

视频镜头检测

序号	学号	专业班级	姓名	性别
	3140102441	计科 1404	卢涛	男

1. Project Introduction

(1) 选题

编写程序进行视频镜头检测（2 种以上算法）

输入：解码后的图像序列

输出：发生镜头切换处的帧编号

(2) 工作简介

先将视频片段处理成解码后的图像序列，再运用 2 种以上的算法，将这些图像序列进行分析，找出发生镜头切换的帧编号。

(3) 开发环境及系统运行要求

Matlab 2016a

2. Technical Details

(1) 工程实践当中所用到的理论知识阐述

在做镜头检测之前，为方便起见，我们先将一个视频短片提取出一定数量的图像序列。每一个图像对应一帧。镜头边缘检测算法要做的是检测出一段视频片段中发生镜头切换的图像帧。一般在同一个镜头中，相邻的图像帧是相似的，所以可以通过检测相邻图像帧特征的突变来实现。我采用的 3 种算法的原理都是基于判别该帧图像与前后帧图像的差别。若相差很大，即发生突变，也就是该图像对应的帧为镜头变换的帧编号。

(2) 具体的算法

解码视频片段成图像算法：

读入视频，将每一帧处理成 jpg 格式的图片

伪代码：

```
obj = VideoReader('test.avi');%读入视频
for i = 0 to obj.NumberOfFrames %处理每一帧
    Img_I=read(obj,i);
    imwrite(Img_I,[directory,[num2str(i)'.jpg'];]); %输出jpg图片
```

镜头切换检测算法

算法一：绝对帧间差法：比较相邻图像帧像素的亮度和之差，当大于某个阈值时，则判定发生突变。

伪代码：

```
img_i = imread(imagename); %读取图像
img_i_next = imread(nextimgname); %读取下一幅图像
img_diff(i)=norm(double(img_i(:,:,1)-img_i_plus(:,:,1)))+norm(double(img_i(:,:,2)-img_i_plus(:,:,2)))+norm(double(img_i(:,:,3)-img_i_plus(:,:,3))); %计算前后两幅图像的亮度差

Threshold=mean(img_diff)*4; %设定阈值
for i = 0 to totalframe
    if(img_diff(i)>img_diff(i-1)&&img_diff(i)>img_diff(i+1)&&img_diff(i)>Threshold) %若亮度差大于前后图像以及阈值，则判定为切换帧
        print(i) %输出帧编号
```

算法二：感知哈希法：感知哈希法是用于相似图片搜索的一种快速算法。由于镜头检测算法所面向的对象也是成千上百帧的图像帧，所以我尝试将这种算法用于镜头检测。这种算法的原理是先将图片缩小成 8*8 尺寸，64 像素的图片，摒弃不同尺寸、比例带来的图片差异。再将缩小后的图片，转为 64 级灰度。也就是说，所有像素点总共只有 64 种颜色。然后计算 DCT 并计算 64 个值的平均值。DCT 是把图片分解频率聚集和梯状形。最后，计算哈希值，并组合在一起，就构成了一个 64 位的整数，这就是这张图片的指纹。得到指纹以后，就可以对比不同的图片，看看 64 位中有多少位是不一样的。如果不相同的数据位不超过 5，就说明两张图片很相似；如果大于 10，就说明这是两张不同的图片。我采取的阈值是 2。

伪代码：

```
img_i = imread(imagename); %读取图像
img_i_next = imread(nextimgname); %读取下一幅图像
imbw_i=im2bw(rgb2gray(imresize(img_i,[8,8]))); % 计算哈希
imbw_i_plus=im2bw(rgb2gray(imresize(img_i_plus,[8,8]))); %计算下一幅图像的哈希

for j = 0 to 8
    for k = 0 to 8
        if(imbw_i(j,k)~=imbw_i_plus(j,k)) %比较前后两张图像每位是
```

否相同

```
count(i) ++;
for i = 0 to totalframe                                %遍历
    if(count(i)>Threshold)                               %如果count>阈值，说明前后相
差大，判定为切换帧，输出帧编号
```

算法三：图片相关系数法：我们知道，数学上用 相关系数来表示两个随机变量的相关性。我尝试定义两幅图像的相关系数来衡量相邻图像帧的相似性。如果与后一张图像的相关系数小于阈值，则判定该图像是镜头切换帧。

