深蓝学院
shenlanxueyuan.com

视觉SLAM理论与实践

第五次作业讲评

主讲人 田智豪

2 ORB 特征点

程序注意的部分:

1. 图像边界
2. atan2和tan函数的区别
3. 编译 / 运行 的错误信息要大概阅读下

1. 为什么说 ORB 是一种二进制特征?
2. 为什么在匹配时使用 50 作为阈值,取更大或更小值会怎么样?
3. 暴力匹配在你的机器上表现如何?你能想到什么减少计算量的匹配方法吗?

3 从 E 恢复 R, t

深蓝学院 shenlanxueyuan.com

```
Eigen::JacobiSVD<Eigen::Matrix3d> svd(E, Eigen::ComputeFullU | Eigen::ComputeFullV);
Eigen::Matrix3d U = svd.matrixU();
Eigen::Matrix3d V = svd.matrixV();
Eigen::Vector3d S = svd.singularValues();
S(0) = (S(0)+S(1)) /2;
S(1) = (S(0)+S(1)) /2;
S(2) = 0;

// compute
Eigen::Matrix3d R_positive, R_negative;
R_positive = Eigen::AngleAxisd(0.5 * M_PI, Eigen::Vector3d::UnitZ()).toRotationMatrix();
R_negative = Eigen::AngleAxisd(-0.5 * M_PI, Eigen::Vector3d::UnitZ()).toRotationMatrix();
// END YOUR CODE HERE

// set t1, t2, R1, R2
// START YOUR CODE HERE
Matrix3d t_wedge1;
Matrix3d t_wedge2;
t_wedge1 = U * R_positive * S.asDiagonal() * U.transpose();
t_wedge2 = U * R_negative * S.asDiagonal() * U.transpose();
Matrix3d R1;
Matrix3d R2;
R1 = U * R_positive.transpose() * V.transpose();
R2 = U * R_negative.transpose() * V.transpose();
```

4 用 G-N 实现 BA

问答题:

1. 如何定义重投影误差?
2. 该误差关于自变量的雅可比矩阵是什么?
3. 解出更新量之后,如何更新至之前的估计上?

4 用 G-N 实现 BA

程序:

1. 读取数据: std::ifstream, std:getline
2. 高斯牛顿法迭代优化位姿

```
Vector3d p3d_quote = T_esti*p3d[i];
Vector2d p2d_esti;
p2d_esti << fx * p3d_quote(0) / p3d_quote(2) + cx, fy * p3d_quote(1) / p3d_quote(2) + cy;
//p2d_esti = {fx * p3d_quote(0) / p3d_quote(2) + cx, fy * p3d_quote(1) / p3d_quote(2) + cy};
Vector2d e = p2d[i] - p2d_esti;

// END YOUR CODE HERE

// compute jacobian
Matrix<double, 2, 6> J;
// START YOUR CODE HERE
J(0, 0) = -fx / p3d_quote(2);
J(0, 1) = 0;
J(0, 2) = fx * p3d_quote(0) / p3d_quote(2) / p3d_quote(2);
J(0, 3) = fx * p3d_quote(0) * p3d_quote(1) / p3d_quote(2) / p3d_quote(2);
J(0, 4) = -fx - fx * p3d_quote(0) * p3d_quote(0) / p3d_quote(2) / p3d_quote(2);
J(0, 5) = fx * p3d_quote(1) / p3d_quote(2);

J(1, 0) = 0;
J(1, 1) = -fy / p3d_quote(2);
J(1, 2) = fy * p3d_quote(1) / p3d_quote(2) / p3d_quote(2);
J(1, 3) = fy + fy * p3d_quote(1) * p3d_quote(1) / p3d_quote(2) / p3d_quote(2);
J(1, 4) = -fy * p3d_quote(0) * p3d_quote(1) / p3d_quote(2) / p3d_quote(2);
J(1, 5) = -fy * p3d_quote(0) / p3d_quote(2);
```

5 ICP 实现轨迹对齐

1. 读取数据
2. 根据书上的ICP程序求解
3. 画图

深蓝学院
shenlanxueyuan.com

感谢各位聆听

Thanks for Listening