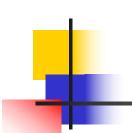


# 计算机视觉

南区计软328

深圳大学 计算机与软件学院





# 实验内容

序号	实验主题	实验内容	实验要求	实验时数	每组人数	实验 类型
1	图像处理应用 实验	1. 熟悉图像的表示及基本元素、通道操作; 2. 掌握基本图像增强方法; 3. 掌握OpenCV计算机视觉库;	必做	6	1	讲授 + 实验
2	图像特征提取 及综合应用实 践	<ol> <li>熟悉图像处理基本操作;</li> <li>掌握图像边缘检测原理;</li> <li>掌握图像基本特征抽取以及在实际问题中的应用;</li> </ol>	必做	6	1	讲授 + 实验
3	计算机视觉系 统实践	1. 熟悉计算机视觉分类任务; 2. 掌握数据集的准备及模型训练过程; 3. 培养应用计算机视觉解决问题的能力;	必做	6	1	讲授 + 实验

# 涉及学科

- ●数字图像处理
- ●计算机视觉
- ●模式识别
- ●程序设计





### 实验考察形式

- 三次实验报告(在Blackboard发布与提交)
- 平时编程练习及表现
- 实验汇报

### 实验考察内容

- 计算机视觉与模式识别基础知识掌握能力
- 算法设计能力
- 编程及系统架构能力
- 论文写作及表达能力





### 实验二 图像处理综合-路沿检测

### 实验目的:

- 1. 熟悉图像处理基本操作;
- 2. 掌握图像边缘检测原理;
- 3. 掌握图像基本特征抽取以及在实际问题中的应用;

### 实验要求:

- 1. 实验提交文件为实验报告和相关程序代码,以压缩包的形式提交,实验报告命名规则为"计算机视觉-学号-姓名-实验报告2. doc",其他文件打包成压缩文件,命名为"计算机视觉-学号-姓名-实验报告2-其他. zip";
- 2. 所有素材和参考材料需列明出处,实验报告中的图片和程序代码建议标注个人水印或标识信息: 姓名,班级,学号信息;

### 二、实验内容:

针对给定的视频,利用图像处理基本方法实现道路路沿的检测;

提示:可利用Hough变换进行线检测,融合路沿的结构信息实现路沿边界定位(图中红色的点位置)。

### 三、实验时间

实验报告统一命名为: 计算机视觉-学号-姓名-实验报告2. doc 如有提交其他文件或程序, 统一命名为: 计算机视觉-学号-姓名-实验报告2-其他. zip

实验报告提交时间: 2024年11月18日下午5:00 (注意按时提交,不要延期)



# ● 实验二

### •一、实验目的与要求

### 实验目的:

- 1. 熟悉图像处理基本操作:
- 2. 掌握图像边缘检测原理;
- 3. 掌握图像基本特征抽取以及在实际问题中的应用:

### 实验要求:

- 实验提交文件为实验报告和相关程序代码,以压缩包的形式提交,实验报告命名规则为"计算机视觉-学号-姓名-实验报告 2.doc",其他文件打包成压缩文件,命名为"计算机视觉-学号-姓名-实验报告 2-其他zip";
- 所有素材和参考材料需列明出处,实验报告中的图片和程序代码建议标注个人水印或标识信息,姓名,班级,学号信息;

### •二、实验内容与方法

实验内容: 针对给定的视频,利用图像处理基本方法实现道路路沿的检测: 提示: 可利用 Hough 变换进行线检测,融合路沿的结构信息实现路沿边界定位(图中 红色的点位置)。



### •三、实验步骤与过程

是示:是之件を用請酬給了底标注提示信息力 根据实验内容自己组织填写。内容极图插入实验报告后人小要调视适中,并且必求加上相关的说明。每页图片建议不超过6米、排版的类型型加入评分标准。(以后所有的实验报告都是这个要求、不再单复证明)。

