# 人体轮廓提取算法改进测试报告

## Changelog

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 变更人 | 变更说明 | 变更时间 |
| V1.0 | 张琛 | 初稿 | 2015/06/28 |

目录

[人体轮廓提取算法改进测试报告 1](#_Toc423286447)

[Changelog 1](#_Toc423286448)

[一、 目的 1](#_Toc423286449)

[二、 算法改进描述 2](#_Toc423286450)

[2.1. 分离地面算法改进 2](#_Toc423286453)

[2.2. 区域增长时间性能改进 4](#_Toc423286454)

[2.3. 前景蒙板分割步骤改进（图1步骤D） 4](#_Toc423286455)

[三、 算法局限性/缺陷分析 7](#_Toc423286456)

[3.1. 算法逻辑方面 7](#_Toc423286458)

[3.2. 代码实现方面 10](#_Toc423286459)

[四、 参考文献 10](#_Toc423286460)

## 目的

对报告[1]中算法的一些部分进行改进，测试结果，并分析目前算法的局限性。



图 1 报告[1]中的算法流程图

## 算法改进描述



### 分离地面算法改进

原始的分离地面滤波方法，采用卷积核{1,1,1, -1,-1,-1}’对深度图进行卷积，再使用经验阈值th进行阈值化操作。

当场景中地面占较大深度区域时，因为深度相机误差，导致**地面像素由远到近呈“弧面”**，如图2红框所示。进而导致滤波结果矩阵中，靠近屏幕的地面值较小（因为不够水平）。

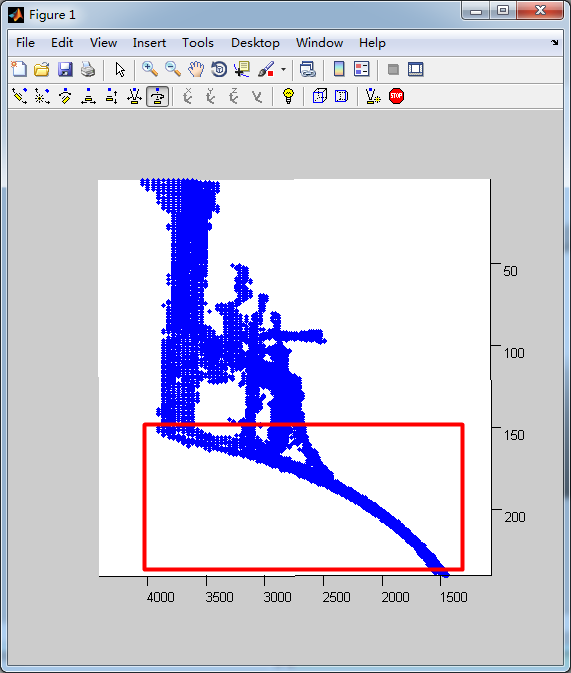


图 2 深度相机误差导致地面呈弧形

此时，若th较大（宽松），则近处的人体腿部下边缘与地面仍然相连，如图3-a红框所示；若th较小（严格），则手部与身体深度差稍大的区域也会被分割，导致后续增长截断，失败，如图3-b红圈所示。

a) th=130 b) th=90

图 3 分离地面滤波二值化结果

改进的处理办法为：

|  |
| --- |
| foreach ***row*** in ***dmat***:  ***row*** \*= (0.1 \* i)^3; //i为行序号  end  threshold(***dmat***) |

即对每一行像素，乘以(0.1 \* i)^3，然后进行阈值化，即可得到很好的去除地面结果，如图4所示。其中，0.1是为了防止数值太大溢出，丢失精度。



图 4 改进的地面除去结果，蒙板颜色与图3相反，黑色为地面

### 区域增长时间性能改进

trackingNoMove 接口中，第一步是使用相邻两帧深度变化较小的点集{S}作为种子点集，进行区域增长。{S}往往是密集的。

之前的代码实现中，将{S}中每个点作为种子点进行增长，过于频繁的队列push/pop操作，导致区域增长过程效率较低。特别是去除墙面背景的过程中，因为对粗估的墙面点集，使用了区域增长获取精确墙面背景，稠密墙面点集{S}导致效率明显下降。

