

2.1.1 可以利用 H 矩阵的稀疏性进行优化。

2.1.2 BA 中需要被参数化的地方包括：相机位姿、相机内参、三维特征点 P 以及投影后的像素坐标；

位姿：

欧拉角、四元数、旋转矩阵+平移；

欧拉角存在万向锁问题；

旋转矩阵直观，但自由度过多；

四元数参数简洁，不直观。

Point 参数化方式：

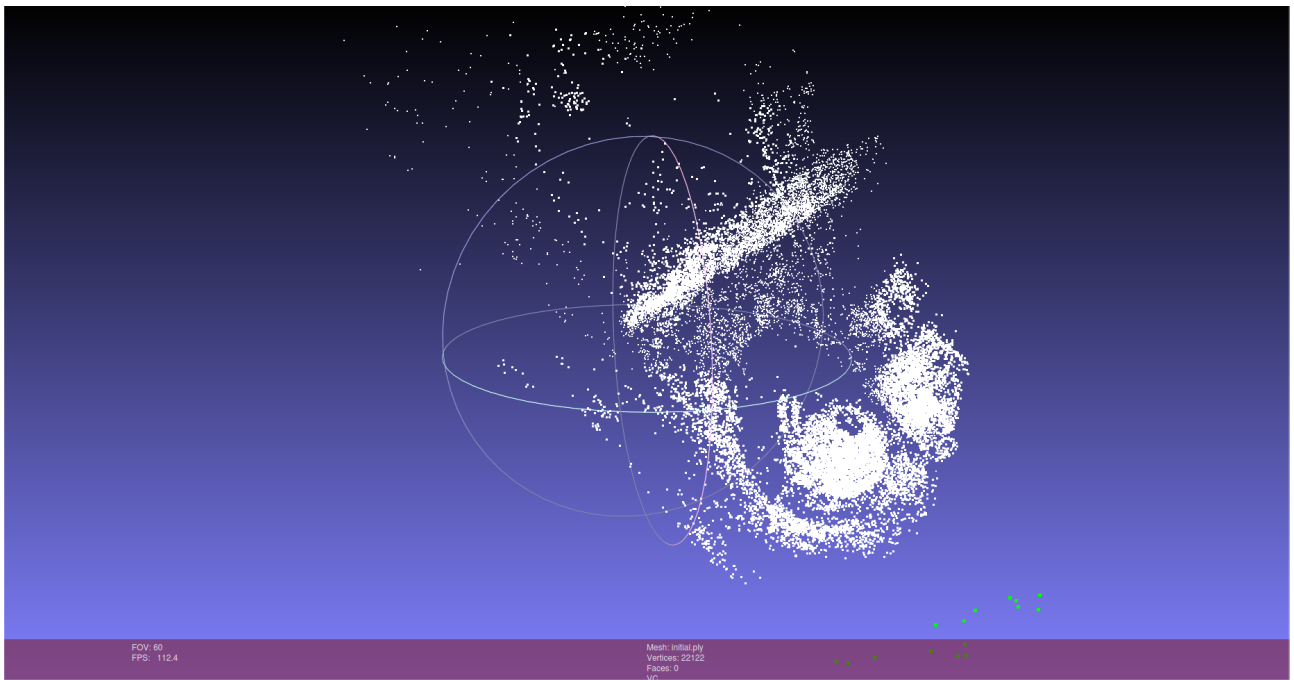
点的参数化表示包括三维坐标 XYZ 和逆深度表示方法。

2.1.3 3.4 节的 Intensity- based methods 就是 BA 在直接法中的应用。第 5 节 Network Structure 可以对应到 SLAM 中的图优化模型；H 的稀疏性可以实现 BA 实时，在 07 年的 PTAM 上实现。

