**第6章作业**

**用户名：MagicTZ**

**时间：11.08.2020**

2 Bundle Adjustment

2.1 文献阅读

我们在第五讲中已经介绍了 Bundle Adjustment，指明它可以⽤于解 PnP 问题。现在，我们又在后端中说明了它可以⽤于解⼤规模的三维重构问题，但在实时 SLAM 场合往往需要控制规模。事实上， BundleAdjustment 的历史远⽐我们想象的要长。请阅读 Bill Triggs 的经典论⽂ Bundle Adjustment: A Modern Synthesis（见 paper/⽬录 )，了解 BA 的发展历史，然后回答下列问题：

1. ***为何说Bundle Adjustment is slow是不对的?***

Ans: 因为BA算法具有稀疏性的特性, 我们可以利用矩阵的稀疏性来加速问题的求解, 例如通过Marginalization(Schur消元). 而且在实际的问题当中, 也需要考虑问题本身, 如果在观测中, 仅需要优化相机位姿, 站点坐标仅仅作为约束, 那么问题的复杂性又会大大减小, BA的求解也会加快.

1. ***BA中有哪些需要注意参数化的地方? Pose和Point各有哪些参数化的方式? 有何优缺点.***

Ans:

* BA需要注意参数化的地方: 相机内参, 相机外参(位姿), 3维点坐标, 投影点坐标以及其他的一些约束条件;
* 参数化的方式:
  + Pose
    - 旋转矩阵(角轴) + 平移向量 (优点: 直观; 缺点: 有其他约束, 如det(R)=1, RT\*R=I)
    - 四元数 (优点: 紧凑, 无奇异性”Singularity”, 只需要4个维度就可以实现绕任意原点的旋转; 缺点:不够直观, 需要转换成其他形式)
    - 变换矩阵 (缺点: 自由度冗余 优点: 描述简单)
    - 欧拉角+平移向量 (优点: 紧凑; 缺点: 万向锁”Gimbal Lock”问题 ,丢失了一个自由度)
  + Point
    - Homogeneous (齐次坐标): 形式如(X,Y,Z,1), 可以通过齐次坐标以有限远点坐标来表示无穷远点
    - 非齐次坐标: 形式如(X,Y,Z)
    - 像素坐标与图像坐标与世界坐标

1. ***本文写于2000年, 但是文中提到的很多内容在后面十几年的研究中得到了印证. 你能看到哪些方向在后续工作中有所体现? 请举例说明.***

Ans:

* 3.4 的Intensity-based methods就是直接法
* 5中的Network Structure是后面的图优化

2.2 BAL-dataset

BAL（Bundle Adjustment in large）数据集（http://grail.cs.washington.edu/projects/bal/）是⼀个⼤型 BA 数据集，它提供了相机与点初始值与观测，你可以⽤它们进⾏ Bundle Adjustment。现在，请你使⽤ g2o，⾃⼰定义 Vertex 和 Edge（不要使⽤⾃带的顶点类型，也不要像本书例程那边调⽤ Ceres来求导），书写 BAL 上的 BA 程序。你可以挑选其中⼀个数据，运⾏你的 BA，并给出优化后的点云图。本题不提供代码框架，请独⽴完成。

提⽰：注意 BAL 的投影模型⽐教材中介绍的多了个负号；

* BAL数据集格式

<相机位姿数量> <三维点数量> <观测值数量(Edge)>

<相机索引(点)> <三维点索引(点)> <x\_obs> <y\_obs> (x\_obs, y\_obs是归一化坐标)

...格式如上(数量=Edge)

...

<相机参数> (格式是9\*1列向量,如下)

R (3\*1, 罗德里格向量)

T (3\*1)

f (fx = fy)

k1

k2

...格式如上 (数量=相机位姿数)

...

<三维点坐标> (3\*1)

...

...

