

# 深度学习

计算机视觉基本理论

DAY02

# 计算机视觉基础

## 计算机视觉基础

### 计算机视觉概述

什么是计算机视觉

计算机视觉的应用

计算机视觉相关学科

计算机视觉与人工智能

### 数字图像处理基础

人眼成像原理

计算机成像原理

图像采样与分辨率

灰度级与灰度图像

彩色图像与色彩空间

颜色空间变化

常用图像处理技术

# 计算机视觉概览

---

# 什么是计算机视觉

- 计算机视觉在广义上是和图像相关的技术总称。包括图像的采集获取，图像的压缩编码，图像的存储和传输，图像的合成，三维图像重建，图像增强，图像修复，图像的分类和识别，目标的检测、跟踪、表达和描述，特征提取，图像的显示和输出等等。
- 随着计算机视觉在各种场景的应用和发展，已有的图像技术也在不断的更新和扩展。



# 计算机视觉的应用

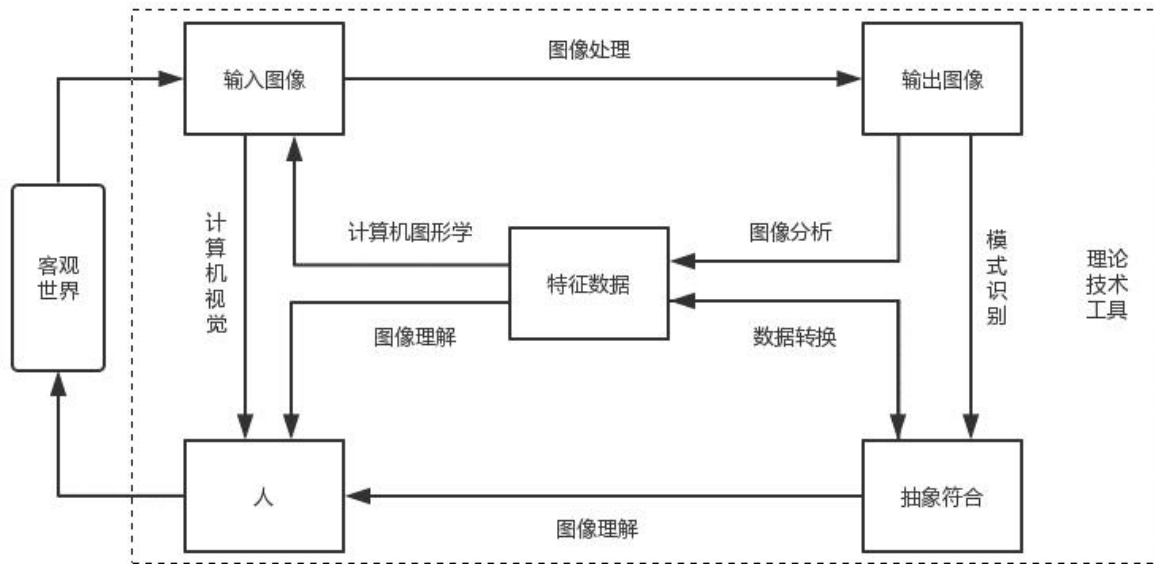
- 计算机视觉技术已经在许多领域得到了广泛的应用，以下是一些典型的例子：

- ✓ 公安安防：人脸识别，指纹识别，场景监控，环境建模。
- ✓ 生物医学：染色体分析，X光、CT图像分析，显微医学操作。
- ✓ 文字处理：文字识别，文档修复，办公自动化，垃圾邮件分类。
- ✓ 国防军事：资源探测，军事侦察，导弹路径规划。
- ✓ 智能交通：公路交通管理，电子警察执法抓拍系统，自动驾驶车辆。
- ✓ 休闲娱乐：电影特效，视频编辑，人像美颜，体感游戏，VR。