### • 四、实验结论或体会





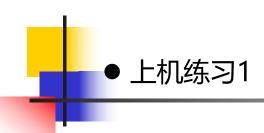












# 实验课上机练习内容:

- 1. OpenCV环境配置;
- 2. 实现对图像的读取;
- 3. 把一副彩色图像的三个通道变成3个单通道图像存储到硬盘上并显示;
- 4. 计算一幅单通道图像的直方图;
- 5. 编程实现对一幅单通道图像的边缘检测。
- 6. 对一幅灰度图像进行直方图均衡化
- 7.对图像进行二值化操作;
- 8. 图像形态学操作;
- 9. 获取图像轮廓;
- 10. 实现目标的计数;
- 11. 目标的旋转;









1.在图像中实现目标匹配; (先获取该目标的一个图像, 然后利用这个已知的图像, 在测试图像上找到该目标)

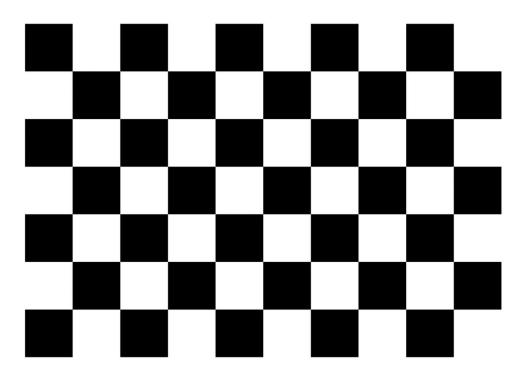






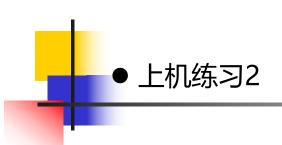


2.在图像上通过霍夫变换寻找直线;

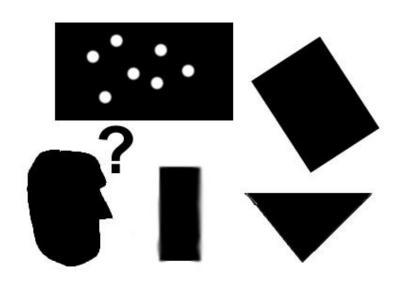


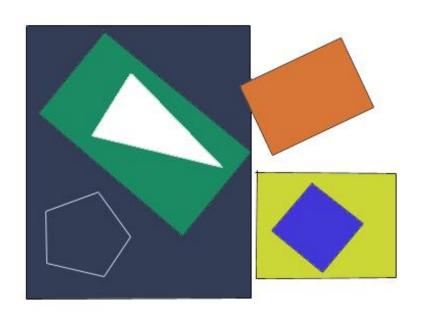
This is a 9x6
OpenCV chessbo





**3.**基于霍夫变换,实现三角形检测,如果是四边形呢?多边形呢?

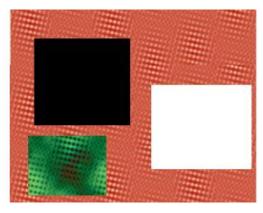


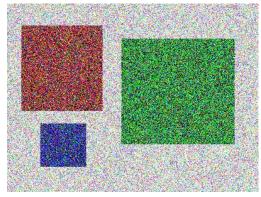




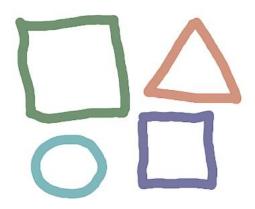


**3.**基于霍夫变换,实现三角形或矩形检测,如果是四边形呢? 多边形呢?













# 4.练习基于OpenCV的视频读取。



### 实验二: 图像处理综合-路沿检测

已附加文件: 计算机视觉-学号-姓名-实验报告2.doc (140 KB)

□ 实验2实验数据.zip ◎ (16.718 MB)

☐ 其他实验图像和视频.zip ○ (17.323 MB)

■ 实验课-第3节.pdf (1.986 MB)

### 实验目的:

- 1. 熟悉图像处理基本操作;
- 2. 掌握图像边缘检测原理;
- 3. 掌握图像基本特征抽取以及在实际问题中的应用;

### 实验要求:

- 1. 实验提交文件为实验报告和相关程序代码,以压缩包的形式提交,实验报告命名规则为"计算机视觉-学号-姓名-实验报告2.doc",其何
- 2. 所有素材和参考材料需列明出处,实验报告中的图片和程序代码建议标注个人水印或标识信息: 姓名,班级,学号信息;
- 二、实验内容:

针对给定的视频,利用图像处理基本方法实现道路路沿的检测;

提示:可利用Hough变换进行线检测,融合路沿的结构信息实现路沿边界定位(图中红色的点位置)。

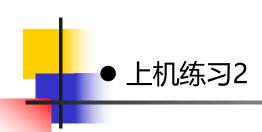
三、实验时间

实验报告统一命名为: 计算机视觉-学号-姓名-实验报告2.doc

如有提交其他文件或程序,统一命名为: 计算机视觉-学号-姓名-实验报告2-其他.zip

实验时间: 2022年4月4日至2022年5月15日(注意按时提交,不要延期)

实验报告提交时间: 2022年5月15日下午5:00





# 4.练习基于OpenCV的视频读取。



### 实验二: 图像处理综合-路沿检测

已附加文件: 计算机视觉-学号-姓名-实验报告2.doc (140 KB)

□ 实验2实验数据.zip ◎ (16.718 MB)

☐ 其他实验图像和视频.zip ○ (17.323 MB)

■ 实验课-第3节.pdf (1.986 MB)

### 实验目的:

- 1. 熟悉图像处理基本操作;
- 2. 掌握图像边缘检测原理;
- 3. 掌握图像基本特征抽取以及在实际问题中的应用;

### 实验要求:

- 1. 实验提交文件为实验报告和相关程序代码,以压缩包的形式提交,实验报告命名规则为"计算机视觉-学号-姓名-实验报告2.doc",其何
- 2. 所有素材和参考材料需列明出处,实验报告中的图片和程序代码建议标注个人水印或标识信息: 姓名,班级,学号信息;
- 二、实验内容:

针对给定的视频,利用图像处理基本方法实现道路路沿的检测;

提示:可利用Hough变换进行线检测,融合路沿的结构信息实现路沿边界定位(图中红色的点位置)。

三、实验时间

实验报告统一命名为: 计算机视觉-学号-姓名-实验报告2.doc

如有提交其他文件或程序,统一命名为: 计算机视觉-学号-姓名-实验报告2-其他.zip

实验时间: 2022年4月4日至2022年5月15日(注意按时提交,不要延期)

实验报告提交时间: 2022年5月15日下午5:00







# 传统Sobel方法存在问题

-1	0	1
-1	0	1
-1	0	1







1	1	1
0	0	0
-1	-1	-1



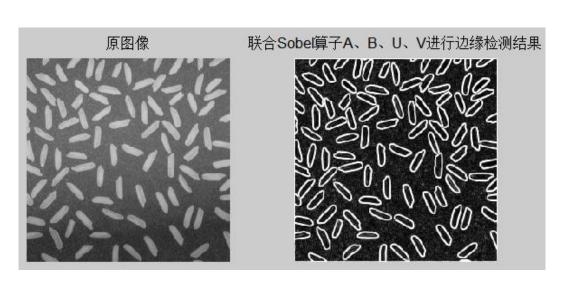
只对X、Y方向边缘具 有较好检测效果。其 他方向呢?