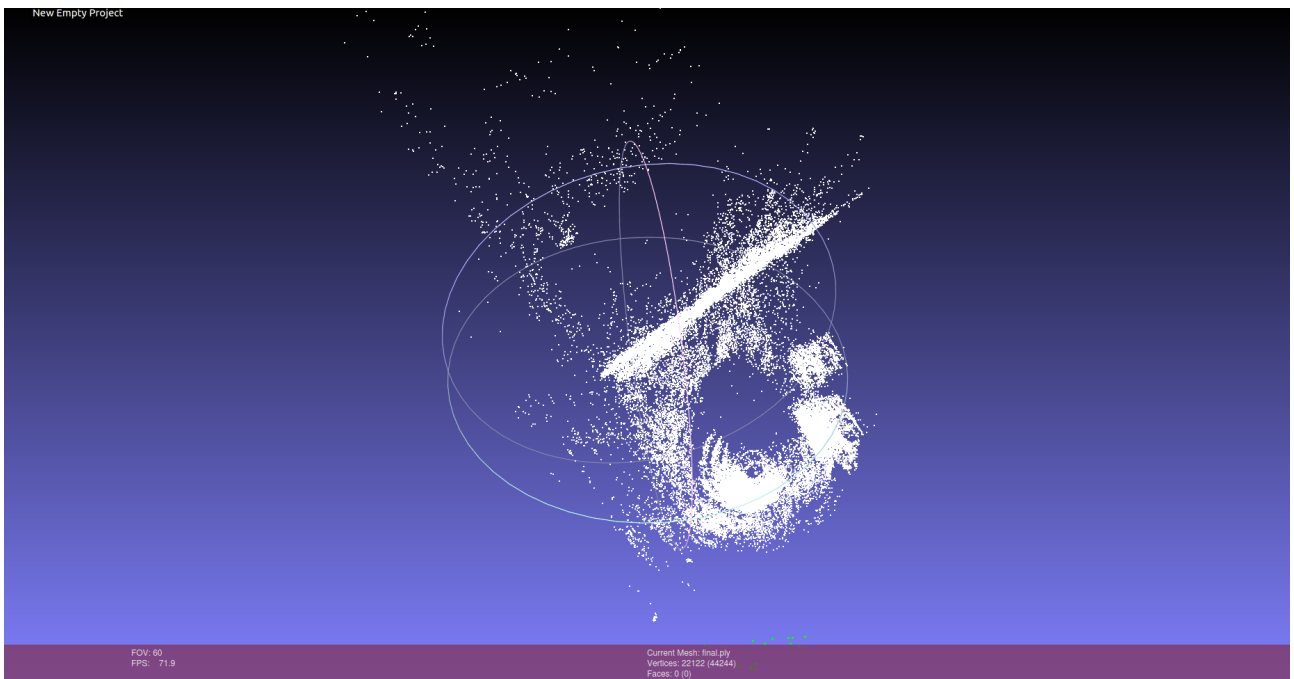
2.2 程序运行如图

iteration= 0	chi2= 8894423.022949	time= 0.209875	cumTime= 0.209875	edges= 83718	schur= 1	lambda= 227.832660	levenbergIter= 1
iteration= 1	chi2= 1772145.050517	time= 0.191742	cumTime= 0.401616	edges= 83718	schur= 1	lambda= 75.944220	levenbergIter= 1
iteration= 2	chi2= 752585.293391	time= 0.19065	cumTime= 0.592267	edges= 83718	schur= 1	lambda= 25.314740	levenbergIter= 1
iteration= 3	chi2= 402814.243627	time= 0.192646	cumTime= 0.784913	edges= 83718	schur= 1	lambda= 8.438247	levenbergIter= 1
iteration= 4	chi2= 284879.378894	time= 0.194471	cumTime= 0.979384	edges= 83718	schur= 1	lambda= 2.812749	levenbergIter= 1
iteration= 5	chi2= 238356.214415	time= 0.191242	cumTime= 1.17063	edges= 83718	schur= 1	lambda= 0.937583	levenbergIter= 1
iteration= 6	chi2= 193550.755079	time= 0.190697	cumTime= 1.36132	edges= 83718	schur= 1	lambda= 0.312528	levenbergIter= 1
iteration= 7	chi2= 146859.909574	time= 0.189375	cumTime= 1.5507	edges= 83718	schur= 1	lambda= 0.104176	levenbergIter= 1
iteration= 8	chi2= 122887.700218	time= 0.189324	cumTime= 1.74002	edges= 83718	schur= 1	lambda= 0.069451	levenbergIter= 1
iteration= 9	chi2= 97810.139925	time= 0.193256	cumTime= 1.93328	edges= 83718	schur= 1	lambda= 0.046300	levenbergIter= 1
iteration= 10	chi2= 80329.940265	time= 0.189731	cumTime= 2.12301	edges= 83718	schur= 1	lambda= 0.030867	levenbergIter= 1
iteration= 11	chi2= 65663.994405	time= 0.194739	cumTime= 2.31775	edges= 83718	schur= 1	lambda= 0.020578	levenbergIter= 1
