

图像处理入门

龚林奥

1748462079@qq.com



目录

C++基础

OpenCV

- 安装
- 常用函数和结构

图像处理基础与概念

- 图像表示
- 预处理
- 形态学
- 其他

如何构建工程

- Cmake使用
- 工程结构



C++ Preview

- 数组
- 函数
 - 形参与实参
 - 传值、传指针和传引用
- 指针
 - 空指针和野指针
 - 常量指针和指针常量
- 结构体



C++ Preview

- 引用
- 类和对象
- 模板
- STL库



图像处理任务

Classification



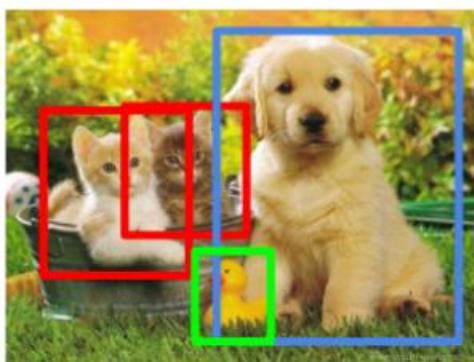
CAT

**Classification
+ Localization**



CAT

Object Detection



CAT, DOG, DUCK

**Instance
Segmentation**



CAT, DOG, DUCK



图像处理基础和相关概念

图像处理基础

- 图像获取
- 颜色空间
- 图像表示
- 数学运算

图像预处理

- 灰度变换
- 空间滤波
- 频域滤波

形态学操作

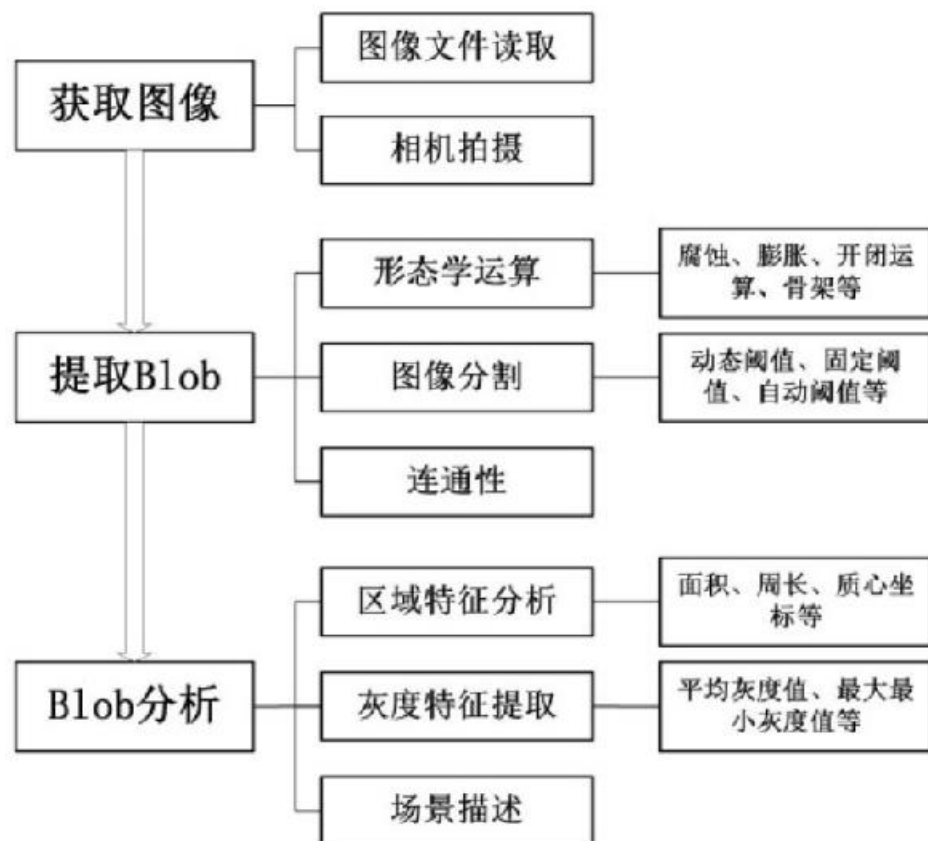
- 腐蚀和膨胀
- 基本算法

其他

- 边缘检测
- 几何形状检测
- 特征提取
- 图像分割



Blob分析流程

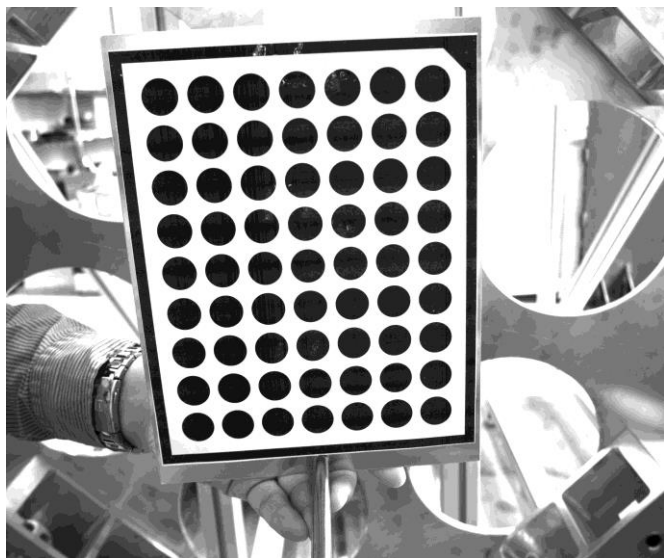


- 适用任务：高对比度、场景简单
- 不适用：低对比度、必要特征不能用二值描述、模板检测

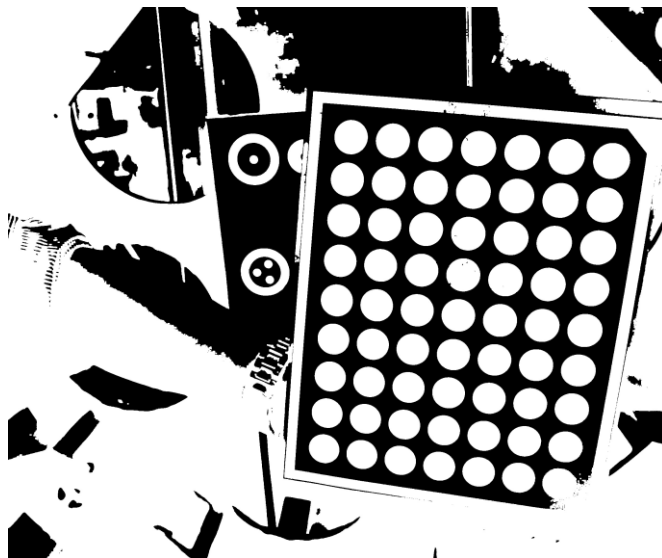


Blob分析举例

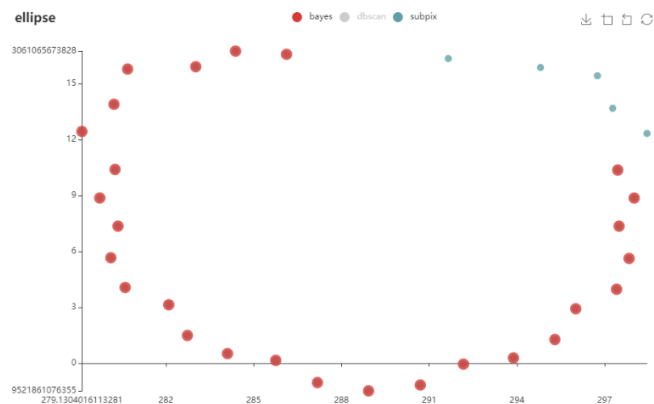
寻找椭圆标定板



二值化分割



连通域提取获取Blob

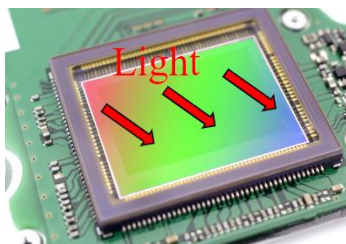


提取亚像素边缘拟合椭圆定位

提取blob特征并筛选：面积、欧拉数、圆形性



图像处理基础 – 图像获取



CMOS/CCD

光子

能量

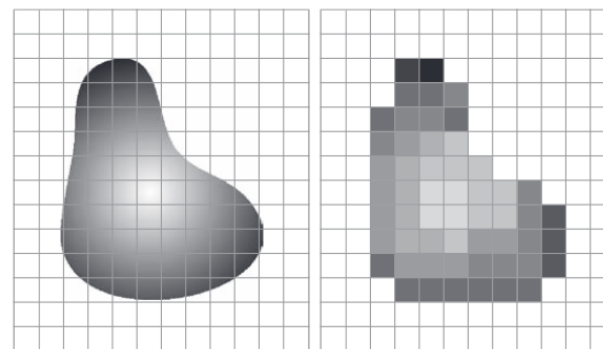
电子

采样

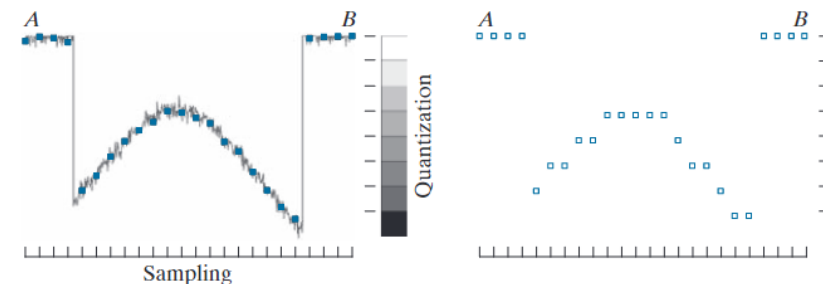
量化

空间离散

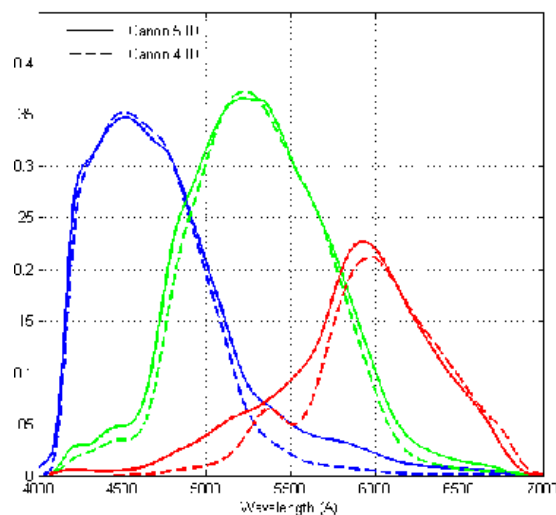
值离散



ADC



