0, u_{1:k}, z_{1:k-1})P(x_k|x_0, u_{1:k}, z_{1:k-1})
\end{aligned}$$

the first term is likelihood, and the second term is prior. To estimate prior, we can expand it:

$$P(x_k|x_0, u_{1:k}, z_{1:k-1}) = \int P(x_k|x_{k-1}, x_0, u_{1:k}, z_{1:k-1})P(x_{k-1}|x_0, u_{1:k}, z_{1:k-1})dx_{k-1} \tag{5}$$

1.1 Linear systems and Kalman filter

The first part of equation 5 can be simplified as:

$$P(x_k|x_{k-1}, x_0, u_{1:k}, z_{1:k-1}) = P(x_k|x_{k-1}, u_k) \tag{6}$$

where we omit the states earlier than $k-1$ since they are not related to the k-th state. The second part can be simplified as:

$$P(x_{k-1}|x_0, u_{1:k}, z_{1:k-1}) = P(x_{k-1}|x_0, u_{1:k-1}, z_{1:k-1}) \tag{7}$$

Linear Gaussian system:

$$\begin{cases} x_k = A_k x_{k-1} + u_k + w_k \\ z_k = C_k x_k + v_k, \quad k = 1, ..., N \end{cases} \tag{8}$$

and we assume that the states and noises are all Gaussian, so $w_k \sim N(0, R)$ and $v_k \sim N(0, Q)$ Using Markov property, suppose we know the posterior state estimation at time k - 1 $\hat{x}_{k-1}$ and its covariance $\hat{P}_{k-1}$, now we want to estimate the posterior distribution of x_k based on input and observation data at time k. We use $\check{\mathrm{x}}_k$ to denote the prior distribution and $\hat{x}_k$ to denote its posterior distribution. Prior of x_k through the equation of motion:

$$P(x_k|x_0, u_{1:k}, z_{1:k-1}) = N(A_k\hat{x}_{k-1} + u_k, A_k\hat{P}_{k-1}A_k^T + R) \tag{9}$$

The second part can be quickly proved:

$$\begin{aligned}
\Sigma_k &= E((x_k - E(x_k))(x_k - E(x_k))^T) \\
&= E((A_k x_{k-1} + u_k - A_k E(x_{k-1}) - u_k)(A_k x_{k-1} + u_k - A_k E(x_{k-1}) - u_k)^T) \\
&= E((A_k(x_{k-1} - E(x_{k-1})))(A_k(x_{k-1} - E(x_{k-1})))^T) \\
&= E(A_k(x_{k-1} - E(x_{k-1}))(x_{k-1} - E(x_{k-1}))^T A_k^T) \\
&= A_k E((x_{k-1} - E(x_{k-1}))(x_{k-1} - E(x_{k-1}))^T) A_k^T \\
&= A_k \Sigma_{k-1} A_k^T
\end{aligned}$$

this step is called prediction:

$$\check{\mathrm{x}}_k = A_k\check{\mathrm{x}}_{k-1} + u_k, \quad \check{\mathrm{P}}_k = A_k\hat{P}_{k-1}A_k^T + R \tag{10}$$

from observation equation, we can calculate what kind of observation data should be generated in a certain state:

$$P(z_k|x_k) = N(C_k x_k, Q) \tag{11}$$

we want to get the posterior $P(x_k|z_k)$, we have the result of $x_k \sim N(\hat{x}_k, \hat{P}_k)$:

$$N(\hat{x}_k, \hat{P}_k) = \eta N(C_k x_k, Q) \cdot N(x_k, P_k) \tag{12}$$

in which η is the normalization factor that makes the integral of the distribution equal to one. expand the exponential part as:

$$(x_k - \hat{x}_k)^T\hat{P}_k^{-1}(x_k - \hat{x}_k) = (z_k - C_k x_k)^T Q^{-1}(z_k - C_k x_k) + (x_k - \check{\mathrm{x}}_k)^T\check{\mathrm{P}}_k^{-1}(x_k - \check{\mathrm{x}}_k) \tag{13}$$

