HW5

July 2019

1 Q1

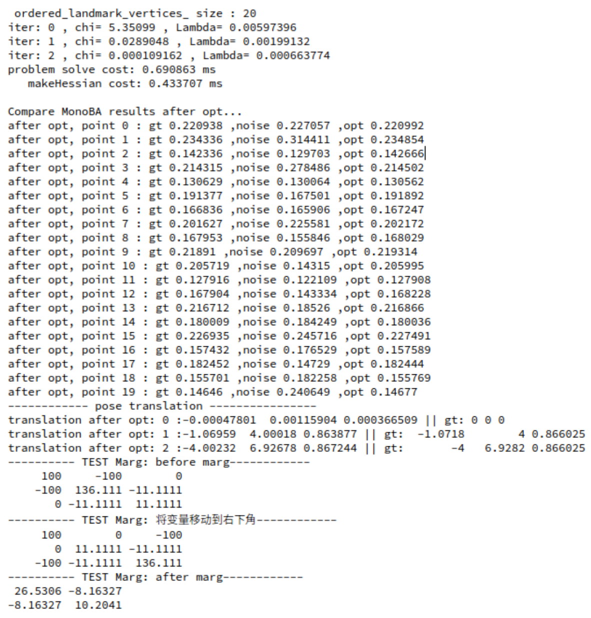
$$\begin{bmatrix} H_{pp} & H_{pl} \\ H_{lp} & H_{ll} \end{bmatrix} \begin{bmatrix} \Delta x_p^* \\ \Delta x_l^* \end{bmatrix} = \begin{bmatrix} -b_p \\ -b_l \end{bmatrix} \tag{1}$$

Apply the Schur Complement,

$$(H_{pp} - H_{pl}H_{ll}^{-1}H_{pl}^T)\Delta x_p^* = -bp + H_{pl}H_{ll}^{-1}b_l \tag{2}$$

Then, we have

$$H_{ll}\Delta x_l^* = -b_l - H_{pl}^T \Delta x_p^* \tag{3}$$



```
 ordered_landmark_vertices_ size : 20
iter: 0 , chi= 5.35099 , Lambda= 0.00597396
iter: 1 , chi= 0.0289048 , Lambda= 0.00199132
iter: 2 , chi= 0.000109162 , Lambda= 0.000663774
problem solve cost: 0.690863 ms
   makeHessian cost: 0.433707 ms

Compare MonoBA results after opt...
after opt, point 0 : gt 0.220938 ,noise 0.227057 ,opt 0.220992
after opt, point 1 : gt 0.234336 ,noise 0.314411 ,opt 0.234854
after opt, point 2 : gt 0.142336 ,noise 0.129703 ,opt 0.142666
after opt, point 3 : gt 0.214315 ,noise 0.278486 ,opt 0.214502
after opt, point 4 : gt 0.130629 ,noise 0.130064 ,opt 0.130562
after opt, point 5 : gt 0.191377 ,noise 0.167501 ,opt 0.191892
after opt, point 6 : gt 0.166836 ,noise 0.165906 ,opt 0.167247
after opt, point 7 : gt 0.201627 ,noise 0.225581 ,opt 0.202172
after opt, point 8 : gt 0.167953 ,noise 0.155846 ,opt 0.168029
after opt, point 9 : gt 0.21891 ,noise 0.209697 ,opt 0.219314
after opt, point 10 : gt 0.205719 ,noise 0.14315 ,opt 0.205995
after opt, point 11 : gt 0.127916 ,noise 0.122109 ,opt 0.127908
after opt, point 12 : gt 0.167904 ,noise 0.143334 ,opt 0.168228
after opt, point 13 : gt 0.216712 ,noise 0.18526 ,opt 0.216866
after opt, point 14 : gt 0.180009 ,noise 0.184249 ,opt 0.180036
after opt, point 15 : gt 0.226935 ,noise 0.245716 ,opt 0.227491
after opt, point 16 : gt 0.157432 ,noise 0.176529 ,opt 0.157589
after opt, point 17 : gt 0.182452 ,noise 0.14729 ,opt 0.182444
after opt, point 18 : gt 0.155701 ,noise 0.182258 ,opt 0.155769
after opt, point 19 : gt 0.14646 ,noise 0.240649 ,opt 0.14677
------------ pose translation ----------------
translation after opt: 0 :-0.00047801  0.00115904 0.000366509 || gt: 0 0 0
translation after opt: 1 :-1.06959  4.00018 0.863877 || gt:  -1.0718        4 0.866025
translation after opt: 2 :-4.00232  6.92678 0.867244 || gt:       -4   6.9282 0.866025
---------- TEST Marg: before marg------------
     100     -100        0
    -100  136.111 -11.1111
       0 -11.1111  11.1111
---------- TEST Marg: 将变量移动到右下角------------
     100        0     -100
       0  11.1111 -11.1111
    -100 -11.1111  136.111
---------- TEST Marg: after marg------------
 26.5306 -8.16327
-8.16327  10.2041
```

