Course6_homework_front_end

0. Summary

1. 证明

1. 证明式(15)中，取 $\mathbf{y}=\mathbf{u}_4$ 是该问题的最优解。提示：设 $\mathbf{y}\prime=\mathbf{u}_4+\mathbf{v}$，其中 $\mathbf{v}$ 正交于 $\mathbf{u}_4$，证明

$$\mathbf{y}\prime^{\top}\mathbf{D}^{\top}\mathbf{D}\mathbf{y}\prime \geq \mathbf{y}^{\top}\mathbf{D}^{\top}\mathbf{D}\mathbf{y}$$

该方法基于奇异值构造矩阵零空间的理论。

- 由于 $\mathbf{D}\in\mathbb{R}^{2n\times 4}$，在观测次于大于等于两次时，很可能 $\mathbf{D}$ 满秩，无零空间。
- 寻找最小二乘解：

$$\min_{\mathbf{y}} \|\mathbf{D}\mathbf{y}\|_2^2, \quad s.t. \|\mathbf{y}\|=1 \tag{14}$$

解法：对 $\mathbf{D}^{\top}\mathbf{D}$ 进行 SVD：

$$\mathbf{D}^{\top}\mathbf{D}=\sum_{i=1}^{4}\sigma_i^2\mathbf{u}_i\mathbf{u}_j^{\top} \tag{15}$$

其中 σ_i 为奇异值，且由大到小排列，$\mathbf{u}_i,\mathbf{u}_j$ 正交。

证明：

SVD分解中 σ_i 沿对角线从大到小排列

$$\begin{aligned}
&(\mathbf{u}_4+\mathbf{v})^{\top}\mathbf{D}^{\top}\mathbf{D}(\mathbf{u}_4+\mathbf{v})\\
&=\sum_{i=1}^{4}\sigma_i^2\ (\mathbf{u}_4+\mathbf{v})^{\top}\mathbf{u}_i\mathbf{u}_i^{\top}(\mathbf{u}_4+\mathbf{v})\\
&=\sum_{i=1}^{4}\sigma_i^2\ (\mathbf{u}_4^{\top}+\mathbf{v}^{\top})\ \mathbf{u}_i\mathbf{u}_i^{\top}(\mathbf{u}_4+\mathbf{v})\\
&=\sum_{i=1}^{4}\sigma_i^2\ (\mathbf{u}_4^{\top}\mathbf{u}_i+\mathbf{v}^{\top}\mathbf{u}_i)\ (\mathbf{u}_i^{\top}\mathbf{u}_4+\mathbf{u}_i^{\top}\mathbf{v})
\end{aligned}$$

SVD 分解中,u1~u4 构成完整的空间，且互相正交；

若v=$\mathbf{u}_1$，根据对称矩阵的svd分解的特征向量的正交特性，

可得：上式等于如下：

$$\sigma_1^2 + \sigma_4^2$$

可知 $\mathbf{y}^\top \mathbf{D}^\top \mathbf{D}\mathbf{y} = \sigma_4^2$ (已知y的最优解为u4)

所以 $\sigma_1^2 + \sigma_4^2 \geq \sigma_4^2$

其余的v=u2,或u3,等可以类似推理；

$\mathbf{y}\prime^\top \mathbf{D}^\top \mathbf{D}\mathbf{y}\prime \geq \mathbf{y}^\top \mathbf{D}^\top \mathbf{D}\mathbf{y}$ 得到证明；

所以y是最优解

2. Code：三角化

Code：

triangulate.cpp

```
62      Eigen::Vector3d P_est;          // 结果保存到这个变量
63      P_est.setZero();
64      /* your code begin */
65      int D_rows = 2 * (end_frame_id - start_frame_id);
66      Eigen::MatrixXd D = Eigen::MatrixXd::Zero(D_rows, cols: 4);
67      for (int i = start_frame_id; i < end_frame_id;++i)
68      {
69
70          Eigen::Matrix3d Rcw = camera_pose[i].Rwc.transpose();
71          Eigen::Vector3d tcw = -Rcw * camera_pose[i].twc;
72          D.block( startRow: 2 * (i - start_frame_id), startCol: 0, blockRows: 1, blockCols: 3) .noalias()
73                  = camera_pose[i].uv( index: 0) * Rcw.block( startRow: 2, startCol: 0, blockRows: 1, blockCols: 3)
74                    - Rcw.block( startRow: 0, startCol: 0, blockRows: 1, blockCols: 3);
75          D.block( startRow: 2 * (i - start_frame_id), startCol: 3, blockRows: 1, blockCols: 1).noalias() =
76                  camera_pose[i].uv[0] * tcw.segment( start: 2, n: 1) - tcw.segment( start: 0, n: 1);
77          D.block( startRow: 2 * (i - start_frame_id) + 1, startCol: 0, blockRows: 1, blockCols: 3).noalias()=
78                  camera_pose[i].uv( index: 1) * Rcw.block( startRow: 2, startCol: 0, blockRows: 1, blockCols: 3) - Rcw.block( startRow: 1, startCol: 0, blockRows: 1, blockCols: 3);
79          D.block( startRow: 2 * (i - start_frame_id) + 1, startCol: 3, blockRows: 1, blockCols: 1).noalias()
80                  = camera_pose[i].uv( index: 1) * tcw.segment( start: 2, n: 1) - tcw.segment( start: 1, n: 1);
```

main

```
    }
    Eigen::JacobiSVD<Eigen::MatrixXd> svd(
            D.transpose() * D, computationOptions: Eigen::ComputeThinU | Eigen::ComputeThinV);
    Eigen::Vector4d lamda = svd.singularValues();
    std::cout << "奇异值 = " << lamda.transpose() << std::endl;
    if(lamda( index: 2)/lamda( index: 3)<1e-3){
        std::cout << "The parallax is not enough! " << std::endl;
        return -1;
    }

    Eigen::Vector4d u4 = svd.matrixU().block( startRow: 0, startCol: 3, blockRows: 4, blockCols: 1);
    if(u4( index: 3)!=0 && u4( index: 2)/u4( index: 3)>0){
        P_est( index: 0) = u4( index: 0) / u4( index: 3);
        P_est( index: 1) = u4( index: 1) / u4( index: 3);
        P_est( index: 2) = u4( index: 2) / u4( index: 3);
    }


    /* your code end */
```

运行结果：

```
(base) ep@ep-VirtualBox:/media/sf__xcloud/notes/vslam_vio/lesson_doc/course6_hw$ ./cmake-build-debug/estimate_depth
奇异值 =     468.406     7.74642    0.723255 5.30104e-16
ground truth:
  -2.9477 -0.330799   8.43792
your result:
  -2.9477 -0.330799   8.43792
```

code 中计算结果还需要除齐次坐标的最后一个元素的值，才得到正确结果；