### 基于最优梯度幅值的边缘检测

### 结合四个(多)方向Sobel算子





### ● 基于最优梯度幅值的边缘检测

```
% 联合Sobel算子A、B、U、V进行边缘检测结果
1
2 -
       close all; clear; clc;
       I=imread('rice.png'):%m*n的单通道灰度图
       subplot (1, 2, 1)
       imshow(I)
       title('原图像'):
       [h.w]=size(I):
       I_reshape=zeros(h+2, w+2);%创建一个上下左右均多一个像素宽的零矩阵
       I reshape (2:h+1, 2:w+1)=I:%把I矩阵拷贝到新创建矩阵中间
       g=zeros(h,w):%用于保存结果的矩阵
       A=[-1 -2 -1;
12
           0 0 0;
13
           1 2 11:
       B=[-1 0 1;
14 -
15
          -2 0 2:
          -1 0 1]:
       U=[-2 -1 0;
          -1 0 1:
19
          0 1 21:
       V= [0 1 2:
         -1 0 1:
21
22
          -2 -1 0];
       %I在I_reshape矩阵中纵向方向的范围是j=2到j=h+1,其中j=1和j=h+2对应0像素值
       %I在I reshape矩阵中横向方向的范围是i=2到i=w+1,其中i=1和i=w+2对应0像素值
     | for i=2:h+1
26 -
           for i=2:w+1
27 -
               aa=0;
               bb=0:
28 -
               uu=0;
29 -
30 -
               vv=0
31 -
              for r=-1:1
32 -
33 -
                     aa=aa+I reshape(i+r,i+s)*A(r+2,s+2): %在3*3领域进行卷积操作
34 -
                    bb=bb+I_reshape(j+r, i+s)*B(r+2, s+2); %在3*3领域进行卷积操作
                     uu=uu+I_reshape(j+r, i+s)*U(r+2, s+2); %在3*3领域进行卷积操作
35 -
                     vv=vv+I_reshape(j+r,i+s)*V(r+2,s+2); %在3*3领域进行卷积操作
36 -
37 -
                  end
38 -
               t=max(abs(aa), abs(bb));
39 -
40 -
               t=max(t,abs(uu));
41 -
               t=max(t,abs(vv));
42 -
               g(j-1, i-1)=t;%取最大值作为模值
43 -
           end
44 -
45 -
       subplot (1, 2, 2)
       g=uint8(g);
       imshow(g)
       title("联合Sobel算子A、B、U、V进行边缘检测结果"):
```

根据算法思想,把程序 转换为VC+OpenCV形式 或Python+OpenCV形式



### ● 基于最优梯度幅值的边缘检测

关于Sobel图像边缘检测在图像边界的处理方式:

- 1)边界不处理;
- 2) 扩大2个像素, 创建更大的图像保存结果;
- 3) 通过约束条件的方式处理边界点

```
Mat <uchar> dataMat=InputImg;
Mat <float> dataResultMat=Gray32Mat;
int oh=OperatorSize/2;//算子Operator的半径
int ow=OperatorSize/2;
for(int j=0;j<InputImg.rows;j++)
  for(int i=0;i<InputImg.cols;i++)
    float datavalue=0:
    for(int p=-oh;p<=oh;p++)
       for(int q=-ow;q<=ow;q++)
         if(p+j<0)|p+j>InputImg.rows-1||q+i<0||q+i>InputImg.cols-1)
           continue:
         datavalue+=dataMat(j+p,i+q)*Operator[(p+oh)*OperatorSize+(q+ow)];
    dataResultMat(j,i)=saturate_cast<float>(datavalue);
```

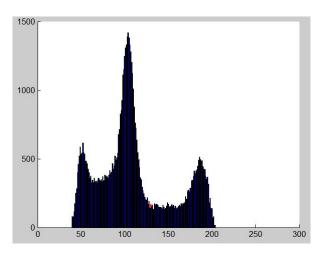


### ● 基于迭代二分法的自动阈值分割

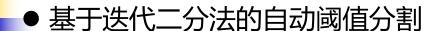
采用迭代二分法可以自适应寻找灰度图像中的最佳分割阈值,迭代二分法的基本原理是首先找出图像的最大灰度值和最小灰度值,然后取平均值作为初始的分割阈值 T,那么 T可以把图像分割为两部分,即灰度值大于 T的部分和灰度值小于 T的部分;分别计算这两部分的平均灰度值 H和 L,更新 T=(H+L)/2;重复以上步骤直至算法收敛。编写迭代二分法求最佳阈值的代码。











Step1:求图像的最大、最小灰度值 $T_{max}$ 和 $T_{min}$ ,令初始阈值为:

$$T_0 = \frac{T_{\text{max}} + T_{\text{min}}}{2}$$

Step2:根据阈值 $T_k$ ,将图像分割成目标和背景两部分,求

出两部分的平均灰度:

$$Z_F = \frac{\sum\limits_{Z(i,j) < T_K} Z(i,j) \times N(i,j)}{\sum\limits_{Z(i,j) < T_K} N(i,j)} \qquad Z_B = \frac{\sum\limits_{Z(i,j) > T_K} Z(i,j) \times N(i,j)}{\sum\limits_{Z(i,j) > T_K} N(i,j)}$$