In order to compute $\hat{x}_k\hat{P}_k$ on the left side, we expand the quadratics and compare their first-order and second-order coefficients of x_k, we have:

$$\hat{P}_k^{-1} = C_k^T Q^{-1} C_k + \check{\mathrm{P}}_k^{-1} \tag{14}$$

which gives the relationship of the covariance matrix. Define an intermediate variable for convenience in the following derivation:

$$K = \hat{P}_k C_k^T Q^{-1} \tag{15}$$

multiply $\hat{P}_k$ on both sides of equation 14:

$$I = \hat{P}_k C_k^T C_k + \hat{P}_k P^{-1} = K C_k + \hat{P}_k P_k^{-1} \tag{16}$$

then we have:

$$\hat{P}_k = (I - KC_k)\check{\mathrm{P}}_k \tag{17}$$

compared the first-order coefficients in equation 13:

$$-2\hat{x}_k^T \hat{P}_k^{-1} x_k = -2z_k^T Q^{-1} x_k - 2\check{\mathrm{x}}_k^T \check{\mathrm{P}}_k^{-1} x_k \tag{18}$$

take the coefficients and transpose them:

$$\hat{P}_k^{-1} \hat{x}_k = C_k^T Q^{-1} z_k + \check{\mathrm{P}}_k^{-1} \check{\mathrm{x}}_k \tag{19}$$

multiply $\hat{P}_k$ on both sides:

$$\begin{aligned}
\hat{x}_k &= \hat{P}_k C_k^T Q^{-1} z_k + \hat{P}_k \check{\mathrm{P}}_k^{-1} \check{\mathrm{x}}_k \\
&= K z_k + (I - K C_K)\check{\mathrm{x}}_k \\
&= \check{\mathrm{x}}_k + K(z_k - C_k \check{\mathrm{x}}_k)
\end{aligned}$$

the general steps are:

A. Predict:

$$\check{\mathrm{x}}_k = A_k \hat{x}_{k-1} + u_k, \quad \check{\mathrm{P}}_k = A_k \hat{P}_{k-1} A_k^T + R \tag{20}$$

B. Update: Calculate K which is the Kalman gain. Following the definition of K, we have:

$$\begin{aligned}
K Q_k &= \hat{P}_k C_k^T \\
&= (\check{\mathrm{P}}_k - K C_k \check{\mathrm{P}}_k) C_k^T \\
K(Q_k + C_k \check{\mathrm{P}}_k C_k^T) &= \check{\mathrm{P}}_k C_k^T \\
K &= \check{\mathrm{P}}_k C_k^T (C_k \check{\mathrm{P}}_k C_k^T + Q_k)^{-1}
\end{aligned}$$

and calculate the posterior:

$$\hat{P}_k = (I - KC_k)\check{\mathrm{P}}_k$$

1.2 Nonlinear systems and EKF

First-order Taylor expansion of the motion equation and the observation equation near a working point. Let the mean and covariance matrix at time $k-1$ be $\hat{x}_{k-1}$ and $\hat{P}_{k-1}$. At the moment k, we do the linearization:

$$x_k \approx f(\hat{x}_{k-1}, u_k) + \left.\frac{\partial f}{\partial x_{k-1}}\right|_{\hat{x}_{k-1}} (x_{k-1} - \hat{x}_{k-1}) + w_k \tag{21}$$

in which $F = \left.\frac{\partial f}{\partial x_{k-1}}\right|_{\hat{x}_{k-1}}$ and for the observation model:

$$z_k \approx h(x_k) + \left.\frac{\partial h}{\partial x_k}\right|_{\hat{x}_{k-1}} (x_k - \hat{x}_k) + n_k \tag{22}$$

in which $H = \frac{\partial h}{\partial x_k}\Big|_{\hat{x}_{k-1}}$ Then the prediction part becomes:

$$P(x_k|x_0, u_{1:k}, z_{1:k-1}) = N(f(\hat{x}_{k-1}, u_k), F\hat{P}_{k-1}F^T + R_k) \tag{23}$$

in which the prior mean and covariance are:

$$x_k = f(\hat{x}_{k-1}, u_k), \quad P_k = F\hat{P}_{k-1}F^T + R_k \tag{24}$$

then for the observation part we have:

$$P(z_k|x_k) = N(h(x_k) + H(x_k - \hat{x}_k), Q_k) \tag{25}$$

Define a Kalman gain K_k:

$$K_k = \check{\mathrm{P}}_k H^T (H\check{\mathrm{P}}_k H^T + Q_k)^{-1} \tag{26}$$

and the posterior can be written as:

$$\hat{x}_k = x_k + K_k(z_k - h(x_k)), \hat{P}_k = (I - K_k H)\check{\mathrm{P}} \tag{27}$$

In SLAM, it gives the maximum a posteriori estimate (MAP) under a single linearization step.

2 Bundle Adjustment and Graph Optimization

2.1 Projection model and cost function

A. Transform the world coordinates of point p into the camera frame using extrinsics:

$$P' = Rp + t = [X', Y', Z'] \tag{28}$$

B. Then project P' into the normalized plane and get the normalized coordinates:

$$P_c = [u_c, v_c, 1]^T = [X'/Z', Y'/Z', 1]^T \tag{29}$$

C. Apply distortion model (only radial distortion here):

$$\begin{cases} u_c' = u_c(1 + k_1 r_c^2 + k_2 r_c^4) \\ v_c' = v_c(1 + k_1 r_c^2 + k_2 r_c^4) \end{cases} \tag{30}$$

D. compute the pixel coordinates using intrinsics:

$$\begin{cases} u_s = f_x u_c' + c_x \\ v_s = f_y v_c' + c_y \end{cases} \tag{31}$$

we denote this entire process as the observation equation:

$$z = h(x, y) \tag{32}$$

and the observation data is the pixel coordinate $z \triangleq [u_s, v_s]^T$, then the error of this observation becomes:

$$e = z - h(T, p) \tag{33}$$

denote z_{ij} as the data generated by observing landmark P_j at the pose T_i, then the overall cost function:

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}||e_{ij}||^2 = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}||z_{ij} - h(T_i, p_j)||^2 \tag{34}$$

which is equivalent to adjusting the pose and road signs at the same time, which is the so-called BA.

2.2 Solve bundle adjustment

Optimize all variables together:

$$x = [T_1, ..., T_m, p_1, ..., p_n]^T \tag{35}$$

when we give an increment to the optimization variable, the objective function becomes:

$$\frac{1}{2}||f(x+\Delta x)||^2 \approx \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}||e_{ij} + F_{ij}\Delta\xi_i + E_{ij}\Delta p_j||^2 \tag{36}$$

in which F_{ij} is the partial derivative of the entire cost function to the i-th pose, and E_{ij} is the partial derivative of function to the j-th landmark. Put the camera pose variable together:

$$x_c = [\xi_1, \xi_2, ..., \xi_m]^T \in \mathbb{R}^{6m} \tag{37}$$

and also the landmarks together:

$$x_p = [p_1, p_2, ..., p_n]^T \in \mathbb{R}^{3n} \tag{38}$$

then equation 36 becomes:

$$\frac{1}{2}||f(x+\Delta x)||^2 = \frac{1}{2}||e + F\Delta x_c + E\Delta x_p||^2 \tag{39}$$

we will face the incremental equation:

$$H\Delta x = g \tag{40}$$

H is either J^TJ or $J^TJ + \lambda I$ depending on whether it's Gauss Newton method or Levenberg-Marquardt method. The Jacobian matrix can be divided into two parts:

$$J = [F \quad E] \tag{41}$$

and we have

$$H = J^TJ = \begin{bmatrix} F^TF & F^TE \\ E^TF & E^TE \end{bmatrix} \tag{42}$$

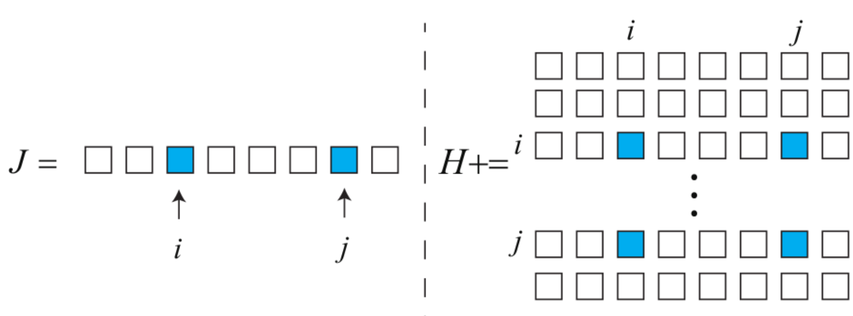
the inversion of H has $O(n^3)$ complexity

2.3 Sparsity

Consider one of the error term e_{ij}, note that this error term only describes the residual about p_j in T_i, and only involves the i-th camera pose and the j-th landmark. The derivatives of all the remaining variables are 0. The Jacobian matrix corresponding to the error term has the following form:

$$J_{ij}(x) = (0_{2\times 6}, ...0_{2\times 6}, \frac{\partial e_{ij}}{\partial T_i}, 0_{2\times 6}, ...0_{2\times 3}, ...0_{2\times 3}, \frac{\partial e_{ij}}{\partial p_j}, 0_{2\times 3}, ...0_{2\times 3}) \tag{43}$$

which causes the sparsity. In the above image, the non-zero blocks are at (i,i), (i,j), (j,i), (j,j):



$$H = \sum_{i,j} J_{ij}^T J_{ij} \tag{44}$$

we can divide H into blocks:

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \tag{45}$$

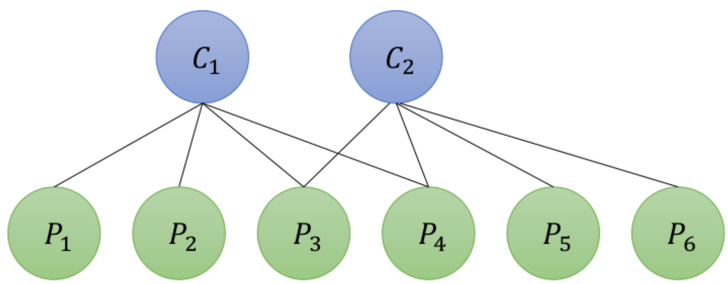
in which H_{11} is only related to camera pose and H_{22} is only related to landmarks. when we iterate over the i, j index, the following properties hold:

A. No matter how i, j changes, H_{11} is always a block-diagonal matrix, with only non-zero blocks at $H_{i,i}$

B. same for H_{22}

C. H_{12} or H_{21} maybe sparse or dense depending on the specific observation data

suppose there are two camera poses (C_1, C_2) and 6 landmarks $(P_1, P_2, P_3, P_4, P_5, P_6)$ in the scene. The variables corresponding to these cameras and point clouds are T_i and p_j. Suppose camera C_1 observes landmarks P_1, P_2, P_3, p_4 and camera C_2 observes landmarks P_3, P_4, P_5, P_6. the cost function the becomes:

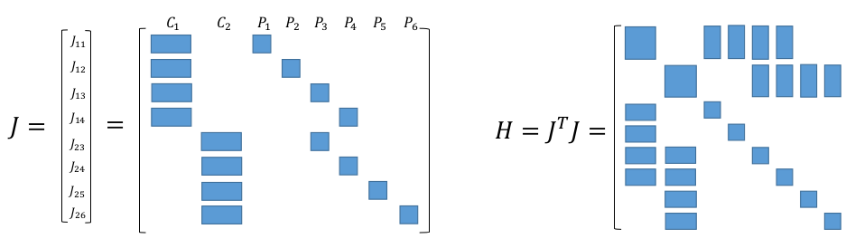


$$\frac{1}{2}(||e_{11}||^2 + ||e_{12}||^2 + ||e_{13}||^2 + ||e_{14}||^2 + ||e_{23}||^2 + ||e_{24}||^2 + ||e_{25}||^2 + ||e_{26}||^2) \tag{46}$$

Let J_{11} be the Jacobian matrix corresponding to e_{11}, and it is not difficult to see that the partial derivatives of ξ_2 and landmarks:

$$J_{11} = \frac{\partial e_{11}}{\partial x} = (\frac{\partial e_{11}}{\partial \xi_1}, 0_{2\times 6}, \frac{\partial e_{11}}{\partial p_1}, 0_{2\times 3}, 0_{2\times 3}, 0_{2\times 3}, 0_{2\times 3}, 0_{2\times 3}) \tag{47}$$

consider there are m cameras and n landmarks, since there are far more landmarks than cameras, we have



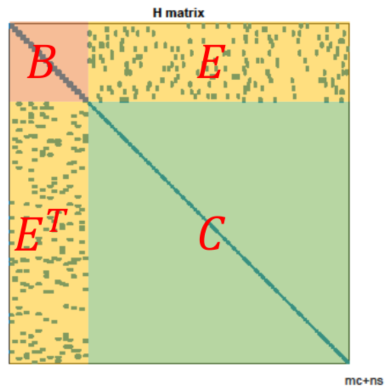
n $\gg$ m. The actual H matrix will be arrow-like.

2.4 Schur trick

The linear equation $Hx = g$ can be rewritten as:

$$\begin{bmatrix} B & E \\ E^T & C \end{bmatrix} \begin{bmatrix} \Delta x_c \\ \Delta x_p \end{bmatrix} = \begin{bmatrix} v \\ w \end{bmatrix} \tag{48}$$

in which B is a block-diagonal matrix, the dimension of each diagonal block is the same as the dimension of the camera pose. The number of diagonal blocks is the number of camera variables. C is often much larger than B, with each block being a 3×3 matrix. Perform Gaussian elimination on the linear equation, we multiply a coefficient matrix on the left side:



$$\begin{bmatrix} I & -EC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} B & E \\ E^T & C \end{bmatrix} \begin{bmatrix} \Delta x_c \\ \Delta x_p \end{bmatrix} = \begin{bmatrix} I & -EC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{49}$$

rearrange it:

$$\begin{bmatrix} B - EC^{-1}E^T & 0 \\ E^T & C \end{bmatrix} \begin{bmatrix} \Delta x_c \\ \Delta x_p \end{bmatrix} = \begin{bmatrix} v - EC^{-1}w \\ w \end{bmatrix} \tag{50}$$

take the first row of the equation and get the incremental equation for the pose part:

$$\left[B - EC^{-1}E^T\right] \Delta x_c = v - EC^{-1}w \tag{51}$$

thus we can solve Δx_c first, then plug into the original equation and solve Δx_p later ($\Delta x_p = C^{-1}(w - E^T \Delta x_c)$). This is called Schur elimination. Another way of marginalization is by Cholesky decomposition. Eliminating camera variables is also commonly used in SLAM.

2.5 Robust kernels

Anomaly data might introduce a large gradient, thus eliminating the influence of other correct edges.

Kernel functions: makes sure error of each edge will not be big enough to cover other edges, replace $\mathcal{L}_2$ norm. For example, the Huber kernel:

$$H(e) = \begin{cases} \frac{1}{2}e^2 & \text{when } |e| \leq \delta \\ \delta(|e| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \tag{52}$$