

Alvin 第四次作业

第一问：补充的三角化的代码如下：

确定 P1 与 P2 矩阵的值：

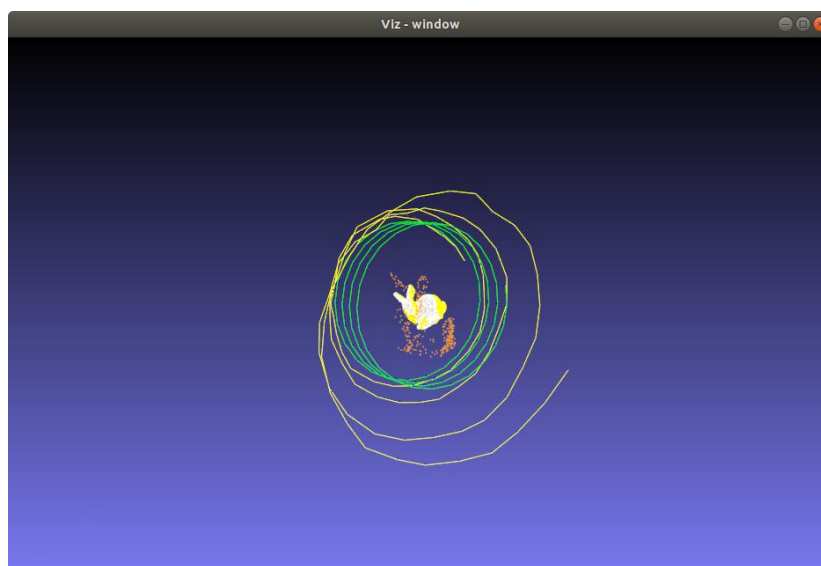
```
Eigen::Matrix3d Rtemp= frame_curr.Twc_.block(0,0,3,3).transpose()*frame_last.Twc_.block(0,0,3,3);
Eigen::Vector3d
ttemp=frame_curr.Twc_.block(0,0,3,3).transpose()*(frame_last.Twc_.block(0,3,3,1)-frame_curr.Twc_.block(0,3,3,1));
cv::eigen2cv(Rtemp, R21);
cv::eigen2cv(ttemp, t21);
R21.copyTo(P2.rowRange(0,3).colRange(0,3));
t21.copyTo(P2.rowRange(0,3).col(3));
P2 = cv_K*P2;
```

其中 P2 矩阵所使用 R, t 为相对于 last_frame 的相对位姿。

使用 TwoViewGeometry 中的三角化函数进行三角化：

```
cv::Point2f point_last, point_curr;
point_curr = cv::Point2f(frame_curr.fts_[idx_curr][0], frame_curr.fts_[idx_curr][1]);
point_last = cv::Point2f(frame_last.fts_[idx_last][0], frame_last.fts_[idx_last][1]);
TwoViewGeometry::Triangulate(point_last, point_curr, P1, P2, p3d_c1);
```

运行所得结果如下图所示：



观察可知基本恢复了运动轨迹，但是对比真值明显有比较大的失真。使用 rpe 评估误差可得：

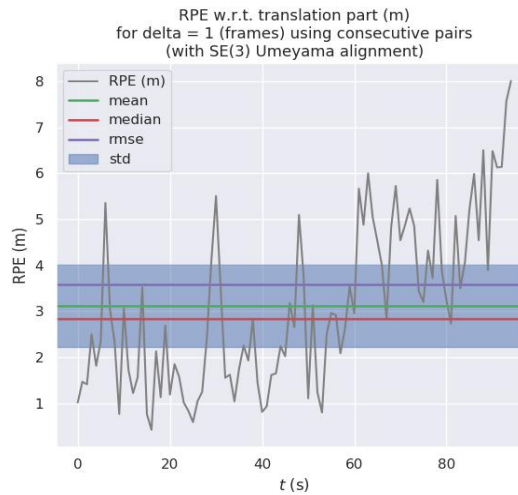
```

jie@5510: ~/Desktop/shenlan/orbslam
File Edit View Search Terminal Help
^C
jie@5510:~/Desktop/shenlan/orbslam
_rpe tum -a frame_traj_gt.txt fram
RPE w.r.t. translation part (m)
for delta = 1 (frames) using conse
(with SE(3) Umeyama alignment)

      max      8.008096
      mean     3.117861
      median    2.832176
      min      0.426418
      rmse     3.591660
      sse      1225.501879
      std      1.782965

jie@5510:~/Desktop/shenlan/orbslam

```



观察可以发现其 rmse 值较大，且 RPE 随时间增加有逐渐发散趋势。

第二问：

所需补全的 g2o 的代码 ,主要功能为向 g2o 所提供的 optimizer 中添加顶点(mappoint,pose)和与其中连接顶点的边的关系。

部分代码如下：

添加顶点两个相机的 pose：

```

//add last pose to vertex
g2o::VertexSE3Expmap * poselast = new g2o::VertexSE3Expmap();
poselast->setEstimate(Converter::toSE3Quat(last_Tcw));
poselast->setId(0);
poselast->setFixed(true);
optimizer.addVertex(poselast);
//add current pose to vertex

g2o::VertexSE3Expmap * posecurr = new g2o::VertexSE3Expmap();
posecurr->setEstimate(Converter::toSE3Quat(curr_Tcw));
posecurr->setId(1);
posecurr->setFixed(false);
optimizer.addVertex(posecurr);