改进后的实现为：对于，若其邻域点，则直接标记为前景点，并不push到种子点队列；否则，将入队。这样，种子点队列的长度大大减小，效率明显提升（320\*240单帧处理时间由6~9ms减少为3~6ms）

### 前景蒙板分割步骤改进（图1步骤D）

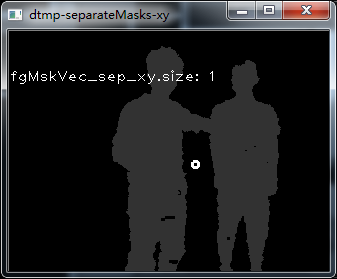
之前的实现中，处理方案是：若XY视角（正视图）前景蒙板包含多个轮廓，则分割，这对人体与某物体（椅子、另一人，etc.）在X、Y方向分离时能正确处理，如图5所示。但对①两人前后部分遮挡的场景，当两人在Z方向分离时，无法正确分割（图6-a）；②人体被细长隔板分为上下两部分时，会错误地分割识别为两个前景（图7-a）。

改进后的处理方案为： 将trackingNoMove 得到的轮廓对应深度图，转到top-down-view（Z轴压缩比==区域增长阈值55mm），若其中包含多个轮廓，则将这几个轮廓分别反投影到XY视角，得到正确分离的多个轮廓，如图6-b, 6-c, 图7-b所示。但此方案的缺点为：对图5所示的场景（手从椅子上**竖直抬起**），仍然判定为同一个前景，无法正确分割。

