1 提升部分 2

在代码中给第一帧和第二帧添加 prior 约束, 并比较为 prior 设定不同权重时, BA 求解收敛精度和速度。

主要修改:

在 main 中加入 vector 来储存不同的 weight，并写个大循环遍历每个 weight 值，

在仿照重投影误差中的添加 edge 的方式，添加先验边 EdgeSE3Prior，

除此之外为了将结果输出还进行了一些小修改。

```
    // store all result of different weights
    ofstream result_csv;
    result_csv.open("result.csv", ios_base::out);
    vector<double> weights{1e-2, 1e-1, 0, 1e1, 1e2, 5e2, 1e5, 1e10};

for (auto weight : weights){ // try all the weights
```

```
// add the prior to the first 2 frames
for (size_t i = 0; i < 2; ++i){
    shared_ptr<EdgeSE3Prior> edge_prior(new EdgeSE3Prior(cameras[i].twc, cameras[i].qwc));
    vector<shared_ptr<Vertex>> edge_prior_vertex{vertexCams_vec[i]};
    edge_prior->SetVertex(edge_prior_vertex);
    edge_prior->SetInformation(edge_prior->Information() * weight);
    problem.AddEdge(edge_prior);
}
```

	A	B	C
1	weight is 0.01	6.57996*10^-6	39.696ms
2	weight is 0.1	6.44961*10^-6	21.1515ms
3	weight is 0	3.61547*10^-6	24.0917ms
4	weight is 10	1.94011*10^-6	8.09791ms
5	weight is 100	0.960739*10^-6	5.89124ms
6	weight is 500	1.12049*10^-6	20.7258ms
7	weight is 100000	1.1428*10^-6	27.3678ms
8	weight is 1e+10	0.724748*10^-6	2.64588ms

图 1: 输出结果，第二列为所有逆深度的误差之和，第三项为 problem 中的 solve(20) 用时输出, 将原代码中特征点和相机数量增加了一些