伪代码：

```
img_i = imread(imagename);                            %读取图像
img_i_next = imread(nextimgname);                      %读取下一幅图像
img_sim(i)=corr2(img_i(:, :, 1),img_i_plus(:, :, 1))+corr2(img_i(:, :,
2),img_i_plus(:, :, 2))+corr2(img_i(:, :, 3),img_i_plus(:, :, 3));
img_sim(i)=img_sim(i)/3;                               %使用矩阵的相关系数计算方法计算相似度

Threshold=0.8;                                         %设置阈值

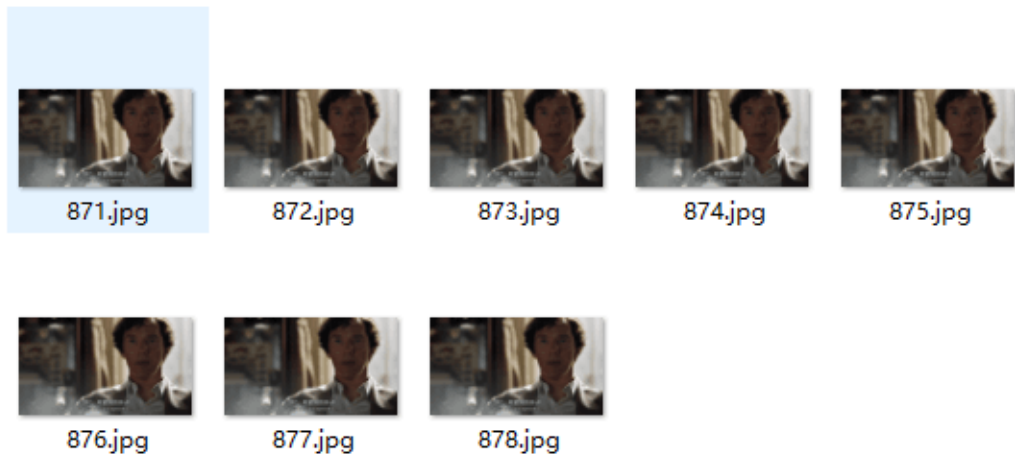
for i = 0 to totalframe                                %遍历
    if(img_sim(i)<Threshold)                             %如果相似度小于阈值，则判断切换帧
    print(i)                                             %输出帧编号
```

(3) 程序开发中重要的技术细节：

整个实验，用到的关于视频处理的库函数都是 matlab 系统自带的，包括 videoreader 读取视频，IMwrite 将每一帧输出 JPG 格式的图片，imread 读取 jpg 格式的图片，norm 计算亮度函数，感知哈希用到的对图像进行缩小、灰度化的函数，以及计算矩阵相似度的 corr2 函数。总的来说，使用的大多数函数都是 matlab 自带的，所以不需要配置太多的库函数，技术上实现也比较简单。

3. Experiment Results

使用的视频片段是时长为 30S 的 avi 格式的视频文件，并且由于字幕会影响图像相似度的判断，所以我事先去掉了字幕。去掉字幕之后，我再将每一帧都处理成图像输出，一共有 878 帧，如下：



这 878 帧中,我先人工判别发生镜头切换的帧编号,分别是 46 102 144 238 259 313 362 394 465 503 632 712 797 858, 一共是 14 帧, 然后, 我再用下面三种算法分别输出发生镜头切换的帧编号, 结果如下:

算法一:

```
命令行窗口
>> edge_check0
46
102
144
238
259
291
302
306
313
362
394
465
503
632
697
712
797
858
fx >>
```

算法一一共输出了 18 个发生镜头切换的帧编号, 其中, 之前判定好的 14 帧全部被输出, 另外 4 帧输出错误, 正确识别率为 77.8%

算法二:

```
命令行窗口
>> edge_check1
102
144
227
230
394
503
632
677
712
797
858
fx >>
```

算法二一共输出了 11 个切换帧编号，其中 8 帧正确识别，6 帧未识别，3 帧识别错误。

算法三:

```
命令行窗口
>> edge_check2
46
102
144
259
313
394
465
503
632
712
797
858
fx >>
```

算法三一共输出了 12 个切换帧的编号，其中，11 帧识别正确，1 帧识别错误，3 帧未识别。

三种算法的分析与比较：

	正确识别	错误识别	未识别
算法一	14	4	0
算法二	8	3	6
算法三	11	1	1

三种算法的相同点：

1、三种算法都含有阈值的确定，而阈值的选择会很大程度上影响识别率，只有选择了合适的阈值，才能有比较高的识别率。从这点上来说，这三种算法都是有待提高的。

2、另外，这三个算法是对渐变的镜头检测的准确率不高，究其原因，是由于特征的定义太过单一，无法完全反应图片的特征。

三种算法的不同点：

1、从速度上来说，算法一比较慢，因为它比较的是图片的亮度，这个提取的过程需要较长时间，而另外两个算法相对来说比较快，因为它们分别是计算哈希和相似度，匹配起来效率比较高。

2、从识别率上来看，算法一和算法三的正确率是可以接受的，并且还算比较高，但是算法二的识别率比较低，分析其原因，我猜想是适用条件的问题。我采用的视频片段可能在亮度上差别较大，而在颜色上差别不大，另外，就是算法二特征属性是整数值，是不连续的，而不同于算法一、三的特征值是实数，是连续的，这可能也是准确率差别的原因。

4. Experience

这个大程，在代码实现上，总体难度不是很大，因为大部分的算法在网上都有介绍，因此很方便就能找到参考代码，并且代码实现也不是很困难，很好去理解。但是，在做的过程中，我还是遇到了几个困难。一个就是顺序读取一个目录下的帧图像的时候，它是按照字典序去读，而不是1 2 3顺序去读，所以导致1后面一帧是10，而不是2，这造成的结果就是3个算法一开始的准确率非常的低，因此帧的顺序乱了。我遇到的另一个困难就是字幕的问题。一开始，我截取的视频片段含有字幕，这会导致两张没有发生镜头切换的帧图像，由于字幕的不同，而导致算法判定其为切换帧，所以一开始算法输出的切换帧包含了很多由于字幕不同而导致的判定错误的帧图像，我分析了很久才发现这一问题，去除字幕后，问题也就迎刃而解了。

总的来说，这个大程让我对视频镜头切换的判定有了更加清楚的认识，对视频切分以及matlab的掌握更进了一步，收获很大。

References:

参考网站：<http://blog.csdn.net/1740450789/article/details/46622493>