

结果如下

```
(base) wang@wang-GL63-8RE:~/workspace/VIO-SLAM/week5/hw_course5/build/app$
./testMonoBA
0 order: 0
1 order: 6
2 order: 12

ordered_landmark_vertices_size : 20
iter: 0 , chi= 5.35099 , Lambda= 0.00597396
iter: 1 , chi= 0.565419 , Lambda= 0.00611916
iter: 2 , chi= 0.396546 , Lambda= 8555.2
iter: 3 , chi= 0.395999 , Lambda= 365022
problem solve cost: 5.18119 ms
    makeHessian cost: 1.5764 ms

Compare MonoBA results after opt...
after opt, point 0 : gt 0.220938 ,noise 0.227057 ,opt 0.209286
after opt, point 1 : gt 0.234336 ,noise 0.314411 ,opt 0.229329
after opt, point 2 : gt 0.142336 ,noise 0.129703 ,opt 0.134867
after opt, point 3 : gt 0.214315 ,noise 0.278486 ,opt 0.20958
after opt, point 4 : gt 0.130629 ,noise 0.130064 ,opt 0.122902
after opt, point 5 : gt 0.191377 ,noise 0.167501 ,opt 0.183971
after opt, point 6 : gt 0.166836 ,noise 0.165906 ,opt 0.163685
after opt, point 7 : gt 0.201627 ,noise 0.225581 ,opt 0.189041
after opt, point 8 : gt 0.167953 ,noise 0.155846 ,opt 0.163978
after opt, point 9 : gt 0.21891 ,noise 0.209697 ,opt 0.207129
after opt, point 10 : gt 0.205719 ,noise 0.14315 ,opt 0.20486
after opt, point 11 : gt 0.127916 ,noise 0.122109 ,opt 0.119493
after opt, point 12 : gt 0.167904 ,noise 0.143334 ,opt 0.163878
after opt, point 13 : gt 0.216712 ,noise 0.18526 ,opt 0.207956
after opt, point 14 : gt 0.180009 ,noise 0.184249 ,opt 0.176345
after opt, point 15 : gt 0.226935 ,noise 0.245716 ,opt 0.22075
after opt, point 16 : gt 0.157432 ,noise 0.176529 ,opt 0.151683
after opt, point 17 : gt 0.182452 ,noise 0.14729 ,opt 0.173098
after opt, point 18 : gt 0.155701 ,noise 0.182258 ,opt 0.143187
after opt, point 19 : gt 0.14646 ,noise 0.240649 ,opt 0.139413
----- pose translation -----
translation after opt: 0 :-0.101357  0.109785 0.0168622 || gt: 0 0 0
translation after opt: 1 :-0.922321  4.06589  0.786338 || gt: -1.0718      4
0.866025
translation after opt: 2 : -3.9061  7.04537 0.872799 || 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
```

修改的代码：

```
// TODO:: home work. 完成 H index 的填写.
// H.block(?,?, ?, ?).noalias() += hessian;
    H.block(index_i, index_j,
dim_i, dim_j).noalias() += hessian.transpose();
    if (j != i) {
        // 对称的下三角
        // TODO:: home work. 完成 H index 的填写.
        // H.block(?,?, ?, ?).noalias() += hessian.transpose();
        H.block(index_j, index_i,
dim_j, dim_i).noalias() += hessian.transpose();
    }

// TODO:: home work. 完成矩阵块取值, Hmm, Hpm, Hmp, bpp, bmm
    MatXX Hmm = Hessian_.block(reserve_size, reserve_size, marg_size,
marg_size);
    MatXX Hpm = Hessian_.block(0, reserve_size, reserve_size, marg_size);
    MatXX Hmp = Hessian_.block(reserve_size, 0, marg_size, reserve_size);
    VecX bpp = b_.segment(0, reserve_size);
    VecX bmm = b_.segment(reserve_size, marg_size);

    // TODO:: home work. 完成舒尔补 Hpp, bpp 代码
    MatXX tempH = Hpm * Hmm_inv;
    H_pp_schur_ = Hessian_.block(0, 0, reserve_size, reserve_size) - tempH *
Hmp.transpose();
    b_pp_schur_ = bpp - tempH * bmm;

    // TODO:: home work. step3: solve landmark
    VecX delta_x_ll(marg_size);
    // delta_x_ll = ???;
    delta_x_ll = Hmm_inv * (bmm - Hpm.transpose() * delta_x_pp);
    delta_x_.tail(marg_size) = delta_x_ll;

    // TODO:: home work. 将变量移动到右下角
    /// 准备工作: move the marg pose to the Hmm bottown right
    // 将 row i 移动矩阵最下面
    Eigen::MatrixX<double> temp_rows = H_marg.block(idx, 0, dim, reserve_size);
    Eigen::MatrixX<double> temp_botRows = H_marg.block(idx + dim, 0, reserve_size - idx
- dim, reserve_size);
    H_marg.block(idx, 0, reserve_size - idx - dim, reserve_size) = temp_botRows;
    H_marg.block(reserve_size - dim, 0, dim, reserve_size) = temp_rows;

    // TODO:: home work. 完成舒尔补操作
    Eigen::MatrixX<double> Arm = H_marg.block(0, n2, n2, m2);
    Eigen::MatrixX<double> Amr = H_marg.block(n2, 0, m2, n2);
    Eigen::MatrixX<double> Arr = H_marg.block(0, 0, n2, n2);
```