```

添加顶点所有 mappoint 的 pose：

```

g2o::VertexSBAPointXYZ * vPoint = new g2o::VertexSBAPointXYZ();
vPoint->setId(map_index);
poselast->setFixed(true);
vPoint->setEstimate(mpt);
vPoint->setMarginalized(true); // seperately optimize mappoint and frame
optimizer.addVertex(vPoint);

```

添加边 last_pose 和 mappoint(第二个 pose 添加基本相同)：

```

g2o::EdgeSE3ProjectXYZ* edge_last = new g2o::EdgeSE3ProjectXYZ();
edge_last->setVertex(0,dynamic_cast<g2o::OptimizableGraph::Vertex*>(optimizer.vertex(map_index)));
edge_last->setVertex(1,dynamic_cast<g2o::OptimizableGraph::Vertex*>(optimizer.vertex(0)));
edge_last->setMeasurement(Eigen::Vector2d(features_last[idx_last][0],features_last[idx_last][1]));
edge_last->setInformation(Eigen::Matrix2d::Identity());

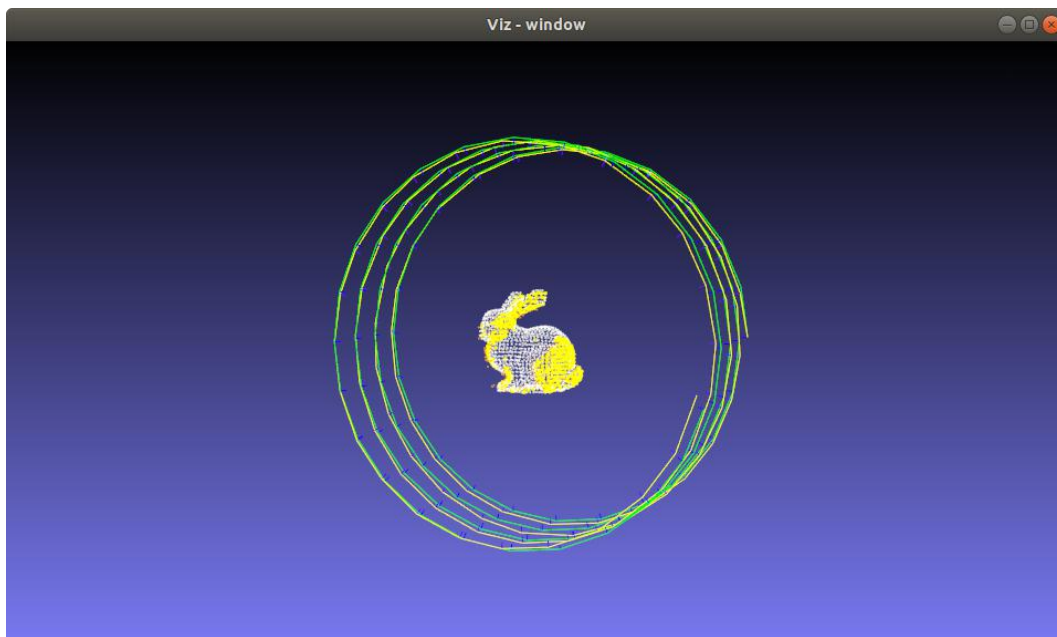
if(bRobust)
{
    g2o::RobustKernelHuber* rk1 = new g2o::RobustKernelHuber;
    edge_last->setRobustKernel(rk1);
    rk1->setDelta(thHuber2D);
}

edge_last->fx = fx;
edge_last->fy = fy;
edge_last->cx = cx;
edge_last->cy = cy;

optimizer.addEdge(edge_last);

```

使用 BA 优化后的结果如下所示：



使用 evo_rpe 评估轨迹的结果如下所示：

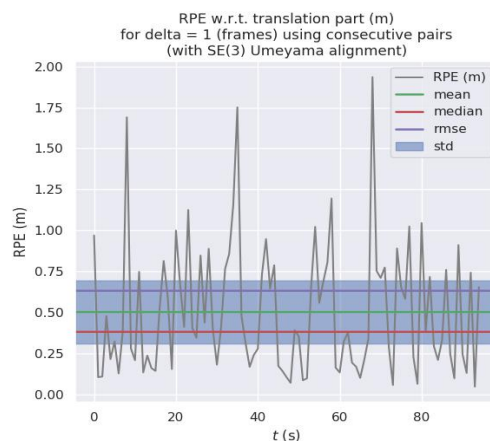
```

jie@5510:~/Desktop/shenlan/orbslan
m -a frame_traj_gt.txt frame_traj
RPE w.r.t. translation part (m)
for delta = 1 (frames) using conse
(with SE(3) Umeyama alignment)

      max      1.935652
     mean      0.502689
    median      0.382502
      min      0.046701
      rmse      0.634978
       sse     38.303758
       std      0.387945

jie@5510:~/Desktop/shenlan/orbslan

```



可以观察到：优化后所得轨迹其 rmse 明显优于 BA 优化之前。且 RPE 误差的值没有明显的随时间发散的趋势。

第三问：

在 `feature_match` 加入 `outliner_rate = 0.05` 后，slam 程序无法完整运行提示 vtk 内存崩溃。