

# 图像分割

班级：自 73

姓名：陈昱宏

## 一、任务描述：

要求实现 SLIC 算法，并将 SLIC 算法的聚类结果、用户选择的点和原图当作 lazysnapping 函数的输入，实现图像分割。

此外，要求显示 SLIC 算法的中间过程，并尝试改进参数和其他方式，对 SLIC 算法进行改进。

## 二、实验步骤：

根据任务要求，设计了以下实验步骤：

- （一）根据论文给出的算法结构，自己完成 SLIC 算法的设计。
- （二）调整 SLIC 算法中的两个重要参数 K 和 M，观察不同参数的分割结果。
- （三）根据论文中的建议，修改距离计算函数，观察改进版的 ASLIC 算法的性能。
- （四）在 SLIC 算法的基础上，结合库函数 lazysnapping，制作用户交互的图像切割界面。

## 三、算法设计思路：

### （一）SLIC 算法思路：

论文中给出了具体的算法，伪代码如下：

**Algorithm 1.** SLIC superpixel segmentation

```

/* Initialization */
Initialize cluster centers  $C_k = [l_k, a_k, b_k, x_k, y_k]^T$  by sampling
pixels at regular grid steps  $S$ .
Move cluster centers to the lowest gradient position in a  $3 \times 3$ 
neighborhood.
Set label  $l(i) = -1$  for each pixel  $i$ .
Set distance  $d(i) = \infty$  for each pixel  $i$ .

repeat
/* Assignment */
for each cluster center  $C_k$  do
for each pixel  $i$  in a  $2S \times 2S$  region around  $C_k$  do
Compute the distance  $D$  between  $C_k$  and  $i$ .
if  $D < d(i)$  then
set  $d(i) = D$ 
set  $l(i) = k$ 
end if
end for
end for
/* Update */
Compute new cluster centers.
Compute residual error  $E$ .
until  $E \leq \text{threshold}$ 

```

SLIC 算法要求的图像是在 Lab 空间，其中距离函数由两个部分组成，分别为颜色距离和像素距离：

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

综合起来，合成一个距离计算式：

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

用这个距离算法来进行聚类，并更新聚类中心，论文中指出，最多 10 次即可收敛，因此我的算法设定十次循环便结束。

## (二) 改进算法——ASLIC

这个改进方法是在论文中给出的，主要改进的是距离计算方

式，将距离计算方式改成以下形式：

$$D = \sqrt{\left(\frac{d_c}{m_c}\right)^2 + \left(\frac{d_s}{m_s}\right)^2}$$

上式中的 $m_c$ 和 $m_s$ 是上一次迭代过程中，该类的最大颜色距离和空间距离，因此在原先的 SLIC 算法，另外记录一下每次循环中，该类出现的最大颜色距离和最大空间距离。

### （三）GUI 界面设计：

这一部分的设计步骤是先将图像根据用户选择的  $k$  和  $m$  值进行 SLIC 超像素分割，并且在 GUI 界面中显示 SLIC 算法的中间结果（包括边界和中心），得到分类图之后，用户可以选择前景和背景，进行分割，而且可以让用户再次重新选择和补充前背景。

四、实验结果：

(一) 图片一 (Picture1.m)：

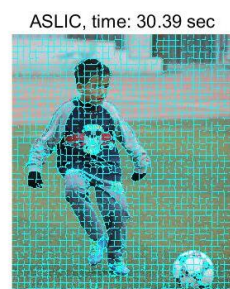
改变 K 值：



改变 m 值：

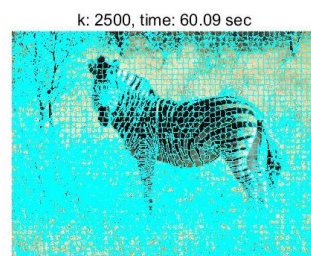
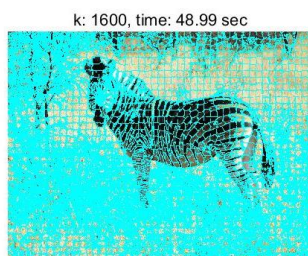
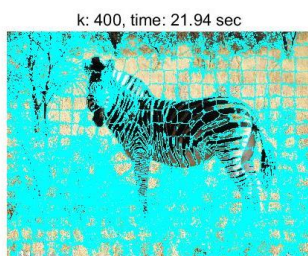
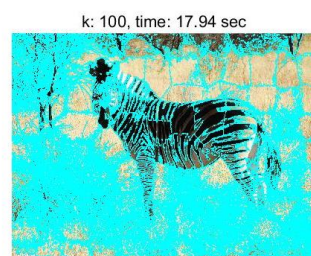
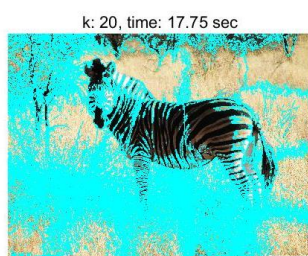
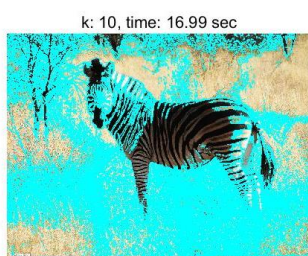