代码:

附在code文件夹中

结果图:



Figure 2.1 部分的迭代结果

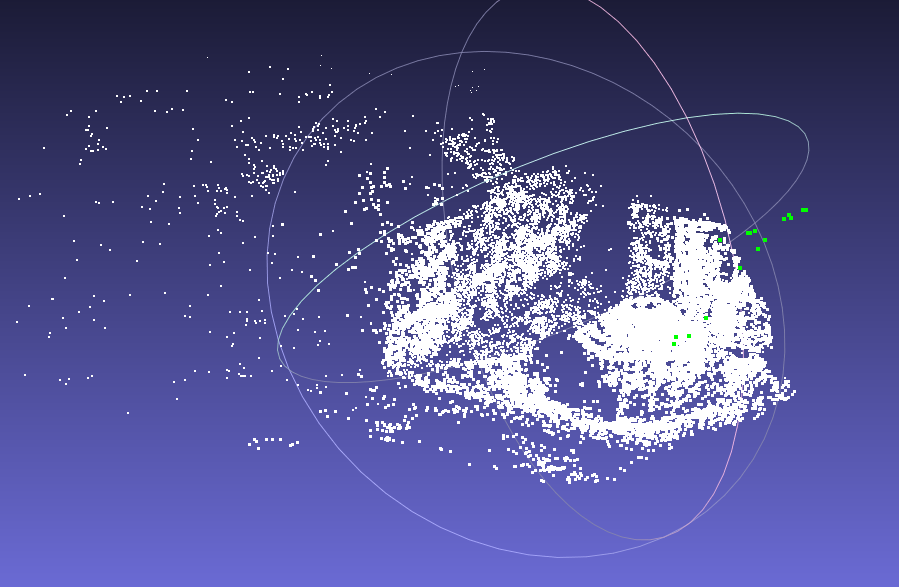
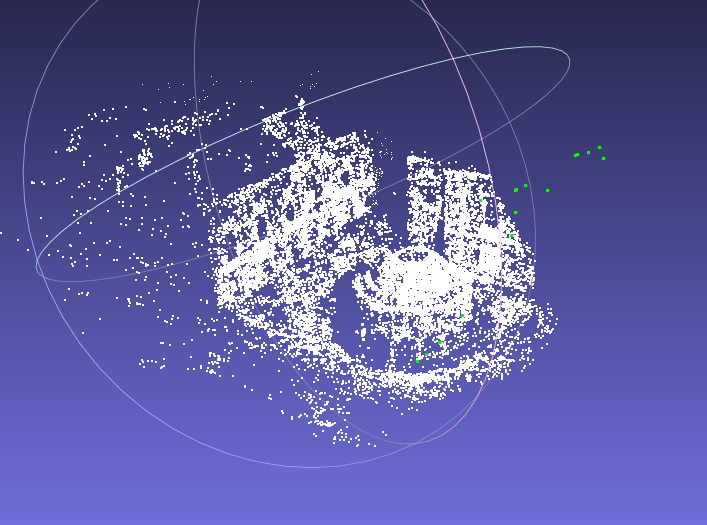
 

Figure 2.2 优化前后对比 (Left: BA前; Right: BA后)

在实现的过程中, 需要注意以下几个方面(都是坑):

* BAL文件的与众不同, 投影模型与正常的相比需要乘以-1, 注意格式
* 自己定义BAL类参数的读取. private内部参数在public中通过函数读取, 不能直接进行外部调用, 什么时候用const 也必须要注意.
* 最后将得到的点通过写入ply文件利用MeshLab进行显示的时候, ply文件的格式也要格外注意, 否则无法正常显示
* Normalize() 使用的是slambook2中的部分函数, 也可以自己定义

**3 直接法的Bundle Adjustment**

3.1 数学模型

特征点法的 BA 以最⼩化重投影误差作为优化⽬标。相对的，如果我们以最⼩化光度误差为⽬标，就得到了直接法的 BA。之前我们在直接法 VO 中，谈到了如何⽤直接法去估计相机姿。但是直接法亦可⽤来处理整个 Bundle Adjustment。下⾯，请你推导直接法 BA 的数学模型，并完成它的 g2o 实现。注意本题使⽤的参数化形式与实际的直接法还有⼀点不同，我们⽤ x; y; z 参数化每⼀个 3D 点，⽽实际的直接法多采⽤逆深度参数化 [1]。

本题给定 7 张图⽚，记为 0.png ⾄ 6.png，每张图⽚对应的相机位姿初始值为 Ti，以 Tcw 形式存储在 poses.txt ⽂件中，其中每⼀⾏代表⼀个相机的位姿，格式如之前作业那样：