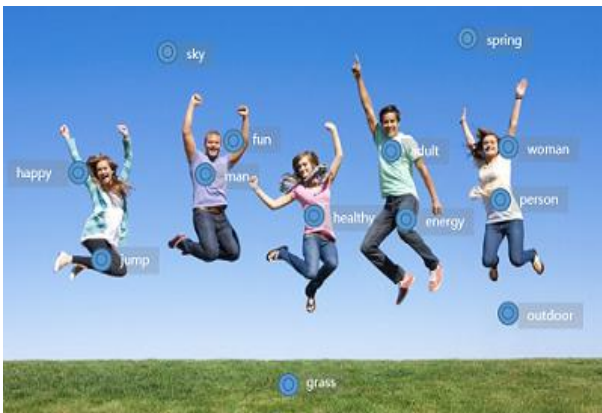
# 计算机视觉相关学科

- 计算机视觉是一门研究图像理论、技术和应用的交叉学科。计算机视觉不仅和传统的数学、物理学、生理学、心理学、计算机科学、电子工程等学科相关。并且还涉及到计算机图形学、图像模式识别、图像工程等专业技术，这几个技术名词相互关联，经常混合使用，在许多情况下他们只是专业背景不同的人习惯使用的不同称呼术语。



# 计算机视觉与人工智能

- 计算机视觉中的大部分理论运用了人工智能的技术。人工智能的发展离不开计算机视觉，计算机视觉中的很多应用问题给人工智能技术提供了研究方向。
- 人工智能在计算机视觉中最成熟的技术方向是图像识别，它实现了如何让机器理解图像中的内容。



内容识别



人脸检测

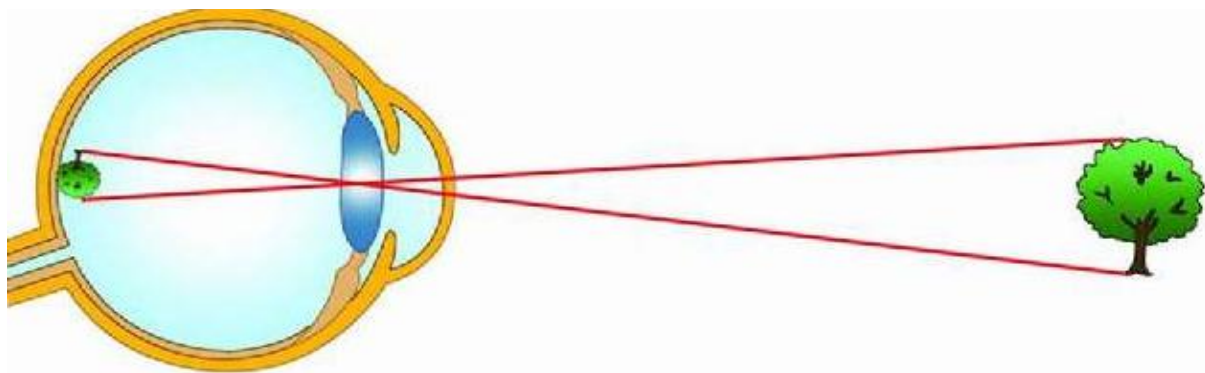
# 数字图像处理基础

---



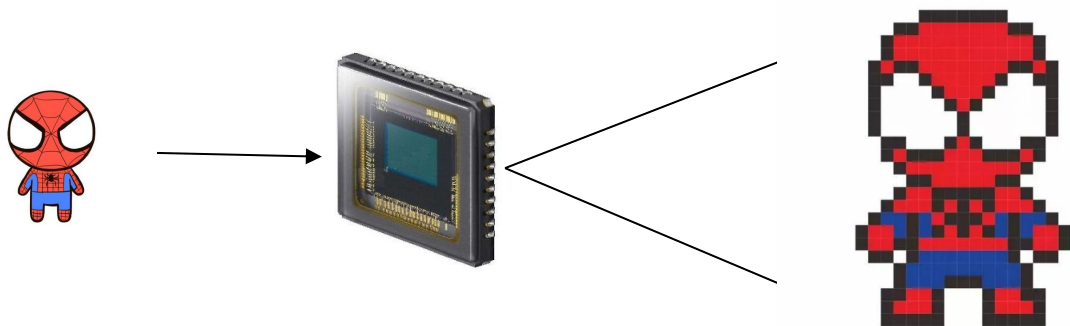
# 人眼成像原理

- 人的眼睛近似为一个球体。物体的光线通过角膜和晶状体的折射，在视网膜上成倒立缩小的实像。
- 视网膜上分布光线接收的神经细胞，分为锥状体和杆状体。每只眼睛有600万- 700万个锥状体，其对颜色灵敏度很高，负责亮光视觉。有7500万- 15000万杆状体，杆状体没有颜色感觉，负责暗视觉。