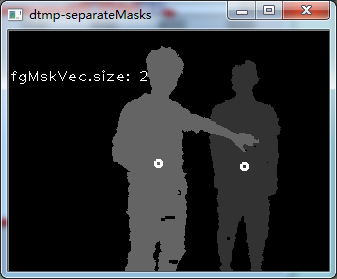
   

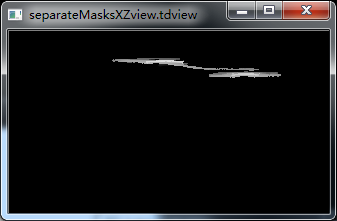
图 5 报告[1]中2.7小节图示



a) 在XY视角分割，因只有一个轮廓，故分割失败

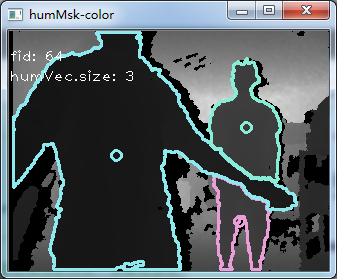


b) 转到XZ视角，可以正确分割为两个轮廓

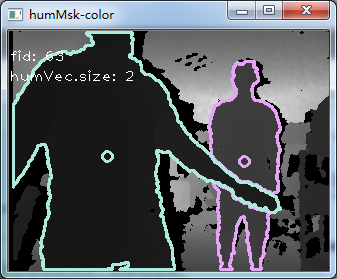


c) 子图b)对应的XZ视角top-down-view

图 6 原始前景蒙板分割结果（a）及改进后结果（b, c）



a) 人体被横隔为上下两部分时，原分割算法将两部分识别为两个前景



b) 改进分割算法，对人体被横隔为几部分的情境，正确地将这几部分识别为一个前景

图 7 两人前后部分遮挡时，原始前景蒙板分割结果（a）及改进后结果（b）

## 算法局限性/缺陷分析



### 算法逻辑方面

#### 区域增长问题

a) 区域增长算法的代码使自己实现的，本质上是洪泛法，当人体与某物体靠近太近时，两者会**瞬间增长成为一个轮廓**，实际场景中可能存在一些错误：

① 人靠近墙面时，首先被错误地增长为背景，故检测不到人体，如图8；

② 两人靠近时，瞬间增长为同一个前景轮廓，如图9；

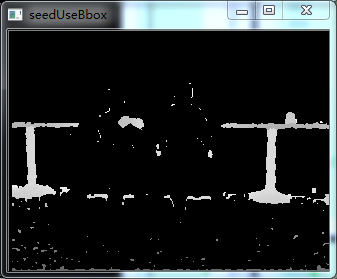
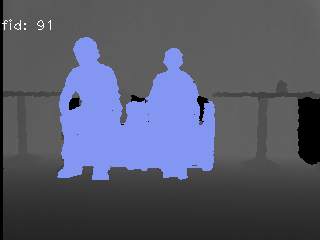
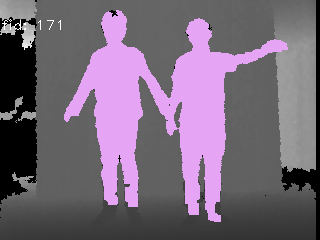


图 8 人坐在沙发上，靠墙太近，被墙面区域增长为背景

a)报告[1]-2.5小节 b)报告[1]-2.6小节

图 9 报告[1]中两人合并为同一前景轮廓示例

可能的改进方案： 每检测到一个前景轮廓（初始化），为其独立地维护一个背景模型，并在此后对各个前景轮廓单独更新背景模型。

b) 区域增长阈值为固定经验阈值（rgThresh=55），当手臂与躯干相隔较远，且无平滑过渡区域时，手臂区域无法被增长成为前景。

此问题暂无改进思路。

c) 若手臂横举，自头顶自上而下移动，起始一段时间内，头部区域无法增长为前景，如图10所示。用top-down-view处理方法仍无法解决此问题。

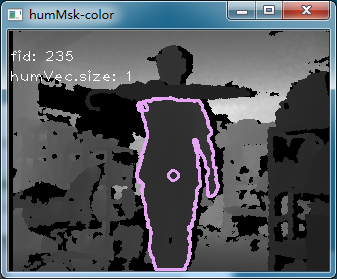
 

图 10 头部区域无法增长为前景

#### 前景蒙板分割步骤改进（图1步骤D）问题

如2.3小节所述，前景蒙板分割步骤，改进后能够正确处理两物体在XZ视角分离的场景，而无法正确处理两物体上下分离的场景，如图5所示。

此步骤作为区域增长的后处理步骤，在目前的区域增长算法下，无法兼顾图5、图7两种情境，暂无改进方案。

#### 种子点生成算法问题

目前的种子点算法，当场景中存在较小狭长区域时，会因为轮廓满足条件被检出为前景，如图11所示。这说明仅依靠“狭长区域”特征的种子点算法不够稳定。

可能的改进方案：考虑将“狭长区域”特征，结合头部（圆形区域）模板匹配，进行初始化前景检测。

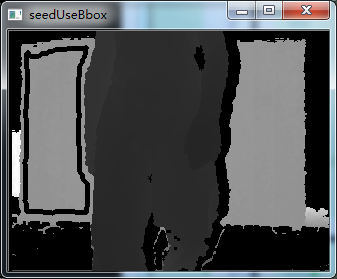
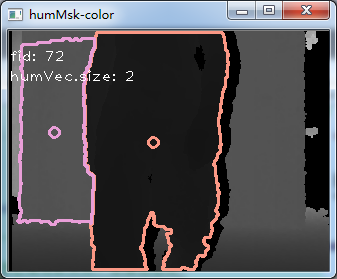
 

图 11 错误的种子点导致将背景误检为前景

### 代码实现方面

a) 目前自己实现的区域增长算法，对于稠密种子点集的情形，接口效率仍然不够高（3~6ms）；而整个流程中区域增长耗时占比约50%（使用vs-profiler测试）。因为opencv也有一个类似实现（cv::floodFill），可以考虑使用opencv的基本实现做包装实现，进行效率对比。

## 参考文献

[1] 采用跟踪算法的人体轮廓提取测试报告, 张琛