图像处理基础 – 彩色图像获取

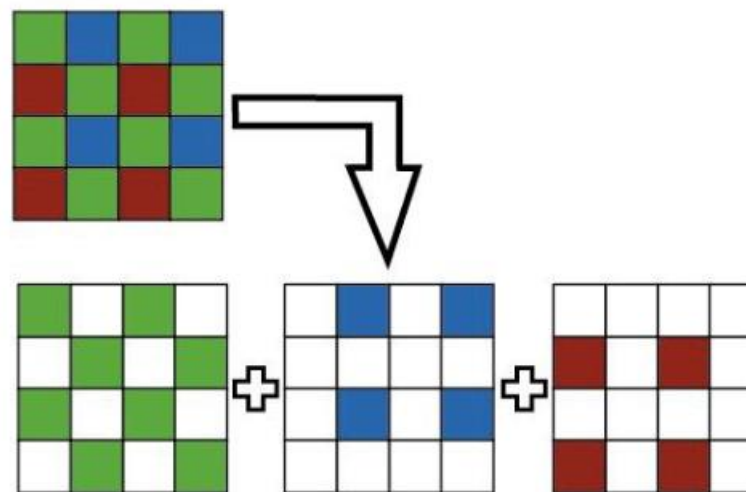


CCD/CMOS对不同波长的光子有不同的激发效率(灵敏度), 一般会有会考虑:

1. 量子效率
2. 满井容量 (HDR有关)

分别利用3种不同波长敏感传感器获取对应灰度图像, 最后叠加即为彩色图。现在常用的彩色面阵相机都是BAYER阵列, 最后的结果使用插值得到

Bayer Pattern



图像处理基础 – 颜色空间

RGB颜色空间与色度图

利用三种RGB波长的光线性叠加混合

常见空间
:sRGB\CIE
RGB\Adobe
RGB\Linear RGB

CMYK

利用二次色叠加混合，用于印刷品

HSV

色调、饱和度、亮度

HIS

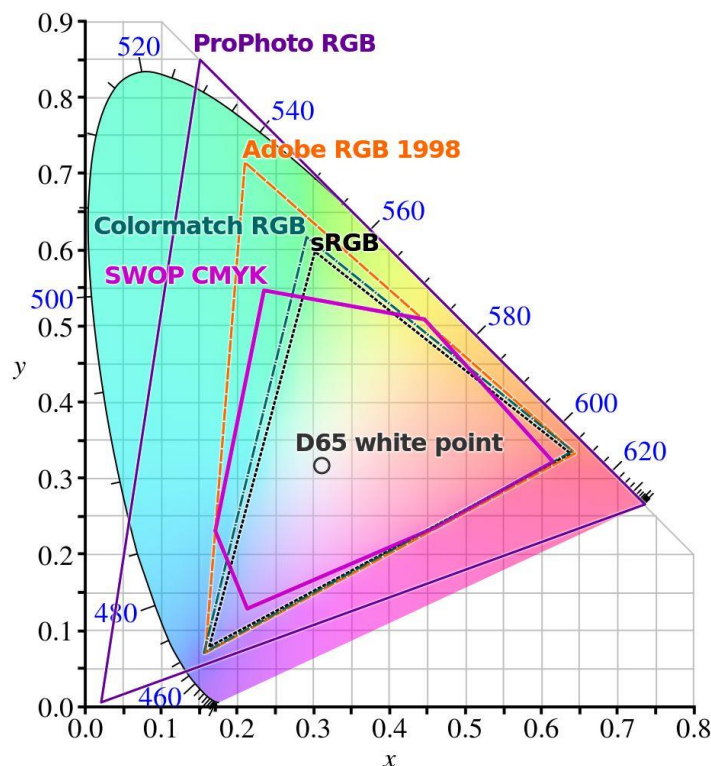
色调、饱和度、明度

CIE Lab

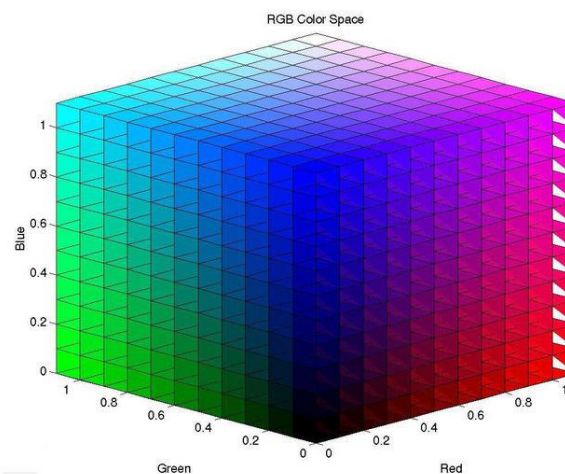
中间转换空间



图像处理基础 – RGB颜色空间



- RGB颜色空间使用R\G\B三种波长的光线性叠加合成，CIE RGB表示了人能够感知的颜色范围，常见的色度图是CIE XYZ在 $x+y+z = 1$ 平面的投影。常见的色彩空间为sRGB、aRGB、Linear RGB，是表示是选取不同的点(即不同颜色的光)作为基底，各个分量在 $[0,1]$ 之间，即空间的立方体，投影后是三角形。关于sRGB和Linear RGB的区别，会在后面介绍gamma变换的时候引入。需要额外注意的是，在RGB空间中， $R=G=B$ 的直线对应灰度级。



参考:<https://zhuanlan.zhihu.com/p/24281841>



图像处理基础 – CMYK颜色空间



Red, green & blue are the primaries of the additive color system, and combine to make white.

CMY
Subtractive Color

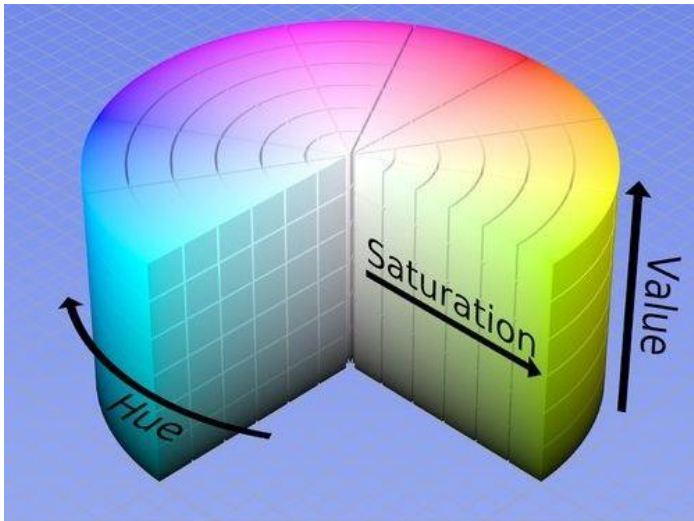


Cyan, magenta & yellow are the primaries of the subtractive system, and combine to make black.

- RGB颜色空间适合发光物体表示，而印刷品是通过反射光来感受颜色的，故可以使用白光 – R/G/B得到的二次色C\Y\M表示，而由于没有纯净的二次色，因此需要额外加入黑色K来进行混合。