普通算法和改进算法的对比：



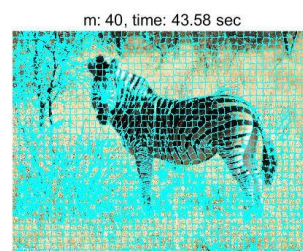
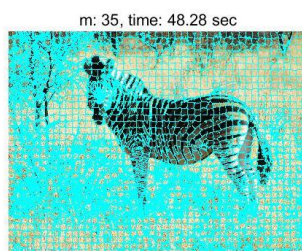
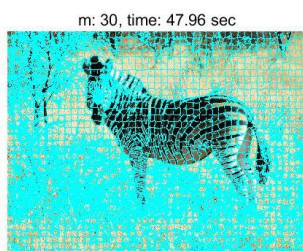
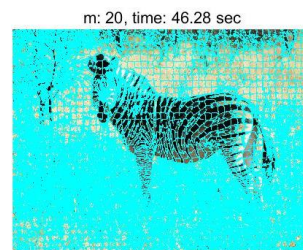
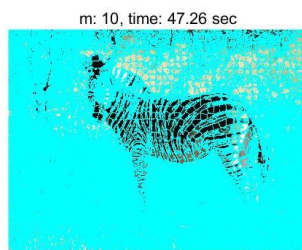
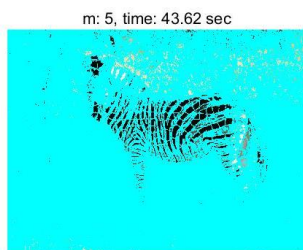
## (二) 图片二 (Picture2.m):

改变 K 值:

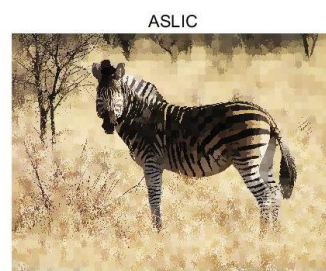
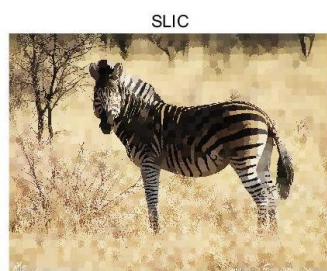
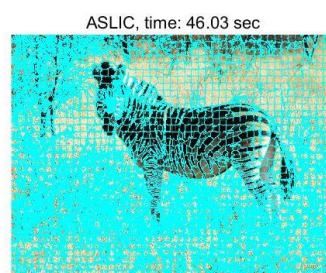
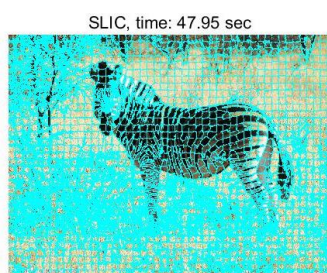


改变 m 值:



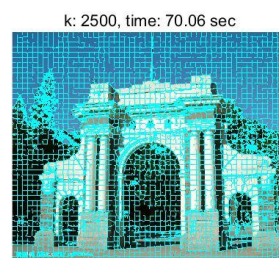
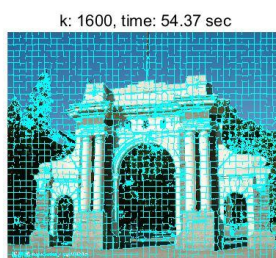
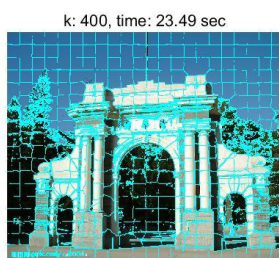
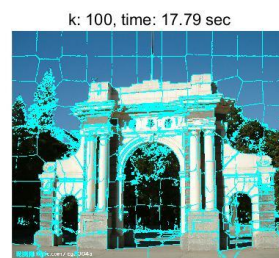
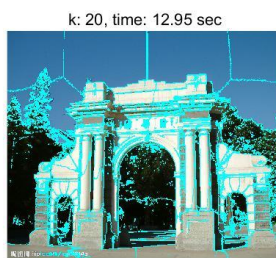
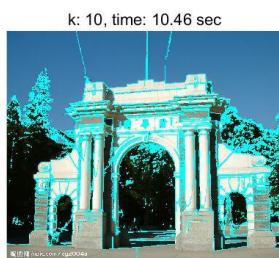