# 计算机成像原理

- 数字图像的采集过程类似人眼，使用大量的光敏传感器构成的阵列获取图像。成像的质量由传感器的单元数，尺寸和传感性能决定。
- 多数传感器的输出是连续的电压波形，图像数字化就是将一副画面的数据转换为计算机能够处理的数字形式。
- 图像数字化包括两种处理过程：采样和量化。



# 图像采样与分辨率

- 将空间上连续的图像变换成离散点的操作称为**采样**。
- 采样是按照某种时间间隔或空间间隔，采集模拟信号的过程，即空间离散化。
- 图像数字化的采样过程是将空间上连续的图像变化为离散的点。
- 采样的效果由传感器的采样间隔和采样孔径决定，采样间隔和采样孔径的大小是两个很重要的参数。



# 图像采样与分辨率（续1）

- 采样后得到离散图像的尺寸称为图像分辨率。分辨率是数字图像可辨别的最小细节。
- 分辨率由宽（width）和高（height）两个参数构成。宽表示水平方向的细节数，高表示垂直方向的细节数。
- 例如：
  - ✓ 一副640\*480分辨率的图像，表示这幅图像是由  
 $640*480=307200$ 个点组成。
  - ✓ 一副1920\*1080分辨率的图像，表示这幅图像是由  
 $1920*1080=2073600$ 个点组成。



# 图像采样与分辨率（续2）

- 采样间隔越小，所得图像像素数越多，空间分辨率高，图像质量好，但数据量大。

下图展示了lena图的分辨率从512x512依次降低到8x8的图像效果。



# 灰度级与灰度图像

- 灰度级 ( depth ) 表征了每个采样点的传感器输出中可分辨的最小变化。
- 灰度级通常是2的整数次幂。我们用m级或者n位来表示灰度级。图像数据的灰度级越多视觉效果就越好。计算机中最常用的是8位图像。
- 例如：
  - ✓ 一副8位的图像，表示每个采样点有 $2^8=256$ 级。从最暗到最亮，可以分辨256个级别。
  - ✓ 一副32级的灰度图像，每个采样点从最暗到最亮，可以分辨32个级别。



# 灰度级与灰度图像（续1）

- ✓ 量化等级越多，所得图像层次越丰富，灰度分辨率高，图像质量好，但数据量大。下图展示了lena图的灰度级从256级依次降低到4级的图像效果。

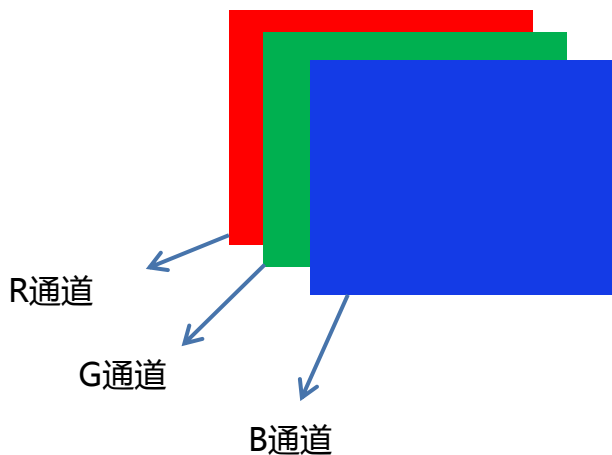













# 彩色图像与色彩空间

- 为了表征彩色图像，我们需要使用多通道数字图像。最普遍的方式是使用RGB颜色空间。RGB颜色空间中每个像素点有三个维度，分别记录在红（Red）、绿（Green）、蓝（Blue）三原色的分量上的亮度。