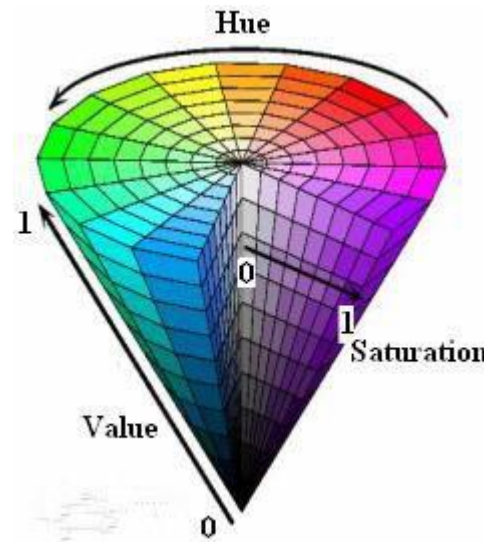


图像处理基础 – HSV颜色空间



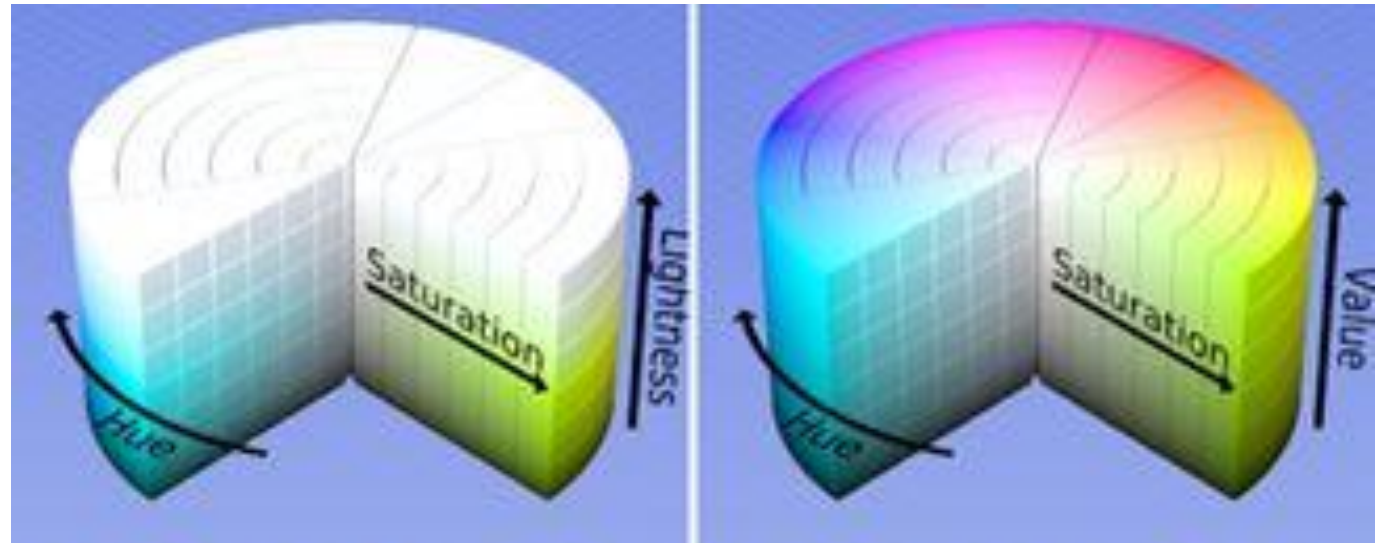
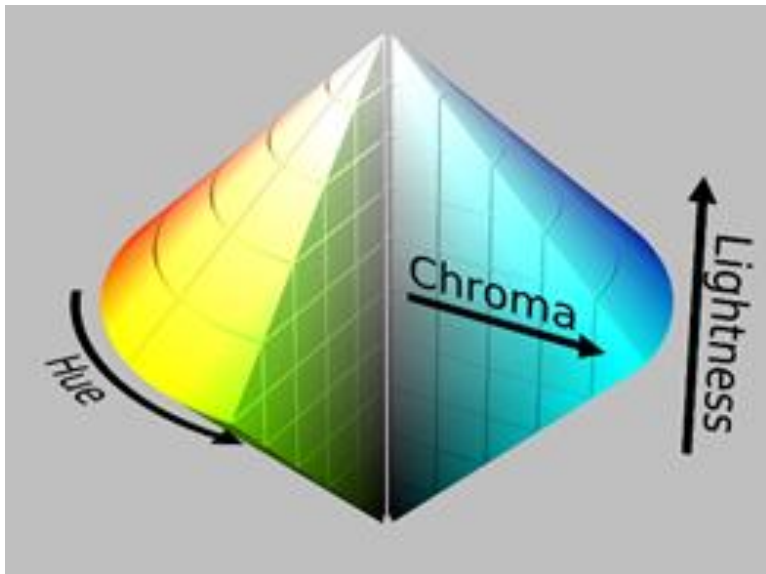
亮度在0-1之间，代表从黑色到白色，对于RGB纯色，其饱和度为1(不掺杂白光)，亮度也为1

- 除了使用三种不同波长的光混合对颜色进行表示，还有其他表示方法：色调(H)、饱和度(S)、亮度(V/B),而色调和饱和度统称为色度。色调与主波长有关；饱和度为相对纯度，与掺杂白光量有关。

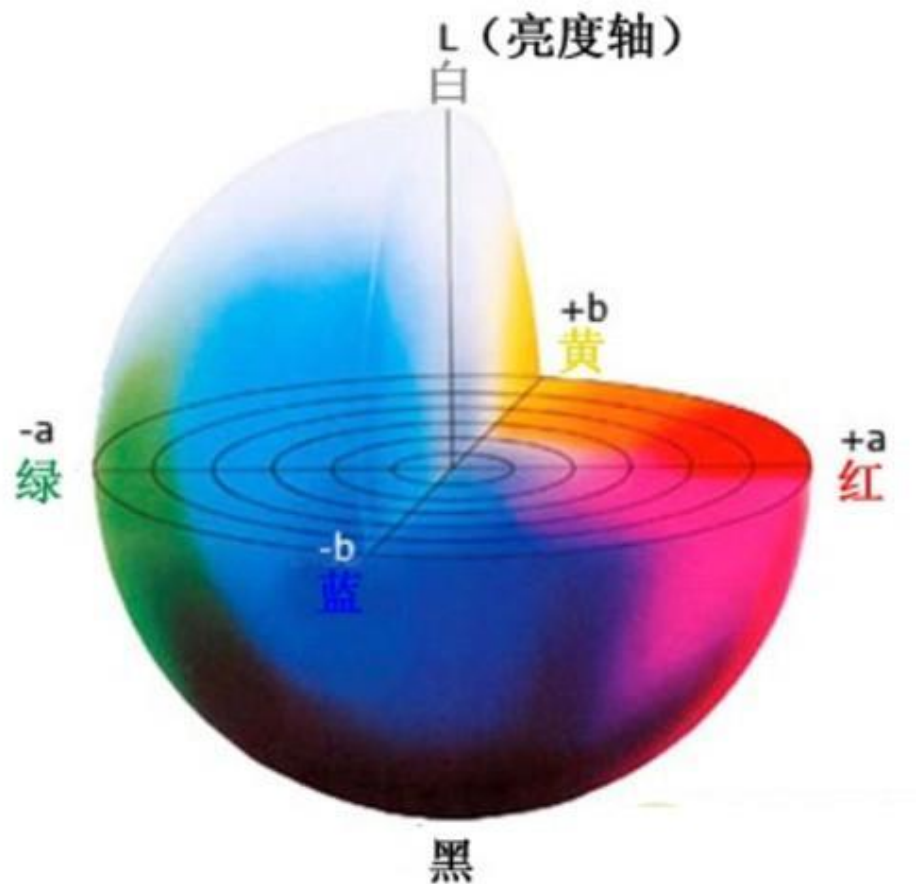


图像处理基础 – HSI颜色空间

- HIS颜色空间与HSV类似，只有S和I有所不同，分别代表色饱和度和明度。对于纯色，其明度为0.5，色饱和度为1，为一个对称的锥体。



图像处理基础 – CIE L a b颜色空间



CIE Lab是一种设备无关的颜色空间，常用来做转换的中间空间，其色域大于RGB空间，即表达的颜色超过人的可感知颜色。



图像处理基础 – 图像表示

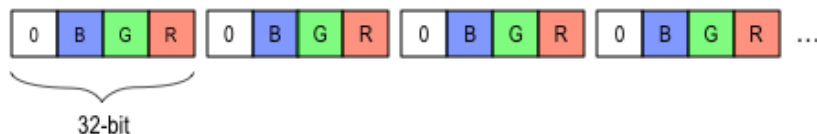


- 对于一个灰度图像，可以使用一个二维矩阵表示。对于OpenCV，这个二维矩阵其会以行优先存储，需要特别注意一下OpenCV里行列宽高对应的规则。图像的原点在左上角，x轴对应宽即列数，y轴对应高即行数。

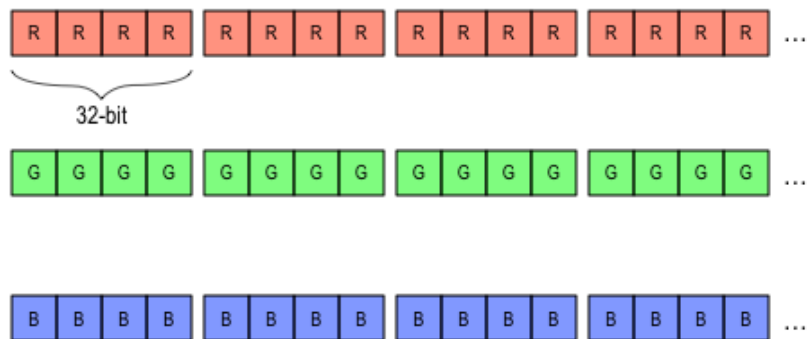


图像处理基础 – 图像表示

HWC



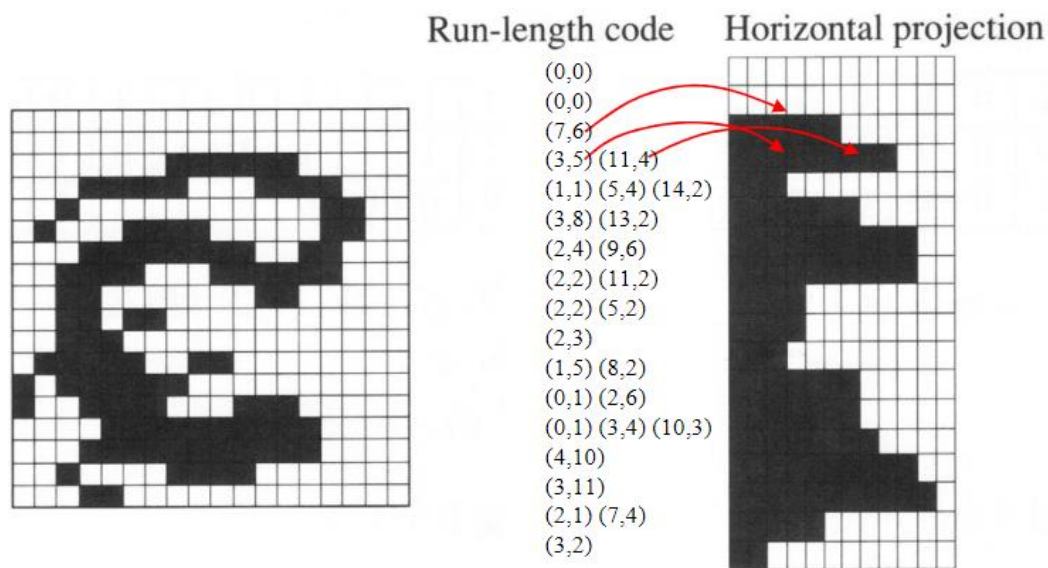
CHW



- (N)HWC和(N)CHW.对于一副RGB彩色图像，其使用3个二维矩阵存储RGB分量，对于整体的存储有两种方式，分别是HWC和CHW，OpenCV 彩色图像存储方式为HWC，一般的卷积神经网络要求输入格式为CHW。特别需要注意的是，OpenCV默认存储格式为BGR而非RGB