iteration= 12	chi2= 55960.726637	time= 0.195212	cumTime= 2.51296	edges= 83718	schur= 1	lambda= 0.013719	levenbergIter= 1
iteration= 13	chi2= 53275.547797	time= 0.215142	cumTime= 2.7281	edges= 83718	schur= 1	lambda= 0.009146	levenbergIter= 1
iteration= 14	chi2= 35983.312124	time= 0.245864	cumTime= 2.97396	edges= 83718	schur= 1	lambda= 0.006097	levenbergIter= 2
iteration= 15	chi2= 32091.891518	time= 0.283715	cumTime= 3.25768	edges= 83718	schur= 1	lambda= 0.016259	levenbergIter= 3
iteration= 16	chi2= 31156.262647	time= 0.239151	cumTime= 3.49683	edges= 83718	schur= 1	lambda= 0.021679	levenbergIter= 2

初始点云如图



优化后的点云如图



3.1.1

$$error = I(p_i) - I_j(\pi(KT_j p_i))$$

3.1.2 每个 error 关联两个优化变量，6 自由度的相机位姿和 3 自由度的空间点

3.1.3

$$J = -\frac{\partial I_2}{\partial u} \frac{\partial u}{\partial \delta \xi}, \quad \frac{\partial u}{\partial \delta \xi} = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} & -\frac{f_x XY}{Z^2} & f_x + \frac{f_x X^2}{Z^2} & -\frac{f_x Y}{Z} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} & -f_y - \frac{f_y Y^2}{Z^2} & \frac{f_y XY}{Z^2} & \frac{f_y X}{Z} \end{bmatrix}$$

$$\frac{\partial e}{\partial \xi} = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} & -\frac{f_x XY}{Z^2} & f_x + \frac{f_x X^2}{Z^2} & -\frac{f_x Y}{Z} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} & -f_y - \frac{f_y Y^2}{Z^2} & \frac{f_y XY}{Z^2} & \frac{f_y X}{Z} \end{bmatrix}$$