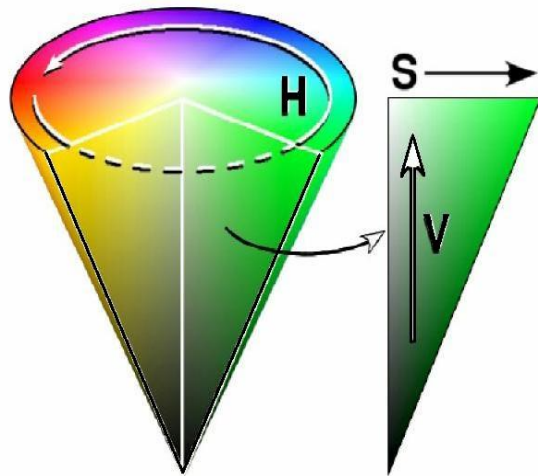


	(255,0,0)	纯红
	(0,255,0)	纯绿
	(0,0,255)	纯蓝
	(0,0,0)	纯黑
	(255,255,255)	纯白
	(124,252,0)	草坪绿
	(135,206,235)	天蓝色



# 彩色图像与色彩空间（续1）

- 另一种常用的颜色空间是HSV，该颜色空间可以用一个圆锥来表示。
- HSV表示色相(hue)、饱和度(saturation)和亮度(value)。
- H表示颜色的相位角(hue)，取值范围是0---360；S表示颜色的饱和度(saturation)，范围从0到1，它表示成所选颜色的纯度和该颜色最大的纯度之间的比率；
- V表示色彩的明亮程度(value)，范围从0到1。



# 彩色图像与色彩空间（续2）

- YUV：亮度信号Y和两个色差信号 $R - Y$ 、 $B - Y$ ，最后发送端将亮度和色差三个信号分别进行编码。采用YUV色彩空间的重要性是它的亮度信号Y和色度信号U、V是分离的。如果只有Y信号分量而没有U、V分量，那么这样表示的图就是黑白灰度图。YUV(亦称YCrCb)是被欧洲电视系统所采用的一种颜色编码方法。YUV色彩空间正是为了用亮度信号Y解决彩色电视机与黑白电视机的兼容问题，使黑白电视机也能接收彩色电视信号。与RGB视频信号传输相比，YUV只需占用极少的频宽。
- CMYK：CMYK颜色空间应用于印刷工业,印刷业通过青(C)、品(M)、黄(Y)、黑(BK)四色油墨的不同网点面积率的叠印来表现丰富多彩的颜色和阶调。
- Lab：Lab的色彩空间要比RGB模式和CMYK模式的色彩空间大，自然界中任何一点色都可以在Lab空间中表达出来。



# 颜色空间变换

- 在计算机视觉中，尤其是颜色识别相关的算法设计中，各种颜色空间混合使用是常见的方法。RGB，HSV，YUV等常见颜色空间可以通过计算公式实现相互转化，这个过程叫做颜色空间变化。颜色变换的计算公式比较复杂，通常图像处理库会提供颜色空间变化的API给用户调用。



- 色彩处理
  - ✓ 灰度化：将彩色图像转换为灰度图像
  - ✓ 二值化/反二值化：将灰度图像转换为只有两种颜色的图像
  - ✓ 色彩提取：提取指定的颜色
  - ✓ 直方图均衡化：调节图像统计直方图分布
  - ✓ 亮度、饱和度、色调调整



# 常用图像处理技术（续1）

- 形态相关操作
  - ✓ 仿射变换：旋转、平移
  - ✓ 缩放、裁剪
  - ✓ 图像相加、相减
  - ✓ 透视变换
  - ✓ 图像腐蚀、膨胀、形态学梯度



# 常用图像处理技术（续2）

- 色彩梯度
  - ✓ 模糊
  - ✓ 锐化
  - ✓ 边沿检测
- 轮廓处理
  - ✓ 轮廓查找、绘制
  - ✓ 绘制矩形、圆型、椭圆包围
  - ✓ 多边形拟合



# 图像色彩操作

## 图像色彩操作

图像色彩调整

亮度调整

饱和度调整

色调调整

图像灰度化

什么是图像灰度化

如何进行图像灰度化

二值化与反二值化

二值化

反二值化

直方图均衡化

图像直方图

直方图均衡化



# 图像色彩调整

---

# 亮度调整

- 对HSV空间的V分量进行处理可以实现对图像亮度的增强。
- 直接将彩色图像灰度化，也可以得到代表图像亮度的灰度图进行图像处理，计算量比HSV颜色空间变化低。但在HSV空间中进行处理可以得到增强后的彩色图像。