图像处理基础 – 图像表示



- 其他表示方式：
 - 游程编码(RLE)。对于二值图像，可以使用行程(游程)编码对图像进行无损压缩。基于RLE,也可以更高效的实现形态学操作。



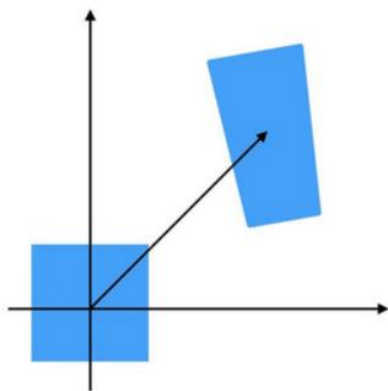
图像处理基础 – 常用数学运算

- 对单个像素
 - Gamma变换、直方图均衡化
- 空间变换
 - 仿射变换、投影变换
- 邻域运算
 - 图像卷积\相关、中值滤波
- 其他
 - 集合、逻辑



图像处理基础 – 空间变换

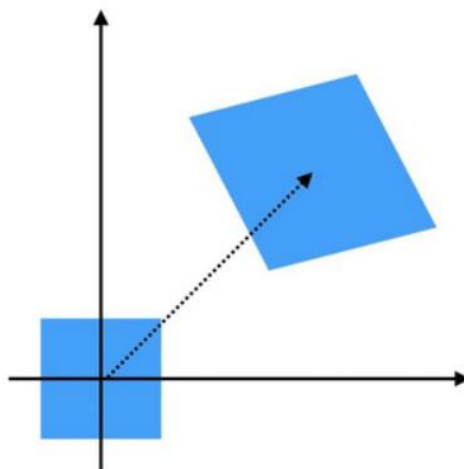
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} A_{2 \times 2} & T_{2 \times 1} \\ V^T & s \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H_{3 \times 3} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



投影变换

仿射变换

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} A_{2 \times 2} & T_{2 \times 1} \\ 0^T & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



- 空间变换有3D->2D和2D->2D两类，这个部分在后续的课程会详细介绍，这里只简单提几种变换以及例子。



图像处理基础 – 卷积与相关

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

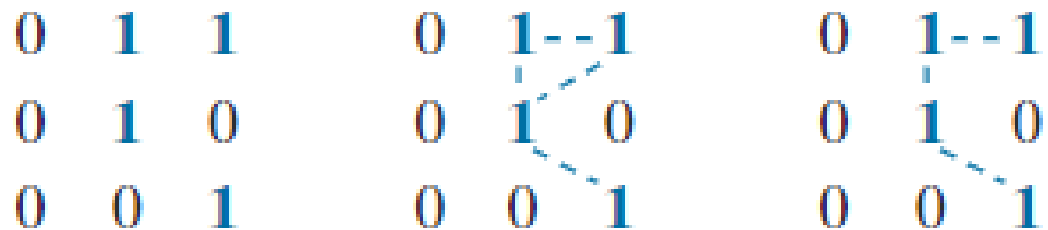
4		

Convolved
Feature

- 卷积和相关是空间滤波和频域滤波的基础，可以视作局部像素的内积，因此在归一化的情况下，卷积结果越大，其与局部像素分布越一致。



图像处理基础 – 邻接



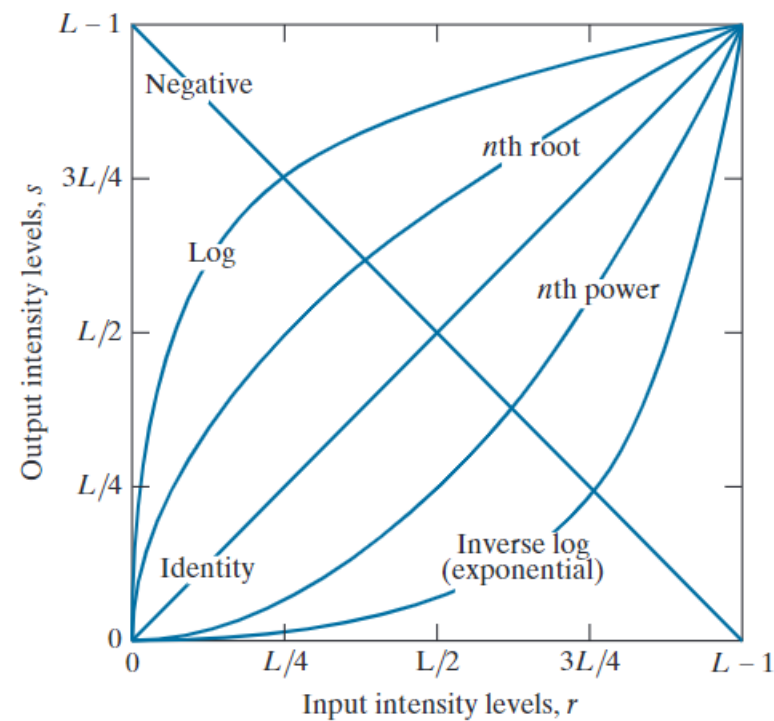
- 对于形态学的连通域分析，需要先定义邻接，邻接有：4邻接、8邻接和m(混合)邻接。



图像预处理 – Gamma变换



- Gamma变换和对数变换。因为人眼对于光强的感知并不是线性的，而是近似对数曲线，因此会更加关注暗部的变化。而sRGB和Linear RGB的差异就在于此。



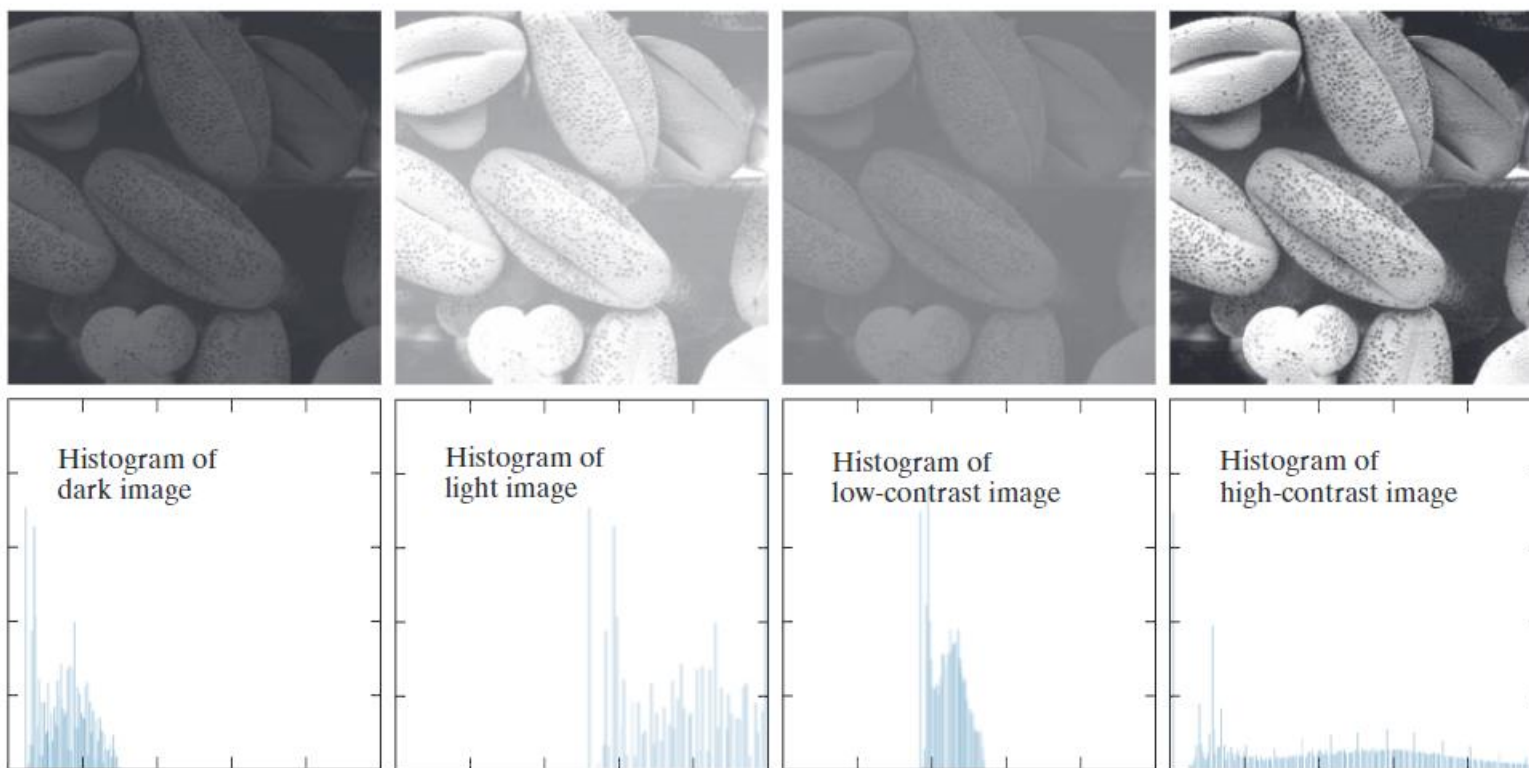
$$\text{Power Law : } s = cr^\gamma$$

$$\text{Log : } s = c \log(1 + r)$$



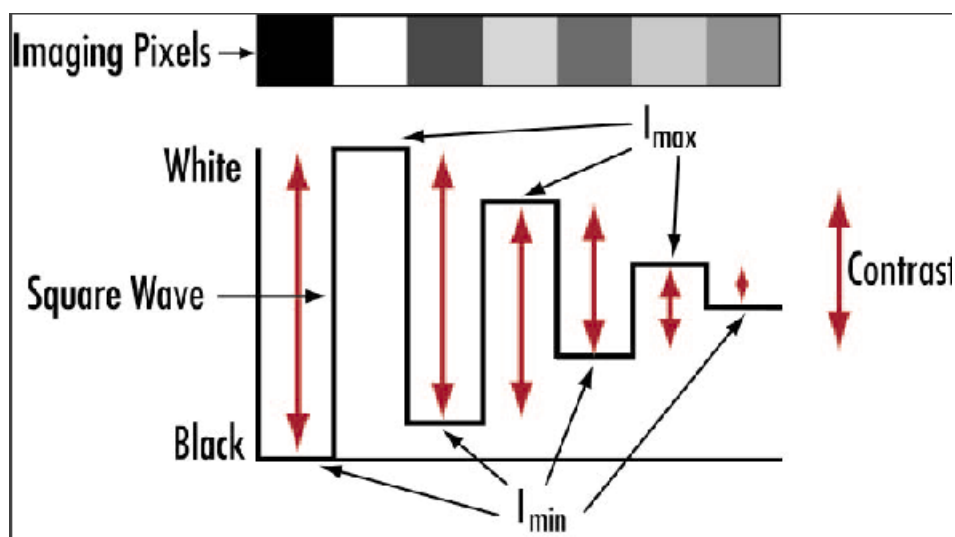
图像预处理 – 直方图均衡化

- 直方图与对比度



图像预处理 – 直方图均衡化

$$\text{Contrast}\% = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}$$



- 增强对比度。直方图均衡化将图像各个灰度所占的比例构成的直方图变换为近似均匀分布，实现局部的像素分布 I_{\max} 和 I_{\min} 存在足够大的差别，进而增强对比度