3.2.1 可以，还可以使用逆深度的方法来参数化路标点。

3.2.2 太小的 patch 不能反应真正的光度变化。固定场景的话可能更大一点好，但会增加运算量。

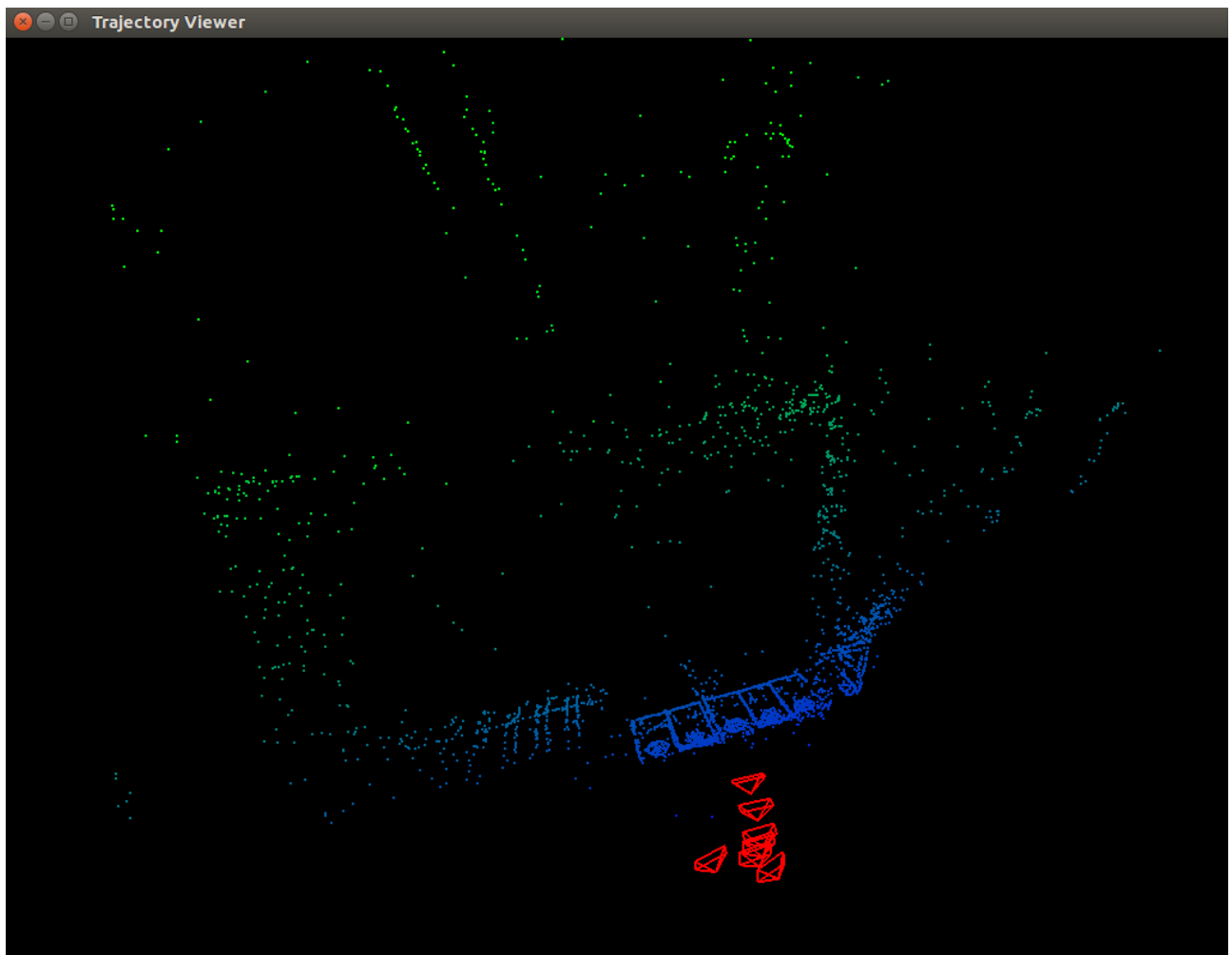
3.2.3 计算误差的方式不同，特征点法在 BA 阶段最小化特征点的重投影误差，直接法最小化的是像素点块的光度误差；因此其雅可比计算也不一样。