# 饱和度调整

- 对HSV空间的S分量进行处理可以实现对图像饱和度的增强。
- 饱和度的调整通常是在S原始值上乘以一个修正系数。
- 修正系数大于1，会增加饱和度，使图像的色彩更鲜明；
- 修正系数小于1，会减小饱和度，使图像看起来比较平淡。



# 色调调整

- 对HSV空间的H分量进行处理可以实现对图像色调的增强。
- 色相H的值对应的是一个角度，并且在色相环上循环。所以色相的修正可能会造成颜色的失真。
- 色相的调整通常在H原始值上加上一个小的偏移量，使其在色相环上有小角度的调整。调整后，图像的色调会变为冷色或者暖色。



# 图像灰度化

---

# 什么是图像灰度化

- 在RGB模型中，如果 $R=G=B$ 时，则彩色表示一种灰度颜色，其中 $R=G=B$ 的值叫灰度值，因此，灰度图像每个像素只需一个字节存放灰度值（又称强度值、亮度值），灰度范围为0-255。将RGB图像转换为灰度图像的过程称为图像灰度化处理。



# 如何进行图像灰度化

- 灰度化处理方法包括：
  - ✓ 分量法。将彩色图像中的三分量的亮度作为三个灰度图像的灰度值，可根据应用需要选取一种灰度图像。
  - ✓ 最大值法。将彩色图像中的三分量亮度的最大值作为灰度图的灰度值。
  - ✓ 将彩色图像中的三分量亮度求平均得到一个灰度值。
  - ✓ 根据重要性及其它指标，将三个分量以不同的权值进行加权平均。例如，由于人眼对绿色的敏感最高，对蓝色敏感最低，因此，按下式对RGB三分量进行加权平均能得到较合理的灰度图像。如：

$$f(i, j) = 0.30 R(i, j) + 0.59 G(i, j) + 0.11 B(i, j)$$



# 二值化与反二值化

---



# 二值化

- 二值化阈值处理是将原始图像处理为仅有两个值的二值图像，对于灰度值大于阈值 $t$ 的像素点，将其灰度值设定为最大值。对于灰度值小于或等于阈值的像素点，将其灰度值设定为0。



# 反二值化

- 反二值化阈值处理的结果也是仅有两个值的二值图像，对于灰度值大于阈值的像素点，将其值设定为0；对于灰度值小于或等于阈值的像素点，将其值设定为255。



# 直方图均衡化处理

---

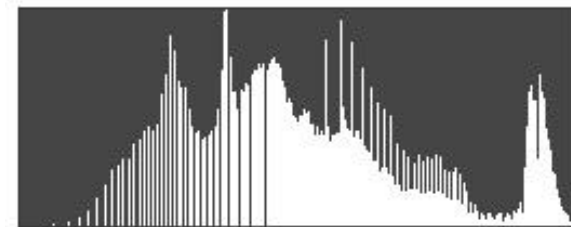
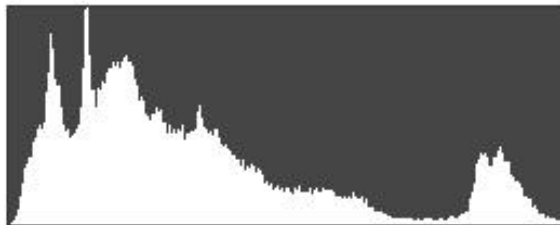
# 图像直方图

- 灰度直方图反映的是一幅图像中各灰度级像素出现的频率。以灰度级为横坐标，纵坐标为灰度级的频率，绘制频率同灰度级的关系图就是灰度直方图。它是图像的一个重要特征，反映了图像灰度分布的情况。
- 使用直方图进行图像变换是一种基于概率论的处理方法，通过改变图像的直方图，修改图像中各像素的灰度值，达到增强图像视觉效果的目的。
- 相对于灰度变化只针对单独的像素点操作，直方图变化综合考虑了全图的灰度值分布。