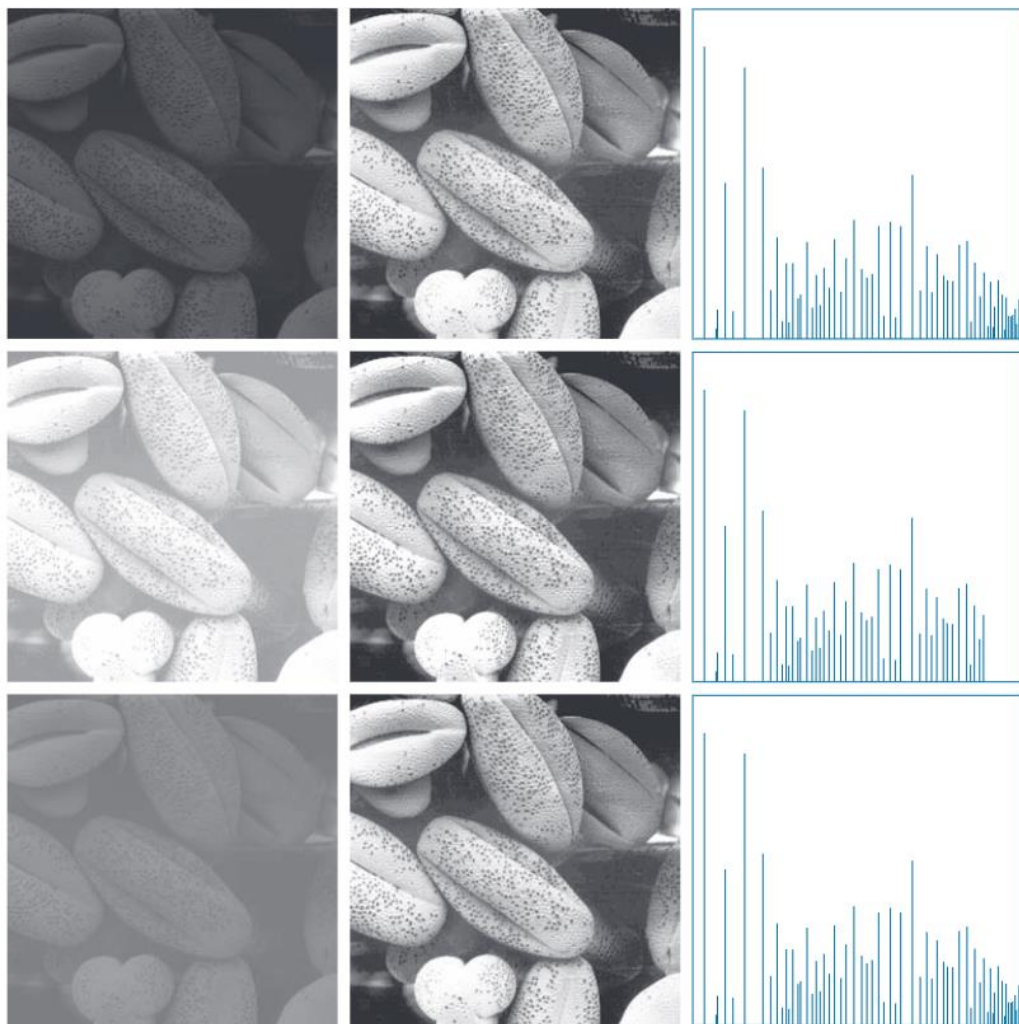
图像预处理 – 直方图均衡化

- 流程

序号	运 算	步骤和结果							
1	原始图像灰度级 r_k	0	1	2	3	4	5	6	7
2	原始直方图 $p_r(r_k) = \frac{n_k}{n}$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
3	计算累积直方图各项 $s_k = \sum_{j=0}^k \frac{n_j}{n}$	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1.00
4	取整扩展: $S(k) = \text{int} \left[\left((L-1) - 0 \right) \cdot s_k + 0.5 \right]$	1	3	5	6	6	7	7	7
5	确定映射对应关系 $r_k \rightarrow S(k)$	0 → 1	1 → 3	2 → 5	3, 4 → 6		5, 6, 7 → 7		
6	根据映射关系计算均衡化直方图		0.19		0.25		0.21	0.24	0.11



图像预处理 – 直方图均衡化



图片参考:数字图像处理(第四版)



图像预处理 – 滤波



- 滤波包括空间滤波和频域滤波，其实现方式多是使用卷积实现，可以实现图像的去噪、锐化和重建等等，这一部分下周会详细介绍。

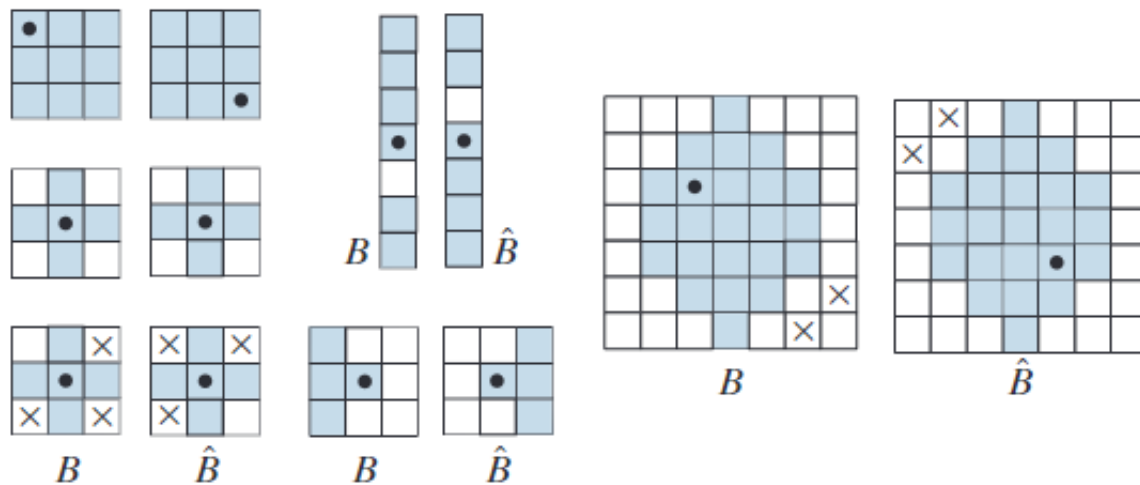


图像形态学 – 简介

- 利用集合定义并进行运算
- 主要用来提取区域相关图像分量：边界、骨架、凸壳
- 也可以用于滤波、细化、分割



图像形态学 – 结构元(SE)

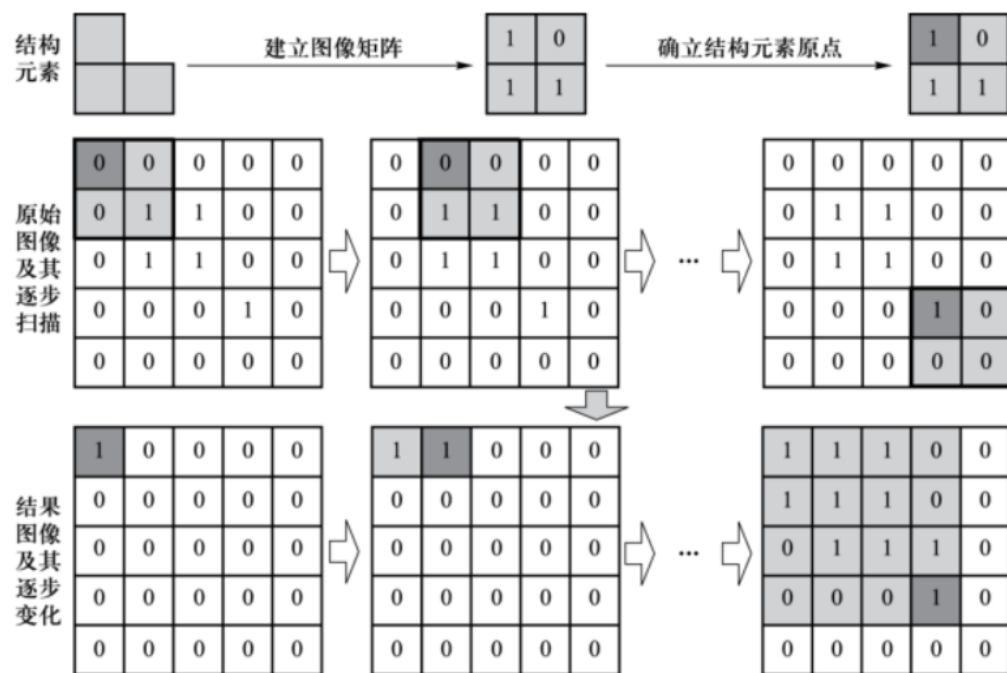


- $SE = \text{前景} + \text{背景}$, 目标 = 前景
- 不关心元素x、前景像素、背景像素
- 定义运算:
- 平移 (滑动窗口)、反射 (相对原点旋转180)

