**北 京 林 业 大 学**

**2023学年-2024学年第一学期数字视频技术及应用实验指导书**

**专业名称： 数字媒体技术 实验学时： 14学时**

**课程名称： 数字视频技术与应用 任课教师： 刘文萍**

**实验题目： 关键帧提取与分割**

**实验环境： VS + OpenCV**

**实验性质： 综合性设计实验**

**实验内容及要求：**

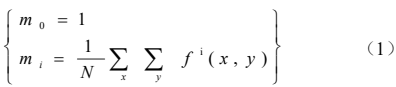
阅读实验一指导书，提取视频关键帧（镜头的第一帧为关键帧），然后分别利用分水岭算法和阈值分割算法对关键帧进行图像分割。 其中，至少任选两种不同的阈值分割算法进行实验。

实验内容指导：

1. 实验步骤：
2. 在VS 中新建工程。
3. 配置 OpenCV。
4. 读取视频。
5. 计算前后两帧灰度值之差，若大于某阈值，则将该帧作为关键帧。
6. 对关键帧采用阈值分割算法或分水岭算法（OpenCV中有封装好的分割函数，但建议按照原理自己实现），生成分割图像。
7. 显示原视频、关键帧和分割后结果。
8. **阈值分割算法（不限于以下两种）：**
9. 矩不变法阈值分割原理：

分割前图像与分割后图像的前三阶矩保持不变。

分割前图像第 i 阶矩 mi 的定义：



分割后图像前三阶矩mi'的定义：

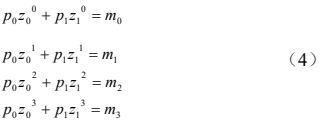


其中f(x,y)为图像在(x,y)的灰度级，pj 表示灰度级为 zj 的概率， 高于阈值的灰度级为 z1，低于阈值的灰度级为 z0。

因此要保持图像分割前后前三阶矩不变则要满足：



设所选阈值为 k ，将高于阈值的灰度用 z1 表示，低于 k 的灰度表示，带入公式 1,2,3 可得公式 4：



因为图像分割前的前三阶矩是可以直接求出的，因此根据公式 4 可得 p1 关于 m1，m2，m3 的关系式。我们可选择阈值k 使其为p0 分位，即

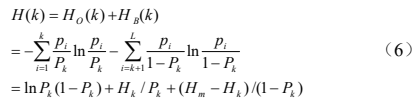


根据公式 5 可求出每个假设的阈值 k 对应的 p0 和 p1，因此使得 p0 和 p1 差最小相对应的阈值 k 则为最终要求的图像分割阈值threshold。

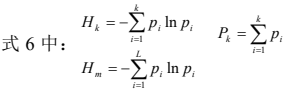
算法步骤:

1. 根据矩不变原理推出p1关于m1，m2，m3的关系式，以保证图像分割前后前三阶矩相等。
2. 假设每个灰度值为阈值k，计算相对应的p0和p1及它们的差值p0-p1
3. 遍历每个灰度值，并求得差值p0-p1最小时对应的灰度值，该灰度值则为分割图像的阈值threshold
4. 遍历灰度图像的像素点，将其与阈值threshold比较。若某点像素值小于阈值threshold，则将像素值更改为0；若是某点像素值大于阈值threshold，则将像素值更改为255，即可得到基于矩不变法的图像分割结果。
5. 最大熵法阈值分割原理：

根据目标和背景两个概率分布用信息论的概念定义两类相互关系测度，计算过程如下：



使熵 H(k)最大的灰度值 k 为图像的分割阈值 Threshold



算法步骤：

1) 假设图像中每个灰度值为阈值k，根据公式6计算与之相对应的熵

2) 遍历每个灰度值，求得熵最大时对应的灰度值，该灰度值则为分割图像的阈值 threshold

3) 遍历灰度图像的像素点，将其与阈值 threshold 比较。若某点像素值小于阈值 threshold，则将像素值更改为 0；若是某点像素值大于阈值 threshold，则将像素值更改为 255，即可得到基于矩不变法的图像分割结果。

1. **分水岭算法：**

1. OpenCV 实现

void cv::watershed (InputArray image, InputOutputArray markers)

image: 输入8-bit的3通道彩色图像

markers:输入/输出32-bit的单通道图像。markers相当于watershed()运行时的种子参数，通常由函数findContours() 和 drawContours()结合使用来获得。

2. 经典分水岭算法步骤

* 1. 图像灰度化。
  2. 根据Sobel 算子计算图像梯度（梯度范围为 0-255），边缘像素的梯度记为其邻域像素的梯度。
  3. 对各像素点的梯度值从小到大排序，梯度值相同的像素为同一层。
  4. 处理第一层所有的像素点，如果其邻域已经被标识属于某一个区域，则将这个像素加入队列。
  5. 队列非空时，弹出第一个元素。扫描该像素的邻域像素，如果其邻域像素的梯度值相等，则根据邻域像素的标识来更新该像素的标识。一直循环到队列为空。
  6. 再次扫描当前梯度值层级的像素，如果还有像素未被标识，说明它是一个新的极小区域，则当前区域的值（当前区域的值从 0 开始计数）加 1 后赋值给该标识的像素。然后从该像素出发继续执行步骤 5）遍历该梯度值层级的所有像素，直至没有新的极小区域。
  7. 返回步骤 4)，处理下一个梯度值层级的像素，直至所有层级的像素都被处理。