平移在前，旋转（四元数形式）在后. 同时，还存在⼀个 3D 点集 P，共 N 个点。其中每⼀个点的初始坐标记作 pi = [x; y; z]T i 。每个点还有⾃⼰的固定灰度值，我们⽤ 16 个数来描述，这 16 个数为该点周围 4x4的⼩块读数，记作 I(p)i，顺序见图 1 。换句话说，⼩块从 u − 2; v − 2 取到 u + 1; v + 1，先迭代 v。那么，我们知道，可以把每个点投影到每个图像中，然后再看投影后点周围⼩块与原始的 4x4 ⼩块有多⼤差异。那么，整体优化⽬标函数为：

即最⼩化任意点在任意图像中投影与其本⾝颜⾊之差。其中 K 为相机内参（在程序内以全局变量形式给定）， 为投影函数， W 指代整个 patch. 下面，请回答：

1. 如何描述任意⼀点投影在任意⼀图像中形成的 error？

Ans:

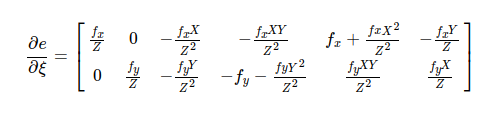
1. 每个 error 关联⼏个优化变量？

Ans:

* 三维点坐标(X,Y,Z)
* 相机的变换矩阵(3个旋转分量+3个平移分量)

1. error 关于各变量的雅可⽐是什么？

Ans:



3.2 实现

下面, 请你根据上述说明, 使用g2o实现上述优化, 并用pangolin绘制优化结果. 程序框架见code/directBA.cpp 文件. 实现过程中, 思考并回答以下问题:

提⽰：

1. 构建 Error 之前先要判断点是否在图像中，去除⼀部分边界的点。

2. 优化之后， Pangolin 绘制的轨迹与地图如图 1 所⽰。

3. 你也可以不提供雅可⽐的计算过程，让 g2o ⾃⼰计算⼀个数值雅可⽐。

4. 以上数据实际取⾃ DSO[1]。

1. 能否不要以的形式参数化每个点?

Ans: 可以的, 逆深度就是一种形式

1. 取4\*4的patch好吗? 取更大的patch好还是取小一点的patch好?

Ans: 应该是比较适中的, 如果太大反而不太好, 计算量会增大, 同时约束也比较弱, 容易造成误匹配, 太小的话会对较大的运动比较敏感.

1. 从本题中, 你看到直接法与特征点法在BA阶段有何不同?

Ans: 基于的误差方程不同,一个基于重投影误差, 一个基于光度误差. 其他的部分基本一致.

1. 由于图像的差异, 你可能需要鲁棒核函数, 例如Huber. 此时Huber的阈值如何选取?

Ans: 这个不是特别清楚, 准备查一下相关的文献资料.

原始图像:

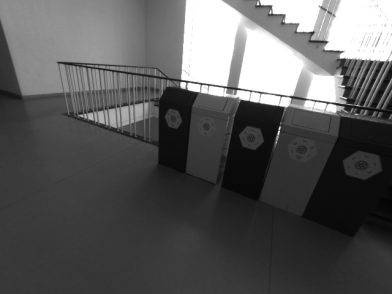
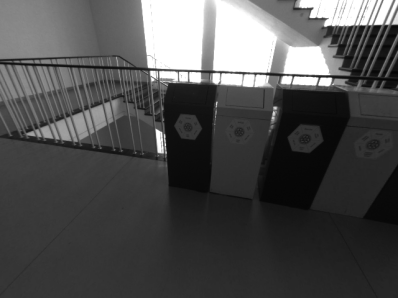
  



Figure 3.1 相机在7个不同位置拍摄的图像 (位姿存储在poses.txt文件中)

结果图:

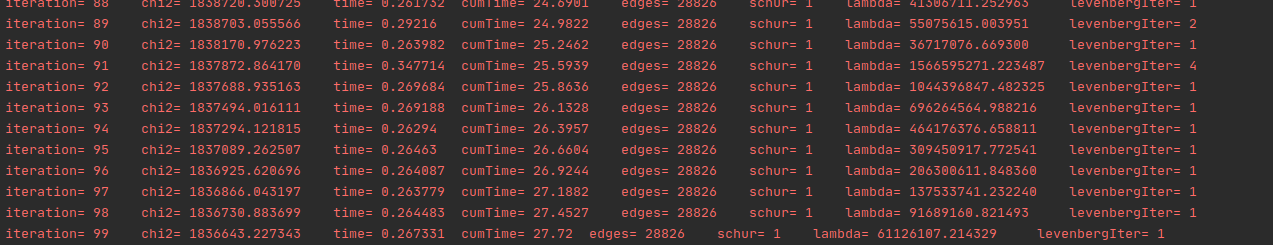


Figure 3.2 迭代100次后的结果

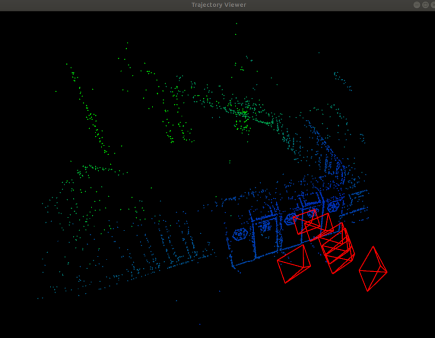
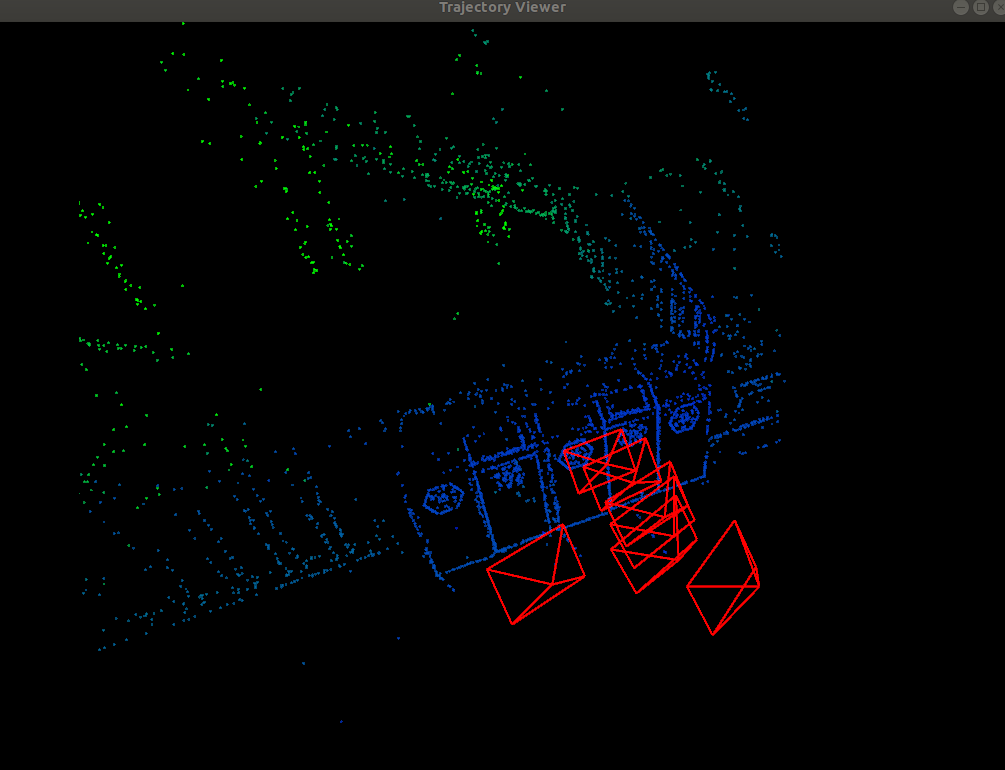
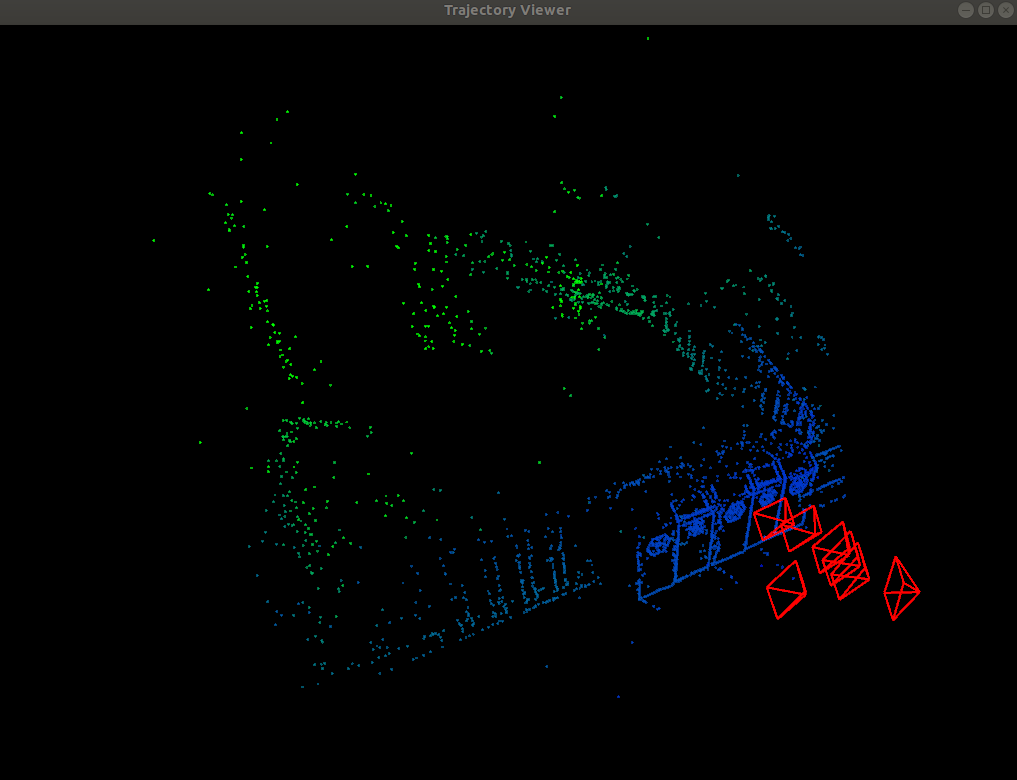
  