3.2.4 做多次测验，取误差最小的阈值作为 Huber 的阈值。

程序运行如下

```
poses: 7, points: 4118
images: 7
iteration= 0   chi2= 3755530.441268   time= 0.180668   cumTime= 0.180668   edges= 28826   schur= 1   lambda= 57528453.790812   levenbergIter= 1
iteration= 1   chi2= 3749287.192116   time= 0.166779   cumTime= 0.347448   edges= 28826   schur= 1   lambda= 38352302.527208   levenbergIter= 1
iteration= 2   chi2= 3745223.294910   time= 0.163275   cumTime= 0.510722   edges= 28826   schur= 1   lambda= 25568201.684806   levenbergIter= 1
iteration= 3   chi2= 3739628.535085   time= 0.167031   cumTime= 0.677753   edges= 28826   schur= 1   lambda= 17045467.789870   levenbergIter= 1
iteration= 4   chi2= 3738741.514767   time= 0.177719   cumTime= 0.854944   edges= 28826   schur= 1   lambda= 11363645.193247   levenbergIter= 1
iteration= 5   chi2= 3737916.057401   time= 0.174442   cumTime= 1.02939   edges= 28826   schur= 1   lambda= 7575763.462165   levenbergIter= 1
iteration= 6   chi2= 3737789.545640   time= 0.192173   cumTime= 1.22156   edges= 28826   schur= 1   lambda= 10101017.949553   levenbergIter= 2
iteration= 7   chi2= 3736137.060606   time= 0.175023   cumTime= 1.39658   edges= 28826   schur= 1   lambda= 6734011.966369   levenbergIter= 1
iteration= 8   chi2= 3735511.179030   time= 0.176035   cumTime= 1.57262   edges= 28826   schur= 1   lambda= 4489341.310912   levenbergIter= 1
iteration= 9   chi2= 3733585.006172   time= 0.189044   cumTime= 1.76166   edges= 28826   schur= 1   lambda= 2992894.207275   levenbergIter= 1
iteration= 10  chi2= 3733163.788255   time= 0.213596   cumTime= 1.97526   edges= 28826   schur= 1   lambda= 3900525.609700   levenbergIter= 2
iteration= 11  chi2= 3731505.883647   time= 0.201159   cumTime= 2.17642   edges= 28826   schur= 1   lambda= 2660350.406467   levenbergIter= 1
iteration= 12  chi2= 3730465.378889   time= 0.258749   cumTime= 2.43516   edges= 28826   schur= 1   lambda= 14188535.501155   levenbergIter= 3
iteration= 13  chi2= 3730155.281849   time= 0.230816   cumTime= 2.66598   edges= 28826   schur= 1   lambda= 18918047.334873   levenbergIter= 2
iteration= 14  chi2= 3728209.728479   time= 0.207871   cumTime= 2.87385   edges= 28826   schur= 1   lambda= 12612031.556582   levenbergIter= 1
iteration= 15  chi2= 3727076.175146   time= 0.229891   cumTime= 3.10374   edges= 28826   schur= 1   lambda= 8408021.037722   levenbergIter= 1
iteration= 16  chi2= 3726372.219265   time= 0.234672   cumTime= 3.33841   edges= 28826   schur= 1   lambda= 5605347.358481   levenbergIter= 1
iteration= 17  chi2= 3726251.465340   time= 0.237286   cumTime= 3.5757   edges= 28826   schur= 1   lambda= 7473796.477975   levenbergIter= 2
iteration= 18  chi2= 3725828.646300   time= 0.23374   cumTime= 3.80944   edges= 28826   schur= 1   lambda= 9965061.970633   levenbergIter= 2
iteration= 19  chi2= 3725641.396886   time= 0.327085   cumTime= 4.13653   edges= 28826   schur= 1   lambda= 53146997.176709   levenbergIter= 3
iteration= 20  chi2= 3723821.173104   time= 0.228971   cumTime= 4.3655   edges= 28826   schur= 1   lambda= 35431331.451139   levenbergIter= 1
iteration= 21  chi2= 3722176.693658   time= 0.22128   cumTime= 4.58678   edges= 28826   schur= 1   lambda= 23620887.634093   levenbergIter= 1
iteration= 22  chi2= 3720684.468952   time= 0.186687   cumTime= 4.77346   edges= 28826   schur= 1   lambda= 15747258.422729   levenbergIter= 1
iteration= 23  chi2= 3720304.250828   time= 0.175729   cumTime= 4.94919   edges= 28826   schur= 1   lambda= 10498172.281819   levenbergIter= 1
iteration= 24  chi2= 3720157.793351   time= 0.175233   cumTime= 5.12443   edges= 28826   schur= 1   lambda= 6998781.521213   levenbergIter= 1
iteration= 25  chi2= 3719659.149600   time= 0.176962   cumTime= 5.30139   edges= 28826   schur= 1   lambda= 4665854.347475   levenbergIter= 1
iteration= 26  chi2= 3718180.816169   time= 0.198189   cumTime= 5.49958   edges= 28826   schur= 1   lambda= 3110569.564983   levenbergIter= 1
iteration= 27  chi2= 3717843.841051   time= 0.192418   cumTime= 5.69199   edges= 28826   schur= 1   lambda= 2073713.043322   levenbergIter= 1
```

优化前点云如下



优化后点云如下