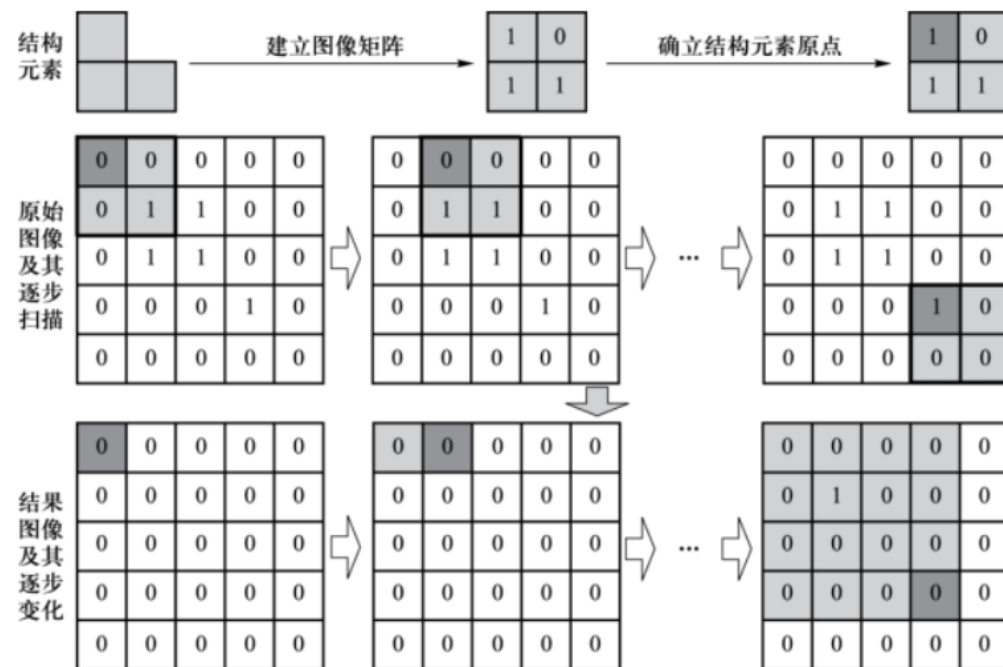


图像形态学 – 腐蚀和膨胀

膨胀：填充空洞、裂缝，注意是SE的反射

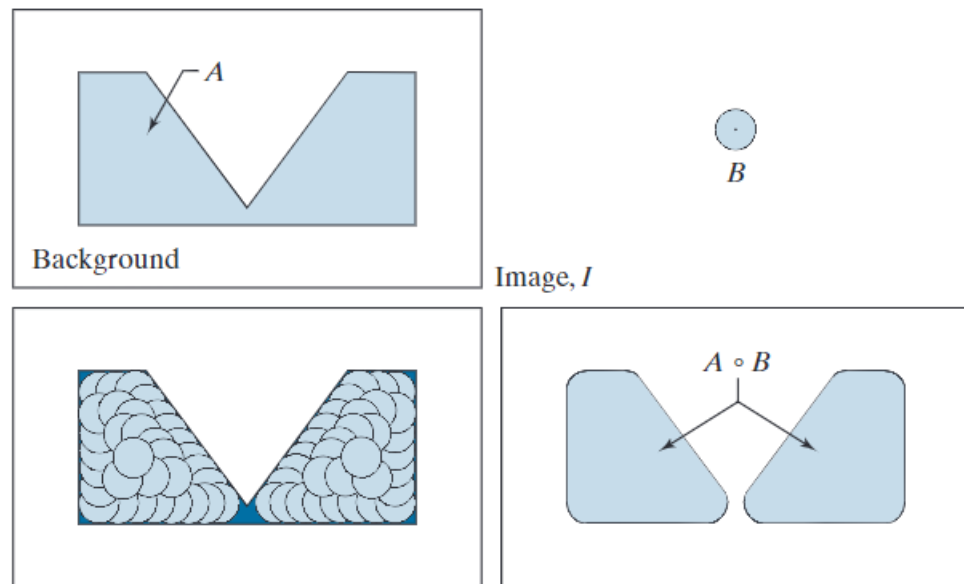


腐蚀：滤除图像中小于SE的细节

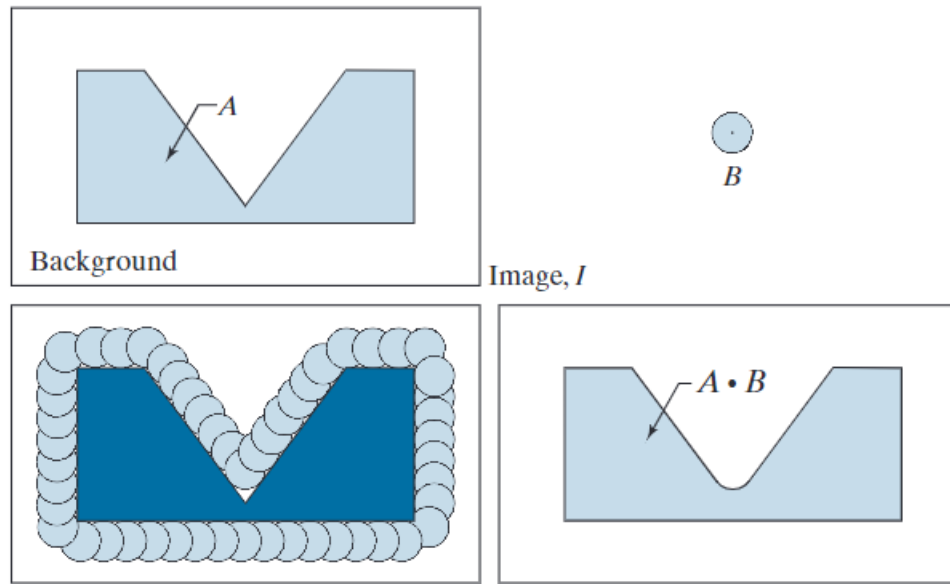


图像形态学 – 开运算和闭运算

开运算 = 腐蚀 + 膨胀
平滑轮廓、滤除clutter、断开连接

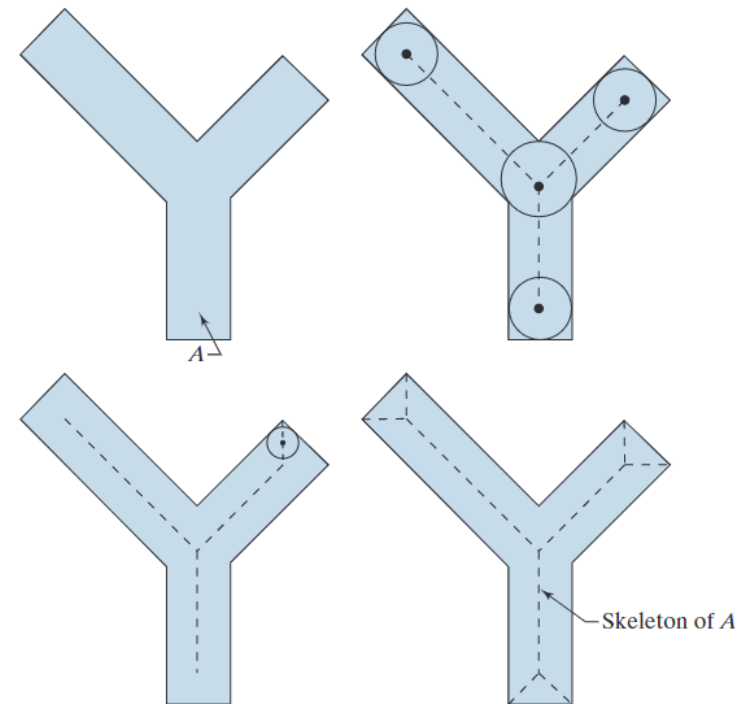
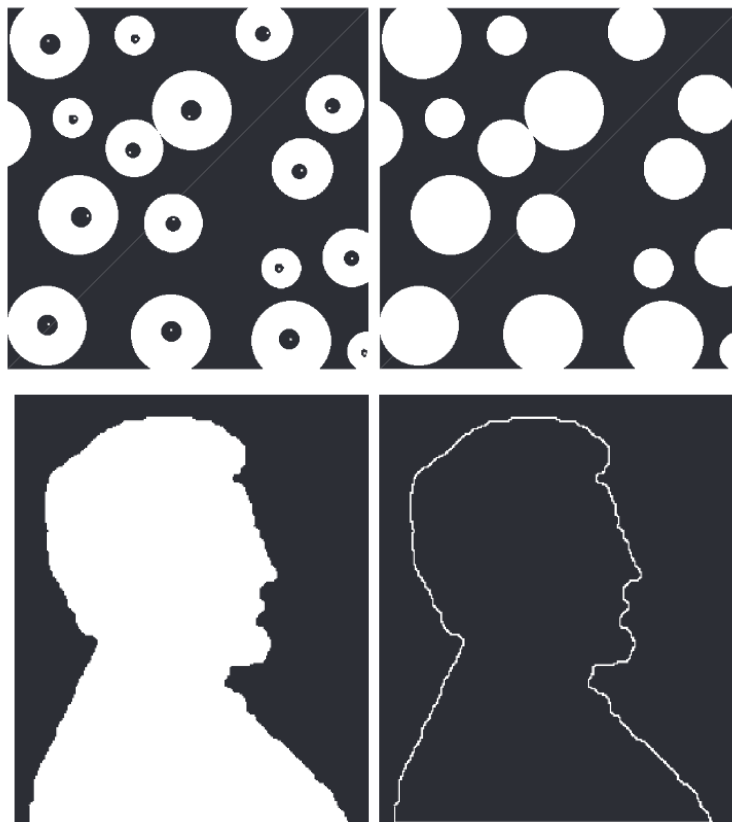


