

IN2106 Practical Course – Vision-based Navigation: Exercise #3

Min-An Chao (03681062) | TUM MS Informatics | ga83fok@mytum.de

06.05.2018

1 Batch MAP

Task 1 By taking $k = 0, 1, 2$ into the system model, We can obtain

$$\mathbf{H} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1)$$

Task 2 From the system model and Eq. 1, we have,

$$\mathbf{e} = \mathbf{z} - \mathbf{H}\mathbf{x} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} - \begin{pmatrix} x_1 - x_0 \\ x_2 - x_1 \\ x_3 - x_2 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ n_1 \\ n_2 \\ n_3 \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{W}). \quad (2)$$

Obviously, if the noise is independent of each other, we have

$$\mathbf{W} = \text{diag}(Q, Q, Q, R, R, R), \quad (3)$$

otherwise \mathbf{W} could be any covariance matrix with this diagonal terms and other nonzero entries.

Task 3 Yes, if the noise is uncorrelated, then it is independent. In this case there is only one exact solution of \mathbf{W} .

2 Iterative Curve Fitting

Task 1 The derivatives of error to parameters could be obtained by

$$\mathbf{J}_i = \begin{pmatrix} \frac{\partial e_i}{\partial a} \\ \frac{\partial e_i}{\partial b} \\ \frac{\partial e_i}{\partial c} \end{pmatrix} = \begin{pmatrix} -f(x_i) \cdot x_i^2 \\ -f(x_i) \cdot x_i \\ -f(x_i) \end{pmatrix}, \quad (4)$$

where

$$f(x_i) = \exp(ax_i^2 + bx_i + c). \quad (5)$$

Then we have

$$\begin{aligned}\mathbf{H} &= \sum_i \mathbf{J}_i \mathbf{J}_i^T, \\ \mathbf{b} &= \sum_i -e_i \mathbf{J}_i, \\ (\Delta a, \Delta b, \Delta c)^T &= \mathbf{H}^{-1} \mathbf{b}.\end{aligned}\tag{6}$$

By implementing this, we can see the results:

```

1 total cost: 3.19575e+06
2 total cost: 376785
3 total cost: 35673.6
4 total cost: 2195.01
5 total cost: 174.853
6 total cost: 102.78
7 total cost: 101.937
8 total cost: 101.937
9 total cost: 101.937
10 total cost: 101.937
11 total cost: 101.937
12 total cost: 101.937
13 cost: 101.937, last cost: 101.937
14 estimated abc = 0.890912, 2.1719, 0.943629

```

Task 2 With Google Ceres, first we have to implement the $e_i = y_i - f(x_i)$ part in the overloaded `()` function, *i.e.* `operator()(const T *const abc, T *residual) const`, inside the `CURVE_FITTING_COST` object. then we instantiate a `Problem` object, adding the cost function of the `CURVE_FITTING_COST` object into a `CostFunction` object by `AutoDiffCostFunction()` method, and using `Problem::AddResidualBlock()` method to add the residual block we have implemented inside the overloaded `()` function. Finally by the solver setups shown in lecture slides and by the solver function call we have the results:

```

1 iter      cost      cost_change |gradient| |step|    ... total_time
2   0  1.597873e+06    0.00e+00  3.52e+06  0.00e+00  ...   1.17e-02
3   1  1.884440e+05    1.41e+06  4.86e+05  9.88e-01  ...   1.66e-02
4   2  1.784821e+04    1.71e+05  6.78e+04  9.89e-01  ...   1.99e-02
5   3  1.099631e+03    1.67e+04  8.58e+03  1.10e+00  ...   2.29e-02
6   4  8.784938e+01    1.01e+03  6.53e+02  1.51e+00  ...   2.58e-02
7   5  5.141230e+01    3.64e+01  2.72e+01  1.13e+00  ...   2.79e-02
8   6  5.096862e+01    4.44e-01  4.27e-01  1.89e-01  ...   2.98e-02
9   7  5.096851e+01    1.10e-04  9.53e-04  2.84e-03  ...   3.17e-02
10 Ceres Solver Report: Iterations: 8, Initial cost: 1.597873e+06,
11 Final cost: 5.096851e+01, Termination: CONVERGENCE
12 estimated a,b,c = 0.890908, 2.1719, 0.943628