像素点的

灰度值

个类

Step3:求出新阈值

$$T_{K+1} = \frac{Z_F + Z_B}{2}$$

如果  $|T_{K+1} - T_K| < \varepsilon$  或者迭代次数大于规定的次数,则程序结束

否则  $K \leftarrow K + 1$  转Step2。

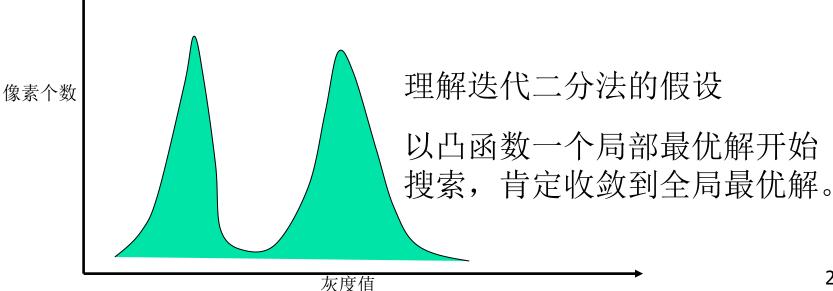




### 基于迭代二分法的自动阈值分割

### 注意的地方:

- 1)每次更新阈值,都直接在原图统计大于某个阈值的像素个数,效率很低;
- 2) 灰度图计算错误,统计的是像素个数,不是像素值累加;
- 3) 要添加收敛参数控制收敛次数;





### ● 基于直方图的颜色特征提取

- 1. 把RGB图像转换到YCrCb空间并提取Cr色度通道(也可以是其他类似的颜色空间,如HSV)
- 2. 对于每一个颜色等级,通过公式  $\hat{q}_u = \sum_{i=1}^n \delta \left[ c(x_i^*) u \right]$  获取其对应的概率,其中: m:m 个颜色等级(如灰度图有256个颜色等级);

 $\{x_i\}_{i=1\cdots n}$ : 图像的 n 个像素值;

 $\{\hat{q}\}_{u=1\cdots m}$ : 某个颜色等级上像素个数;

 $c: \Re^2 \to \{1 \cdots m\}$ : 把第 *i* 个像素映射到其对应颜色等级。

3. 通过公式  $\left\{\hat{p}_u = \min\left(\frac{255}{\max\left(\hat{q}\right)}\hat{q}_u, 255\right)\right\}_{u=1\cdots m}$  把统计直方图的取值范围从区间  $\left[0, \max\left(q\right)\right]$  映射到区间  $\left[0, 255\right]$ 。



