

综合作业 3

2017011534 王思程 自 71

1.myslic.m

为自己写的 SLIC 超像素分割函数

```
function [L,N] = myslic(img, k, m, n_turn)
```

```
% img:输入图像
```

```
% k:希望分成多少个超像素
```

```
% m:取代Nc的常数值，一般在[1, 40]
```

```
% n_turn:迭代几次
```

```
%
```

```
% L:返回的标签图，值为1：N
```

```
% N: 最终的超像素个数
```

1.1 所用数据

$S = \sqrt{N/k}$ 然后用 ceil 函数向上取整。算出行和列上的格数，向下取整。

建一个 $k \times 6$ 的矩阵 C，1: 3 记录平均 Lab 值，4,5 记录质心的 x,y，6 记录这个超像素有多少个像素点。

L 记各个标签

d 存某个像素到最近的超像素中心的距离

1.2 初值化中心点

没有按论文所说找 3×3 里梯度最小地方，作用不大，第一次迭代更新就把这个效果抵消很多。

直接就切分成那么多格子，取他们的中点当初始超像素中心点。

由于之前 S 是向上取整，所以保证切出来的格子数小于 k。然后这样最下面有一部分不能凑成完整的格子的像素，把它们归到上一行的超像素里。

1.3 迭代

1.3.1 查询赋值 d

对每个超像素中心，在 $2S \times 2S$ 的区域里，计算每个像素和超像素中心的距离，如果距离小于改像素的 d 值，则将该像素 d 值赋值为此距离，并将此像素的标签改为对应的超像素的编号。

距离的算法是：

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2},$$
$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2},$$

$$D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2}$$

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2}.$$

1.3.2 更新 C 矩阵的值

更新超像素的平均 Lab 值，质心 x,y 的值，包含的像素的个数

1.4 清除孤立像素

看 L 矩阵，对每一个标签，周围 8 个像素中同标签的个数小于等于 3 个就将其赋值为相邻的其它标签值。

1.5 重新整理 L 和 C，得到 N

因为可能一些超像素中含的像素数已经是零了，就是说有些标签已经不存在了，需要清除包含像素数 0 的超像素，并让标签是从 1 到 N 中心没有跳数的。

先遍历一遍得到每个超像素以前的超像素有多少个是不含像素了的，再遍历便把对应的 L 中标签减去对应个数即可。

2. 显示中间过程

在 GUI 中，没有专门写函数封装功能而在 `myslic` 函数的基础上更改直接把代码粘过去。就是每次迭代一轮结束后，根据当前的 L，用 `boundarymask` 得到边界，将边界和原图一起显示在 axes 上，然后在用 `scatter` 函数把超像素点中心也画上去就 OK 了。

3. 抠图

3.1 选前背景

利用的是 `impoly` 函数，在图片上选择前背景的点即可。为了区分，我用 `line` 函数用不同颜色的线将点连起来了。将点坐标转化为 `index`，就可以放在 `lazy snapping` 里了。

3.2 抠图

利用 `lasysnapping` 得到前景的 `mask`，注意这个 `mask` 是个二维的，而彩色图像是三维的，要在 `axis=3` 上将这个 `mask` 叠三层，然后用在原图上即可