闭运算 = 膨胀 + 腐蚀
平滑轮廓、填补空洞、连接沟壑



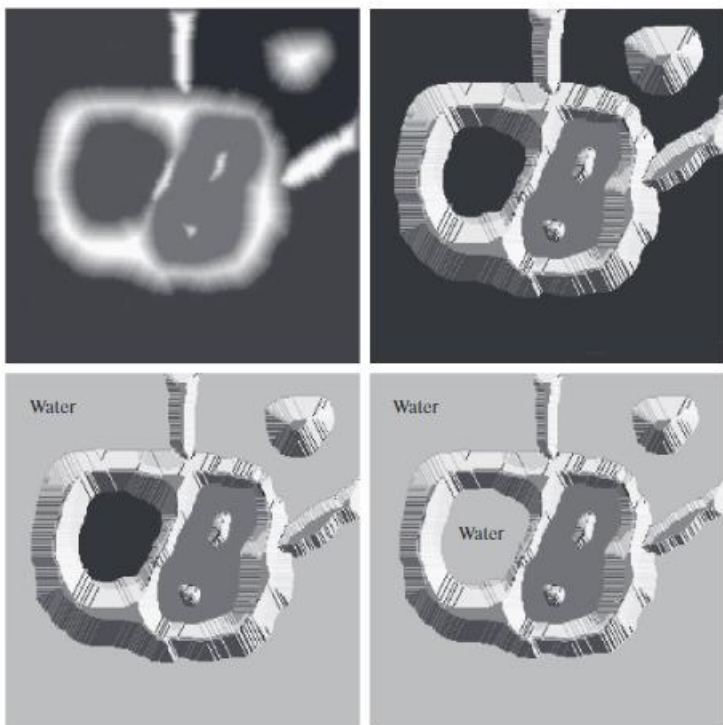
图像形态学 – 其他二值形态学

- 击中-击不中变换
- 边界提取
- 孔洞填充
- 提取连通域
- 骨架提取



图像处理 – 图像分割

- 阈值分割-OTSU
- 区域分割-分水岭算法



图像分割用于精确提取感兴趣的区域，例如提取出道路的可行驶区域。对于简单灰度场景，传统的图像分割算法有很多，这里只提两个比较重要的，分别是大津法和分水岭算法，其中大津法的适用情况和改进大家可以详细了解一下。



图像处理 – 边缘检测

- 边缘检测算子:

- Canny
- Sobel
- Direch
- 拉普拉斯



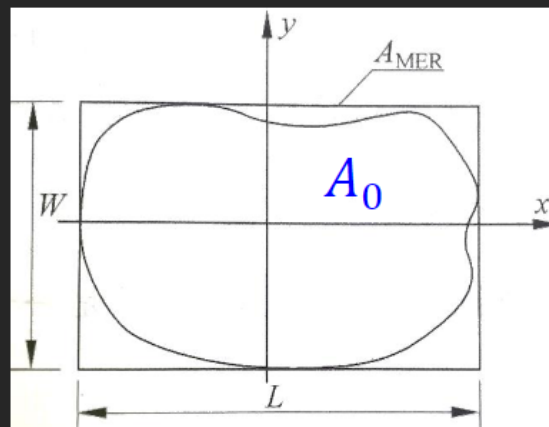
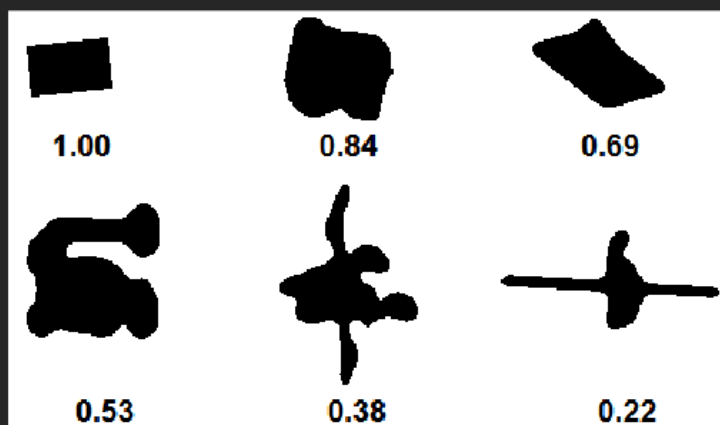
图像处理 – 连通域特征

Rectangularity(矩形度)

The ratio of the area of the object to the area of the minimum enclosing rectangle (MER)

$$R = \frac{A_0}{A_{MER}}$$

$R \in [0,1]$. $R = 1$ for a rectangle object. The rectangularity for a circular object is $\pi/4$, and slender and curved objects have smaller R values.



图像处理 – 连通域特征

Roundness (圆形度)

Roundness can describe the complexity of the boundary of an object.

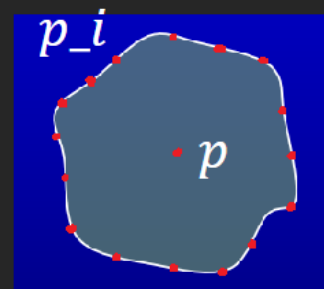
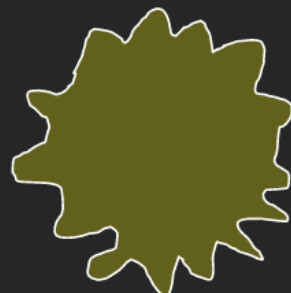
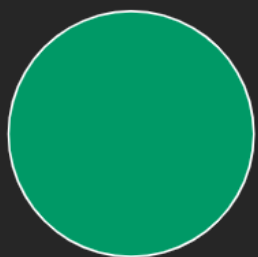
Ratio of perimeter(周长) square and area:

$$C = \frac{P^2}{4\pi A}$$

$$\begin{aligned} \text{Distance} &= \frac{1}{A} \sum \|p - p_i\| \\ \text{Sigma}^2 &= \frac{1}{A} \sum (\|p - p_i\| - \text{Distance})^2 \\ \text{Roundness} &= 1 - \frac{\text{Sigma}}{\text{Distance}} \\ \text{Sides} &= 1.4111 \left(\frac{\text{Distance}}{\text{Sigma}} \right)^{0.4724} \end{aligned}$$

设 p 为区域中心点(质点), p_i 为轮廓上全部像素点, A 为轮廓面积. Distance为轮廓上像素点到中心的平均距离, Sigma为轮廓像素点到中心的距离与平均距离的偏差, Roundness则表示平均值与标准差之间的关系.

P is the perimeter(周长) of the object and A is the area.



图像处理 – 连通域特征

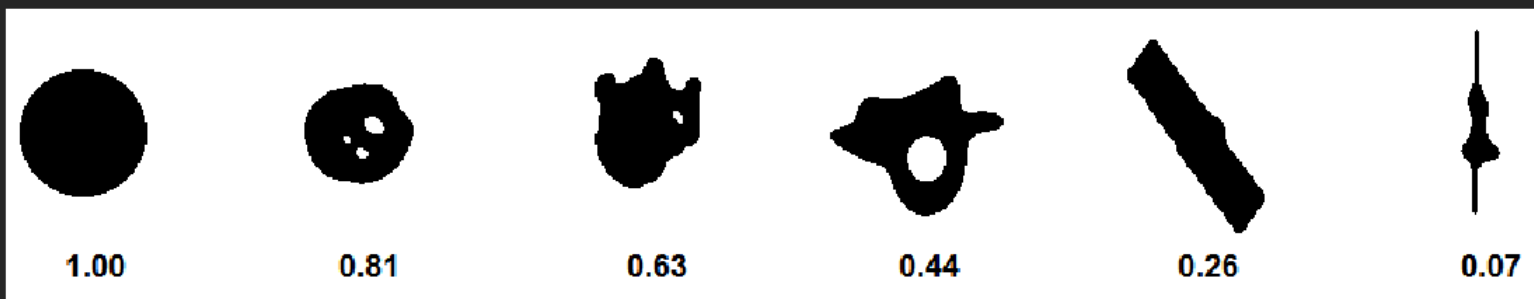
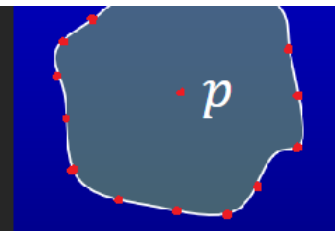
Circularity (圆形性)

Shape factor for the circularity (similarity to a circle) of a region.

$$C = \min(1, C') \quad C' = \frac{A}{(\pi d_{max}^2)}$$

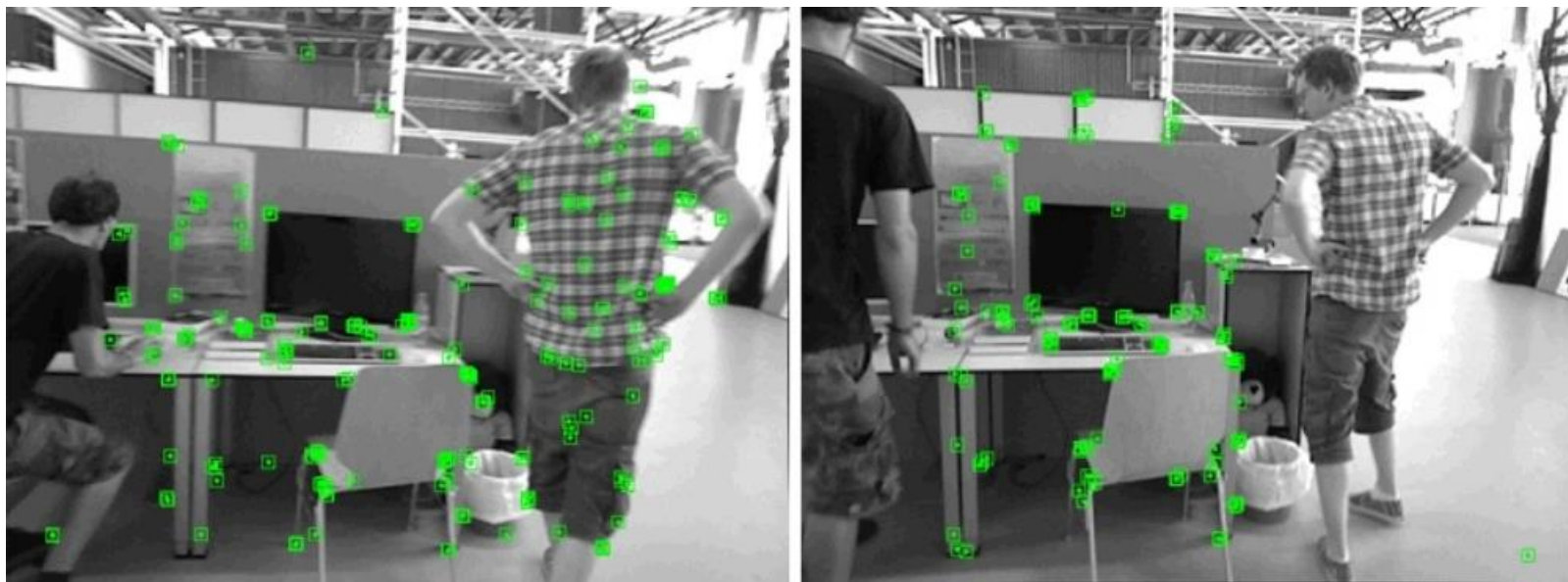
A is the area of the region and d_{max} is the maximum distance from the center to all contour pixels.