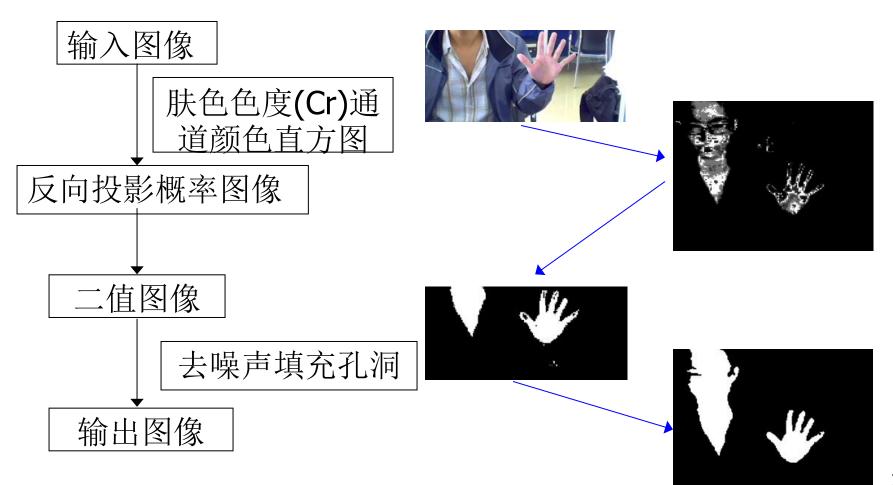






### 基于直方图的颜色特征提取

# 预处理

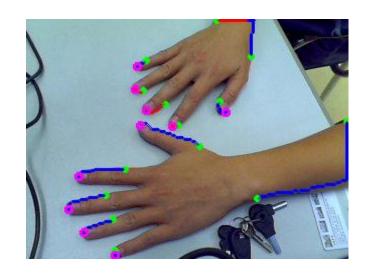




# 深圳大学 SHENZHEN UNIVERSITY

# 基于直方图的颜色特征提取

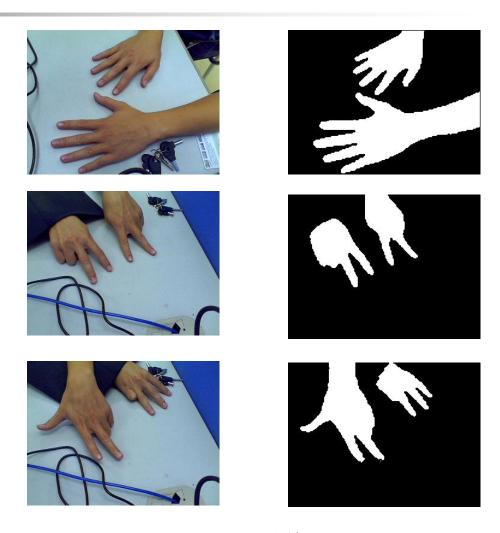


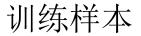


肤色分割



# 基于直方图的颜色特征提取

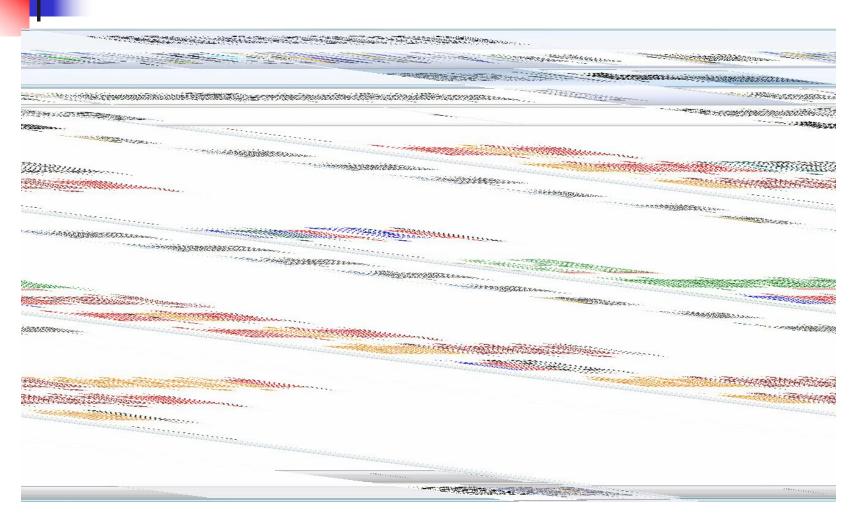




测试图像



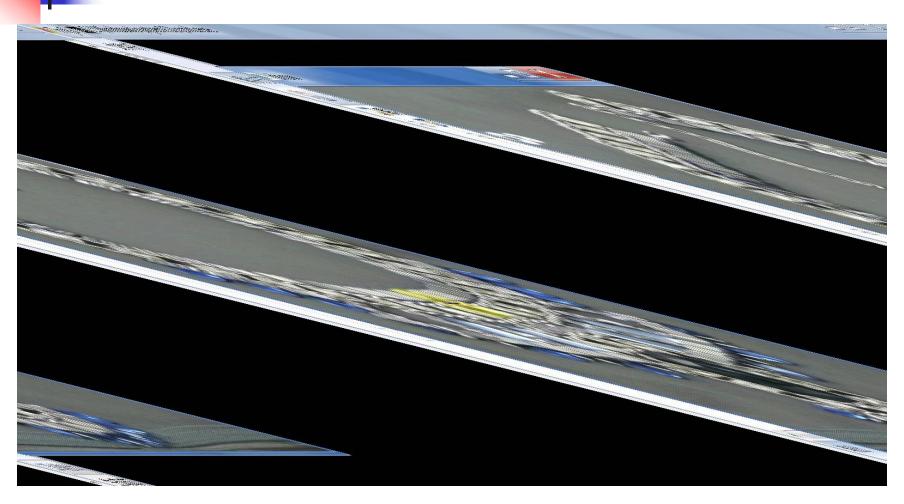
### ● 基于直方图的颜色特征提取



配饰颜色提取



# ● 基于直方图的颜色特征提取



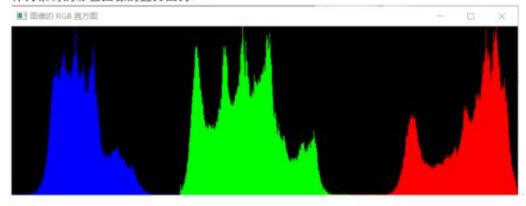
颜色提取的物料计数



# ● 彩色直方图特征提取

# 结果如下: | Compared a production | Delication | Delication

### 作为素材的彩色图像的直方图为:



错误示例1