普通算法和改进算法的对比：

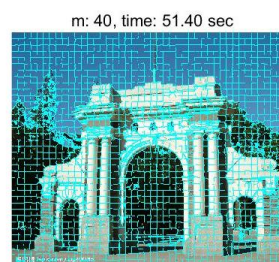
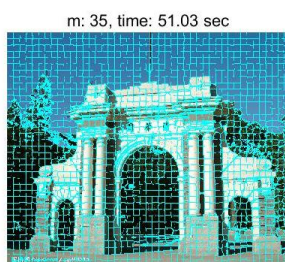
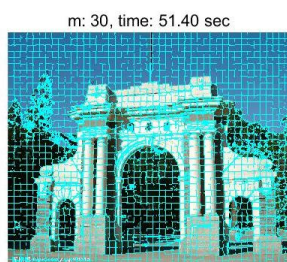
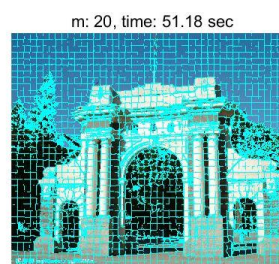
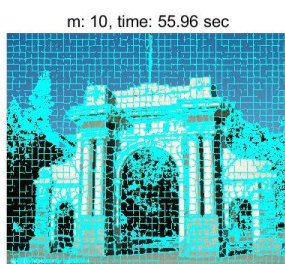
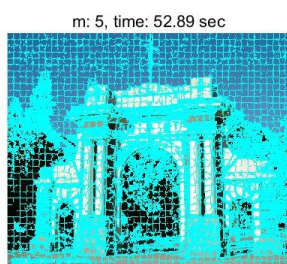


(三) 图片三 (Picture3.m)：

改变 K 值：

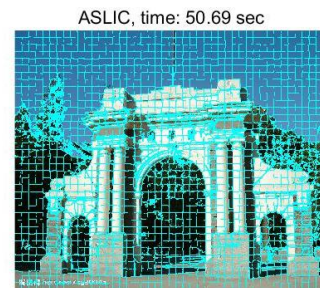
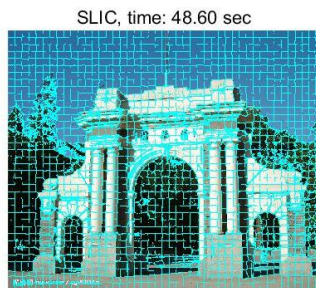


改变 m 值:



普通算法和改进算法的对比:





(四) 图像分割 (以图片一为范例):

GUI 界面如下



五、效果分析:

改变 K 值代表改变聚类的类数, 观察实验结果, 可以看到 K 值较小的时候, 每一类的像素较多, 分出来的块较大, 符合理论结果。而也可以看到, 当 K 值较大的时候, 所需要的计算时间较多,

这一点好像并不符合 matlab 库内的 SLIC 算法，可能造成的原因是因为我在设计 SLIC 算法的时候，都是以循环的方式来写，所以耗时比较久。

改变  $m$  值代表的是改变颜色距离和空间距离在距离表达式中的占比，当  $m$  值较大的时候，空间距离占比大，所以分出来的类更趋近于方形；当  $m$  值较小的时候，颜色距离占比大，所以分出来的类更趋近于图像边缘，这一部分在实验结果上比较符合。

改进算法和原本的算法，在论文中说到，其实原本的 SLIC 算法在性能上表现得比改进的 ASLIC 算法好，但这一部分在实验结果上并没有发现两者有很显著的差距。

## 六、文件目录说明：

folder homework\_2017011507\_陈昱宏

-----folder data（题目所需图片）

-----Picture1.m（图片一脚本）

-----Picture2.m（图片二脚本）

-----Picture3.m（图片三脚本）

-----MySLIC.m（SLIC 算法函数）

-----MyASLIC.m（ASLIC 算法函数）

-----GUI.m（GUI code）

-----GUI.fig (GUI 图形界面)

-----Picture1.pdf (图片一发布文档)

-----Picture2.pdf (图片二发布文档)

-----Picture3.pdf (图片三发布文档)

-----MySLIC.pdf (SLIC 算法函数发布文档)

-----MyASLIC.pdf (ASLIC 算法函数发布文档)

-----GUI.pdf (GUI code 发布文档)

-----2017011507\_陈昱宏\_大作业 3.pdf (实验报告)