Figure 1: Q1 Q2 Result

2 Q3

There are three possible approaches to solving the problem of gauge freedom. The first solution is to fix the corresponding states (i.e., parameters) in the optimization. The second approach is to set a prior on the unobservable states, and the prior essentially acts as a virtual measurement in the optimization. Finally, one way instead allow the optimization algorithm to change the unobservable states freely during the iterations [1].

2.1 Fixed Gauge

This method aims at optimizing in a small parameter space where there are no unobservable states, which means the Hessian matrix is invertible. This method enforces hard constraints on the solution.

By fixing the position and yaw angle of the first camera pose throughout the optimization.

$$p_0 = p_0^0,\ \Delta\phi_{0z} = e_z^T \Delta_0 = 0 \tag{4}$$

where p_0^0 is the initial position of the first camera. Fixing these values of the parameter vector is equivalent to setting the corresponding columns of the Jacobian matrix of the residual vector to zero.

2.2 Gauge Prior

This method is to augment the objective function with an additional penalty, which yields an invertible Hessian matrix. Then the solution is able to satisfy certain constraints. The penalty added to the objective function is $||r_0^P||^2_{\sum_0^P}$, where $r_0^p(\theta) = (p_0 - p_0^0, \Delta\phi_{0z})$.

2.3 Free Gauge

The last method is a generic method which uses the pseudo-inverse of the singular Hessian matrix to implicitly provide additional constraints for a unique solution (e.g. parameter updates with smallest norm).

2.4 Comparison

2.4.1 Accuracy and timing

These three approaches have almost the same accuracy. The gauge prior approach needs to select the proper prior weight to avoid increasing the computational cost. With a proper weight, the gauge prior approach has almost the same performance as the gauge fixation approach. Furthermore, the free gauge approach is slightly faster than the others, because it takes fewer iterations to converge.

2.4.2 Covariance

The covariance matrix from the gauge prior approach is similar to the gauge fixation approach. However, the estimated covariance from the free gauge approach is different. For example, for the gauge fixation approach, the uncertainty of the first position is zero due to the fixation and the position uncertainty increases afterwards. In contrast, the uncertainty in the free gauge method is distributed over all the positions. This due to the fact that the free gauge approach is not fixed to any reference frame. Therefore, the uncertainties directly read from the free gauge covariance matrix are not interpretable in a geometrically-meaningful way. Therefore, the free gauge covariance should be transformed to a geometrically-meaningful form by enforcing a gauge fixation condition.

To transform the free gauge covariance, we first transform θ to the gauge fixation estimation θ_C along M_θ which is the subspace that contains all the parameters that satisfy the gauge fixation condition with the perturbation $\Delta\theta$.

$$\Delta\theta \mapsto \frac{\partial\theta_C}{\partial\theta}\Delta\theta \tag{5}$$

Then we project the perturbation onto the tangent space to the gauge, parallel to the M_θ, using projector $Q^C_{\theta_C}$. Then, the transformed covariance can be calculated using the average of the outer product of these transformed perturbations.

It can be seen that the transformed covariance agrees well with the covariance from the gauge fixation from the experiment result.

References

[1] Z. Zhang, G. Gallego, and D. Scaramuzza, "On the comparison of gauge freedom handling in optimization-based visual-inertial state estimation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2710–2717, July 2018.