```

3 Camera Pose Estimation by Gauss-Newton Method

Task 1 Suppose the rotation matrix of the estimated pose \mathbf{T}_k is \mathbf{R}_k and the translation part is \mathbf{t}_k , where k stands for the iteration count. The projected position of 3D points \mathbf{p}_i would be

$$\begin{aligned}\mathbf{q}_{k,i} &= \mathbf{R}_{k-1} \mathbf{p}_i + \mathbf{t}_{k-1} \\ &= (x_{k,i}, y_{k,i}, z_{k,i})^T.\end{aligned}\tag{7}$$

The re-projection error $\mathbf{e}_{k,i}$ is obtained by applying the intrinsic matrix to the projected position with the normalization term $z_{k,i}$, or equally, $q'_{k,i}{}^{(2)}$, which is,

$$\begin{aligned} \mathbf{q}'_{k,i} &= \mathbf{K} \mathbf{q}_{k,i}, \\ \mathbf{e}_{k,i} &= \mathbf{u}_i - \frac{1}{z_{k,i}} \cdot (q'_{k,i}{}^{(0)}, q'_{k,i}{}^{(1)})^T, \end{aligned} \quad (8)$$

Task 2 The Jacobian matrix of error could be derived, based on the slides, as

$$\mathbf{J}_{k,i} = \begin{pmatrix} -\frac{f_x}{z_{k,i}} & 0 & \frac{f_x x_{k,i}}{z_{k,i}^2} & \frac{f_x x_{k,i} y_{k,i}}{z_{k,i}^2} & -f_x - \frac{f_x x_{k,i}^2}{z_{k,i}^2} & \frac{f_x y_{k,i}}{z_{k,i}} \\ 0 & -\frac{f_y}{z_{k,i}} & \frac{f_y y_{k,i}}{z_{k,i}^2} & f_y + \frac{f_y y_{k,i}^2}{z_{k,i}^2} & -\frac{f_y x_{k,i} y_{k,i}}{z_{k,i}^2} & -\frac{f_y x_{k,i}}{z_{k,i}} \end{pmatrix}, \quad (9)$$

where $x_{k,i}, y_{k,i}, z_{k,i}$ comes from Eq. 7, and $f_x = K_{0,0}, f_y = K_{1,1}$.

Task 3 Updates on estimation pose are made iterative by

$$\begin{aligned} \mathbf{H}_k &= \sum_i \mathbf{J}_{k,i}^T \mathbf{J}_{k,i}, \\ \mathbf{b}_k &= \sum_i -\mathbf{J}_{k,i}^T \mathbf{e}_{k,i}, \\ \Delta \mathbf{T}_k &= \mathbf{H}_k^{-1} \mathbf{b}_k, \\ \mathbf{T}_k &= \exp((\Delta \mathbf{T}_k)^\wedge) \mathbf{T}_{k-1}. \end{aligned} \quad (10)$$

$\Delta \mathbf{T}_k$ here is expressed as a $\mathfrak{se}(3)$ 6-by-1 vector. Updates can be applied by calculating its exponential map, and multiplying it to the previous pose \mathbf{T}_{k-1} . The results are shown below.

```

1 points: 76
2 iteration 0 cost=45538.2
3 iteration 1 cost=413.209
4 iteration 2 cost=302.649
5 iteration 3 cost=301.357
6 iteration 4 cost=301.351
7 iteration 5 cost=301.351
8 iteration 6 cost=301.351
9 iteration 7 cost=301.351
10 iteration 8 cost=301.351
11 cost: 301.351, last cost: 301.351
12 estimated pose:
13 0.997866 -0.0516724 0.0399128 -0.127227
14 0.0505959 0.99834 0.0275274 -0.0075068
15 -0.0412689 -0.0254492 0.998824 0.0613861
16 0 0 0 1
    
```