# 彩色直方图特征提取

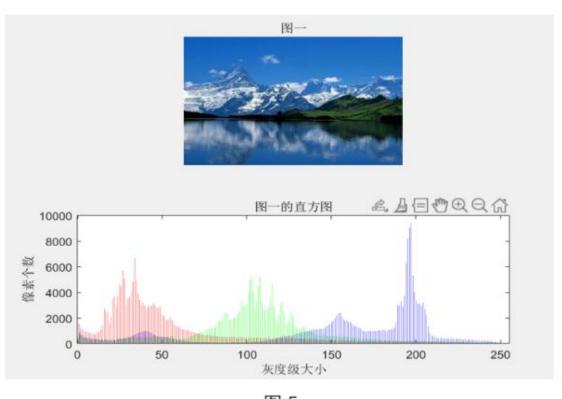


图 5

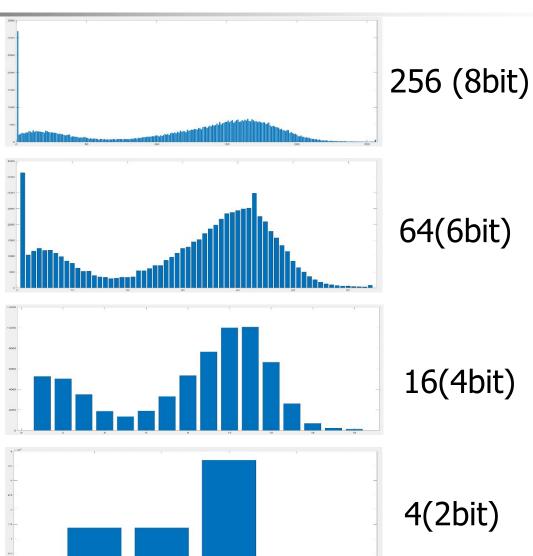
错误示例2





# 彩色直方图特征提取







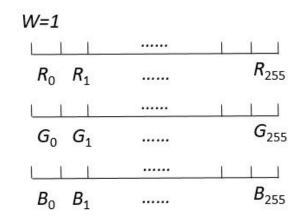


### ● 彩色直方图特征提取

# ◆ 彩色直方图计算思路



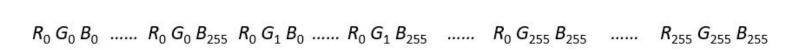






统计彩色直方图特征

一个柱子可表达彩色空间1个点位置上的像素个数



2563=16777216



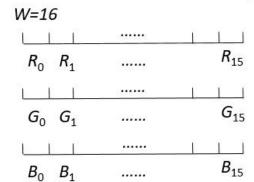
### ● 彩色直方图特征提取

### ◆ 彩色直方图计算思路

 $R_0$ 代表像素值围0-15  $R_{15}$ 代表像素值围240-255





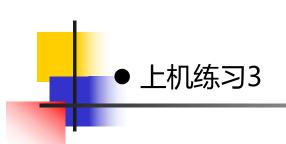




统计彩色直方图特征

163=4096

 $R_0 G_{15} B_{15}$ 表示R通道上落入第1个柱子,G通道上落入第16个柱子,B通道落入第16个柱子的像素总数是多少





● 基于最优梯度幅值的边缘检测

● 基于迭代二分法的自动阈值分割

● 基于直方图的颜色特征提取

● 彩色直方图特征提取