The shape factor C of a circle is 1. If the region is long or has holes, C is smaller than 1.



图像处理 – 区域\图像特征

- 特征描述子：ORB\SIFT\SURF



由于我个人并没有真正用过特征描述子，因此这部分不会讲，对视觉SLAM感兴趣的同学应该会学习到相关知识

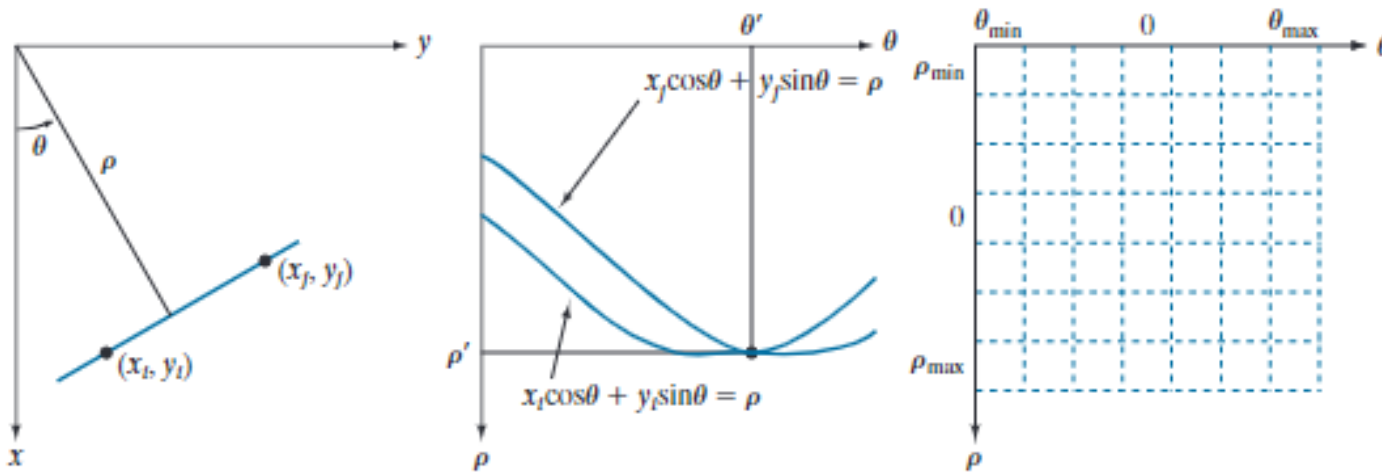


图像处理 – 形状检测

- 直线
 - LSD\Hough Transform\EDLines\RANSAC
- 圆\椭圆
 - (广义)霍夫变换\RANSAC

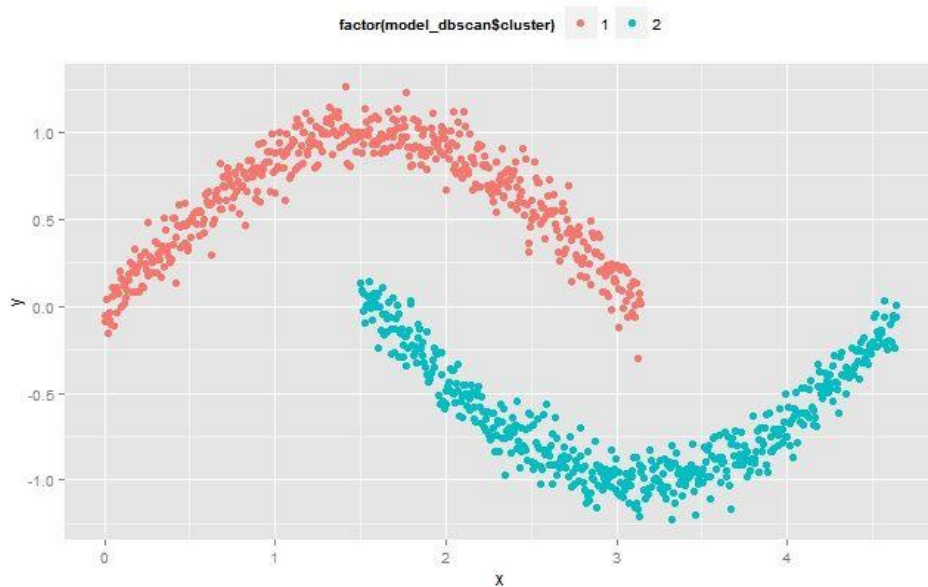
霍夫变换:

将需要匹配的曲线/直线利用变换到另一个空间的点, 利用投票/累加机制得到在该曲线上的点的数目, 进而利用阈值确定曲线



图像处理 – 其他相关算法

- 聚类： K-Means、 DBSCAN
- 拟合： 迭代最小二乘、
- 插值： 多项式插值、 双线性插值



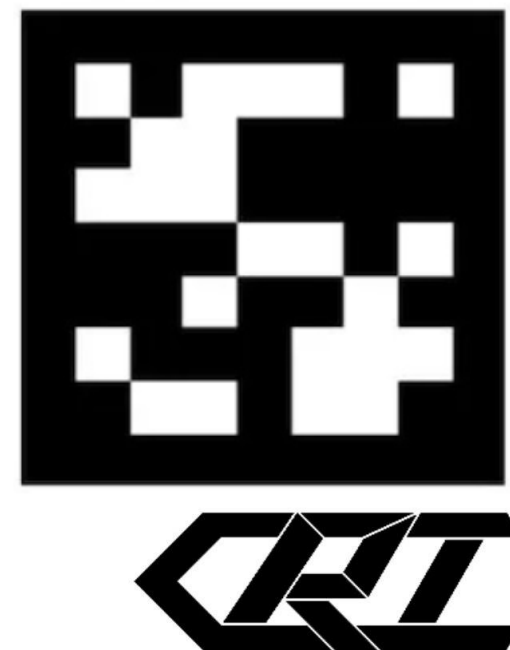
OpenCV

- 简介
- 源码结构
- 常用函数



OpenCV 源码结构

- OpenCV源码结构可以根据`opencv/modules`下面的文件夹划分, 这里只介绍几个常用的模块:
 - Calib3d : 相机标定、畸变校正和3d重建
 - Core : 数据结构
 - Highgui : 图形化界面
 - Imgcodecs : 图像编解码
 - Imgproc : 图像处理模块
 - Video : 视频分析,包括背景前景分离、对象跟踪
 - Videoio : 视频编解码读写
 - Aruco(contrib) : 视觉标签



OpenCV 常用函数

IO操作



- ☐ imread()
- ☐ VideoCapture()
- ☐ imwrite()
- ☐ VideoWriter()
- ☐ FileStorage()

GUI



- ☐ Circle
- ☐ line
- ☐ ellipse
- ☐ rectangle
- ☐ fillPoly
- ☐ putText
- ☐ createTrackbar
- ☐ setMouseCallback

图像操作



- ☐ LUT
- ☐ cvtColor
- ☐ convertTo
- ☐ split
- ☐ merge
- ☐ minMaxLoc
- ☐ reshape
- ☐ mean
- ☐ meanStdDev
- ☐ max()/min()
- ☐ threshold

图像滤波



- ☐ filter2D
- ☐ blur
- ☐ GaussianBlur()
- ☐ medianBlur()
- ☐ convertScaleAbs()
- ☐ Sobel()
- ☐ Scharr()
- ☐ Laplacian
- ☐ Canny

形态学



- ☐ connectedComponents
- ☐ connectedComponentsWithStats
- ☐ getStructuringElement()
- ☐ erode
- ☐ dilate
- ☐ morphologyEx
- ☐ floodFill
- ☐ watershed()



本周任务

- 了解图像处理基础知识，重点内容：
 - 图像表示，灰度、RGB、HSV颜色空间
 - 图像预处理：重点在灰度变换，尤其是直方图均衡化；滤波部分可以下周了解
 - 图像形态学基本操作：腐蚀膨胀、开闭运算、连通域提取
 - 基本区域特征
 - 直线检测：霍夫变换

OpenCV函数了解：由于大家没有学到C++的类、引用、模板这些，也没有学CMake。因此不会强制要求大家使用OpenCV，因此这部分不做太多要求，但是需要根据我们给出的这些基本的算法/理论，找到对应函数并理解可以做什么，参数的理解可以在下一周的小项目里加深一下。



作业相关

- 使用结构体和指针实现一个简单的程序，要求如下：
 - 使用结构体实现灰度图像表示和存储，可以只存储uint8_t或float类型的数据，其长和宽在运行中由用户指定
 - 实现任意一种灰度变换，如直方图均衡化，完成对图像的灰度变换，将结果拷贝到另一个结构体中
 - 实现一个打印函数，根据图像的长和宽在命令行打印出数据。
 - 提示：注意释放申请的内存；函数参数尽量减少拷贝，不做改变的使用const限定
 - 不需要卷，实现就好，因为下周小项目和最后用都是调库实现，这里只是为了熟悉C++



关于答疑

崩溃提问收费表	
态度与请求他人提供帮助时应有的不符	不帮助解决
包含「救命啊」	2000 CNY
问题描述 < 5 字	500 CNY
问题描述 < 10 字	200 CNY
问题描述 < 20 字	50 CNY
问题描述 < 20 字，有日志	10 CNY
问题描述 < 20 字，有截图，有日志	5 CNY
描述详尽，截图全面，日志到位，态度适宜	免费，尽可能帮助解决
推荐使用 paste.ubuntu.com 粘贴内容。	

请优先自己尝试解决，利用好提问模板