Figure 3.3 直接法BA的三维轨迹图, 红色是相机位姿,

颜色从蓝到绿表示深度差异, 越靠近蓝色离相机越近

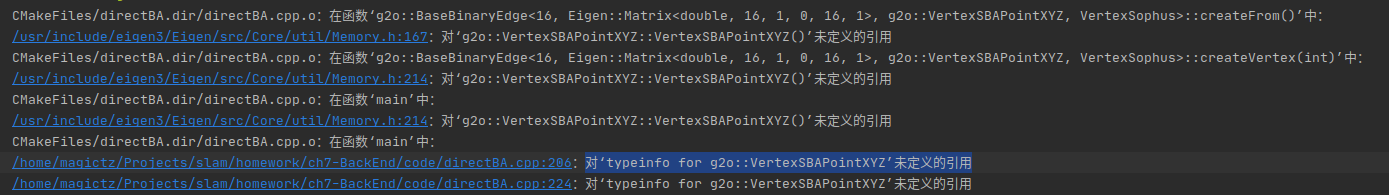
(Left: 没有迭代; Mid: iteration = 100; Right: iteration = 200)

实验的整体过程: 该实验给定了相机初始位姿, 三维路标点文件, 相机内参. 首先通过相机的变换矩阵将三维点从世界坐标系转换到相机坐标系下, 然后利用内参再转到像素坐标系, 然后可以通过双线性内插得到像素值, 与观测数据对比, 得到最小光度误差, 然后进行BA迭代求解.

通过观测上面的结果, 发现BA以后迭代对图像的影响不是特别大, 我个人觉得应该是因为数据量太小, 数据点比较少的缘故, 从而看起来影响并不是特别大, 但是从量化数据看, 确实是在减少的, 后面准备再尝试其他的数据集进行实验对比.

问题:

1. 虽然程序写完了,但是编译的时候遇到了这个问题,不知道是什么原因导致的.



解决方式: 提示显示是g2o::VertexSBAPointXYZ这个类没有成功定义, 但是这个类是g2o自己定义的, 既然他自己定义的出问题了, 那我直接另外定义了一个相同结构的顶点类PointVertex, 作用完全一样, 编译一下就通过了.

1. 新版的g2o在使用块构造器Block\_solver的时候, 由于用的是unique\_ptr, 不支持拷贝构造, 所以需要使用std::move来进行传参

具体解决方式: <https://github.com/gaoxiang12/slambook/issues/88>