4. 尝试不同超像素数，分析分割性能和速度的影响。

GUI 中可选 `m` 值，超像素希望个数 `k`，迭代次数，可方便尝试不同的参数。

`M` 值越大越靠位置分割，`m` 值越小越靠颜色分割。

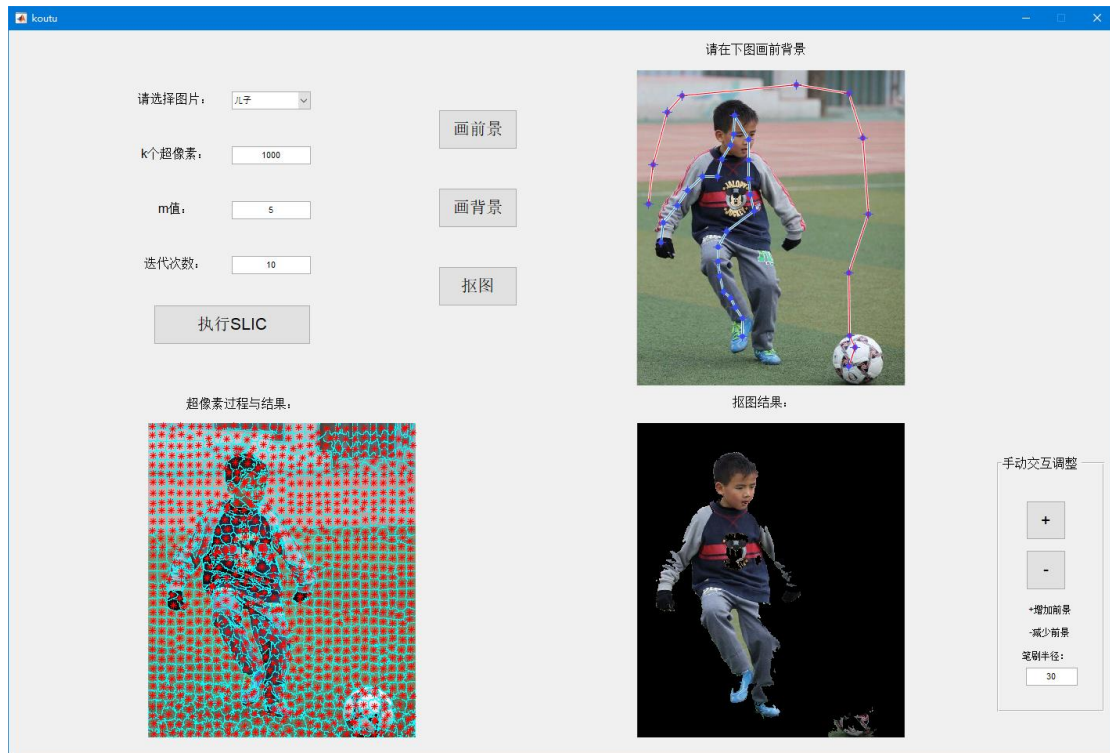
显然迭代次数越少越快，迭代次数一般 10 此就会有较好的效果。

超像素个数 k 太小分割效果不好，需要试出一个合适的值(k 不可选过大，否则会因为选取 S 和行列步数时的取整导致程序报错!!)。SLIC 复杂度与 k 无关，是 $O(N)$ 量级（此处 N 是图片的像素数），所以 k 不影响速度。

$K=30$:



$K=1000$

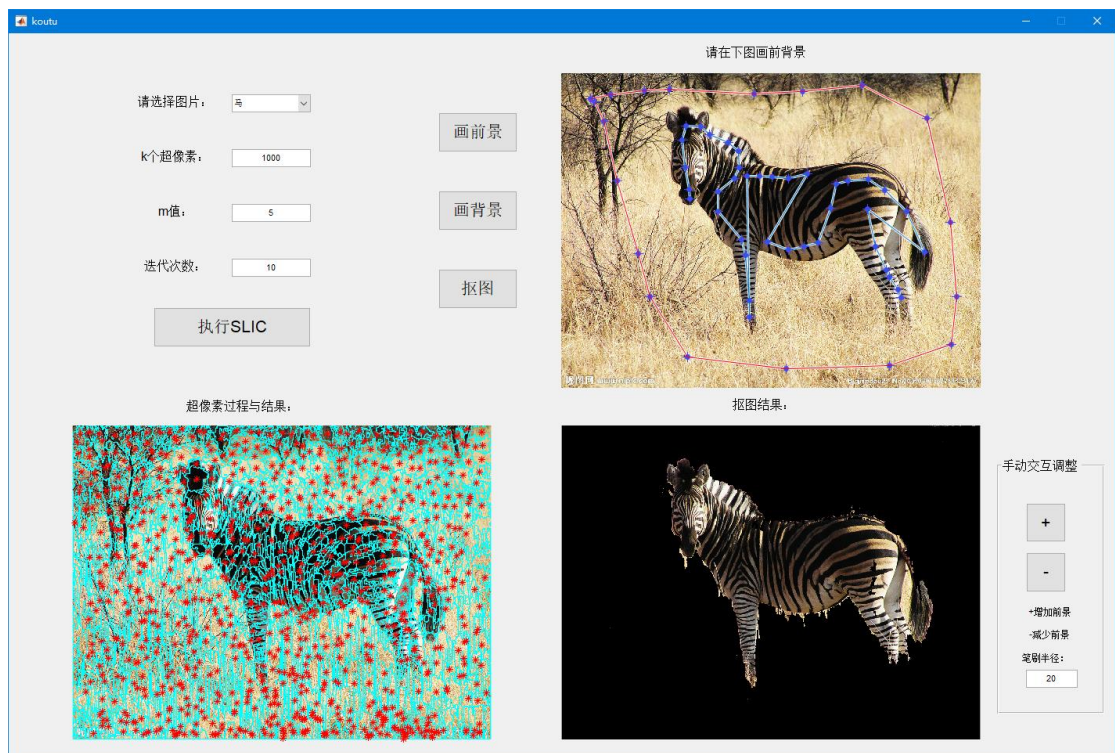
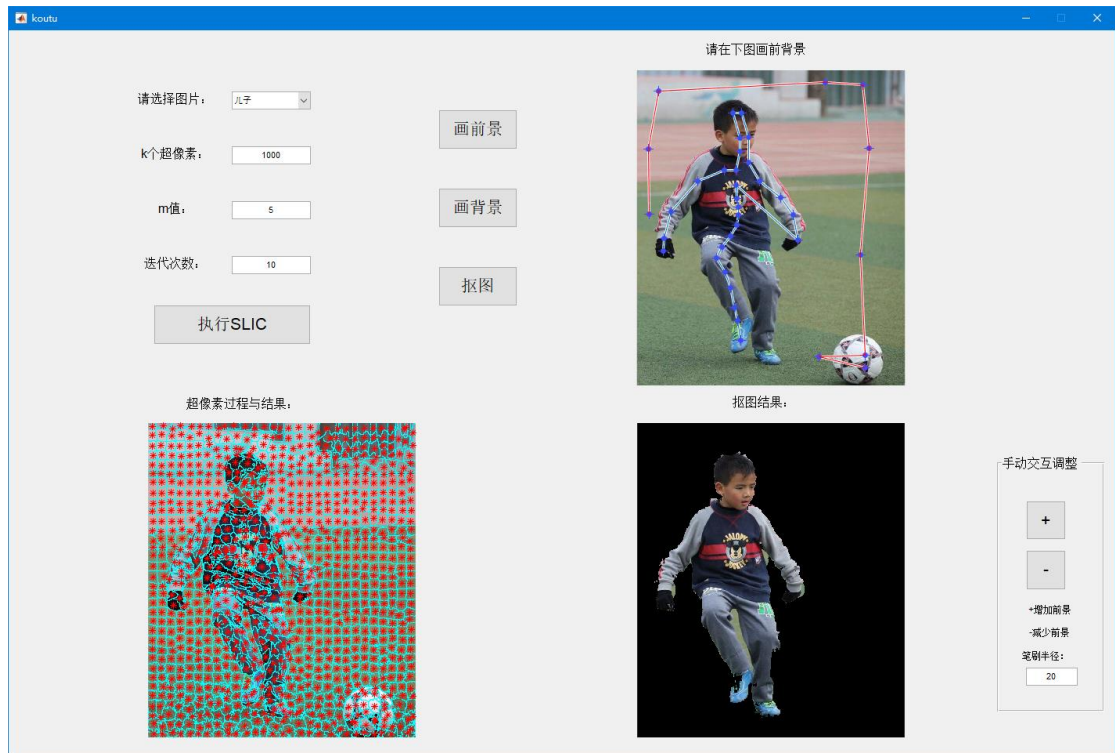


5.改进分割

在局部区域经行交互式分割，可以选择笔刷大小调整最终结果。

点+号后在需要加入前景的地方点点，点完点后回车即可。 -号同理。

6.效果



请选择图片：

1

k个超像素：

1000

m值：

5

迭代次数：

10

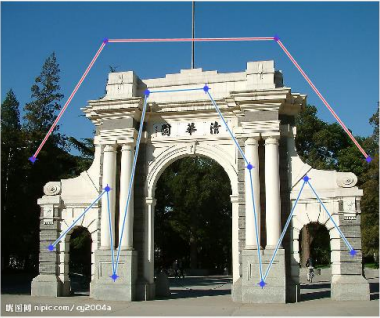
执行SLIC

画前景


画背景

抠图

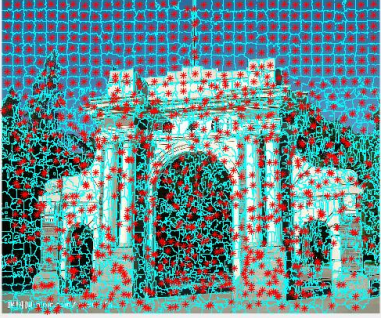
请在下图画前景



抠图结果：



超像素过程与结果：



手动交互调整

+

-

+增加前景

-减少前景

笔刷半径：

20