三维点群数据配准:ICP 程序说明文档

徐远东 10848347

一、ICP 算法描述:

开始：

Step 1：在两组点群数据中选择一组,标记为{P},另一组标记为{Q}
在 P 中随机选取 control number = 3000 个控制点
初始化平移向量和旋转矩阵

Step 2：在 Q 中通过遍历搜索寻找控制点的 3000 个最近邻对应点,并返回误差(距离平方和). 如果误差变化足够小,表示算法收敛,结束算法

Step 3：对两组对应点去除质心,使用四元数表示和最小二乘法计算单步旋转矩阵 R

Step 4：计算单步平移向量 T = Qcenter – R*Pcenter

Step 5：对控制点进行旋转和平移进行位置更新
New_control_points = R * control_points + T

Step 6：更新总体旋转矩阵和平移向量 Rn+1 = R * Rn ；Tn+1 = R * Tn + T

返回执行 Step 2。

二、总体程序执行过程:

1. 从两个.obj 文件读取点群数据
2. 执行 ICP 算法
3. 将计算的旋转平移变换应用到原始数据,输出变换后的点群,保存在.obj 文件中

三、文件说明：

main.cpp	主程序
ICP.h	ICP 算法类头文件
ICP.cpp	ICP 算法类定义文件
1.obj 2.obj	输入点群数据文件
out500.obj out1000.obj	输出点群数据(500 个控制点和 1000 个控制点)

编译环境：Ubuntu 8.10 gcc

依赖包：newmat 矩阵计算函数包

四、补充说明：

1. 使用查老师讲义中 SVD 分解计算最小二乘问题时，有一错误的公式：

原公式：

```
A   = [   Di          (di-di')T
          di'-di           0     ]
```

应修订为：

```
A   = [   0            di'-di
          (di-di')T       Di     ]
```

其中，T 表示转置，Di 为 di'+di 的 Um 形式。

2. 一个遗漏但计算时需要用到的公式：

四元数 q = [q0,q1,q2,q3]与旋转矩阵 R 的关系：

$$R = \begin{bmatrix} q0^2+q1^2-q2^2-q3^2 & 2(-q0q3+q1q2) & 2(q0q2+q1q3) \\ 2(q0q3+q1q2) & q0^2-q1^2+q2^2-q3^2 & 2(-q0q1+q2q3) \\ 2(-q0q2+q1q3) & 2(q0q1+q2q3) & q0^2-q1^2-q2^2+q3^2 \end{bmatrix}$$

五、实验结果：

1000 个控制点的输出结果：

```
read two clouds of points from obj files
initial error = 27.4745
move clouds of control points to their correspond points center
get transform matrix
iterate times = 0
error = 9.0149
delta = 0.0184596
move clouds of control points to their correspond points center
get transform matrix
iterate times = 1
error = 5.08016
delta = 0.00393474
move clouds of control points to their correspond points center
get transform matrix
iterate times = 2
error = 3.71525
delta = 0.00136491
move clouds of control points to their correspond points center
get transform matrix
iterate times = 3
error = 3.07798
delta = 0.00063727
move clouds of control points to their correspond points center
get transform matrix
iterate times = 4
error = 2.72393
delta = 0.000354045
move clouds of control points to their correspond points center
get transform matrix
iterate times = 5
error = 2.53751
delta = 0.000186419
move clouds of control points to their correspond points center
get transform matrix
iterate times = 6
error = 2.43728
delta = 0.000100235
```

```
move clouds of control points to their correspond points center
get transform matrix
iterate times = 7
error = 2.38297
delta = 5.43144e-05
Rotate Matrix =
0.968346 -5.57294e-05 0.249613
0.0105885 0.999109 -0.0408539
-0.249388 0.0422038 0.967484
transform to all data in P
output clouds of points P after transform
```

500 个控制点的输出结果：

```
read two clouds of points from obj files
initial error = 16.0209
move clouds of control points to their correspond points center
get transform matrix
iterate times = 0
error = 4.22259
delta = 0.0235966
move clouds of control points to their correspond points center
get transform matrix
iterate times = 1
error = 2.14591
delta = 0.00415335
move clouds of control points to their correspond points center
get transform matrix
iterate times = 2
error = 1.48321
delta = 0.00132539
move clouds of control points to their correspond points center
get transform matrix
iterate times = 3
error = 1.19862
delta = 0.000569192
move clouds of control points to their correspond points center
get transform matrix
iterate times = 4
error = 1.06904
delta = 0.000259147
move clouds of control points to their correspond points center
get transform matrix
iterate times = 5
error = 1.00911
```

```
delta = 0.000119872
move clouds of control points to their correspond points center
get transform matrix
iterate times = 6
error = 0.978404
delta = 6.14058e-05
Rotate Matrix =
0.966339 0.00378418 0.257246
0.00565488 0.999338 -0.035943
-0.257212 0.0361878 0.965677
transform to all data in P
output clouds of points P after transform
```