# 图像直方图（续1）

- 下面是两幅灰度图像的直方图，直方图的形状能反映图像的视觉效果。

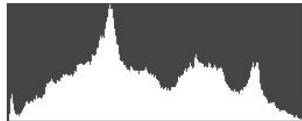


# 图像直方图（续2）

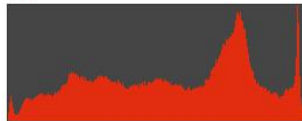
- 对于彩色图像，可以对不同的通道分别统计直方图



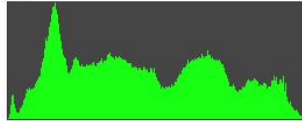
亮度



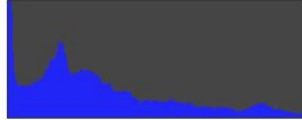
红



绿



蓝

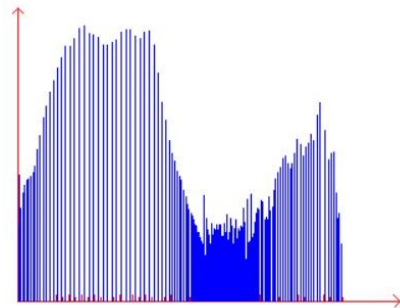
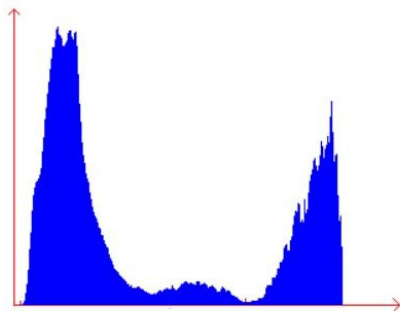


# 直方图均衡化

- 直方图均衡化将原始图像的直方图，即灰度概率分布图，进行调整，使之变化为均衡分布的样式，达到灰度级均衡的效果，可以有效增强图像的整体对比度。
- 直方图均衡化能够自动的计算变化函数，通过该方法自适应得产生有均衡直方图的输出图像。能够对图像过暗、过亮和细节不清晰的图像得到有效的增强。
- 在常用的图像处理库中，直方图操作都有API直接调用实现。



# 直方图均衡化（续）





# 图像形态操作

## 图像形态操作

仿射与透视变换

仿射变换

透视变换

图像算数计算

图像加法

图像减法

图像缩放

图像缩放

图像缩小

图像放大

腐蚀与膨胀

腐蚀

膨胀

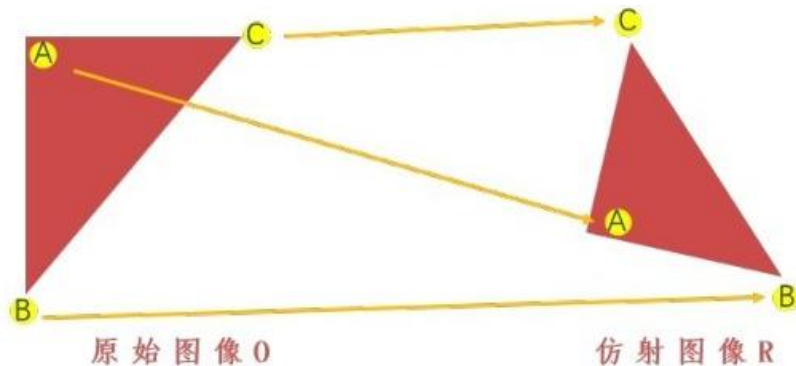
形态学梯度

# 仿射变换

---

# 什么是仿射变换

- 仿射变换是指图像可以通过一系列的几何变换来实现平移、旋转等多种操作。该变换能够保持图像的平直性和平行性。平直性是指图像经过仿射变换后，直线仍然是直线；平行性是指图像在完成仿射变换后，平行线仍然是平行线。



# 平移



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{aligned} x' &= x + \Delta x \\ y' &= y + \Delta y \end{aligned}$$



# 镜像



原图



水平镜像



垂直镜像

$$x' = -x + w$$

$$y' = -y + h$$

$$M = \begin{bmatrix} -1 & 0 & w \\ 0 & 1 & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & h \end{bmatrix}$$

# 旋转



$$\begin{aligned}x' &= x * \cos\theta - y * \sin\theta \\y' &= x * \sin\theta + y * \cos\theta\end{aligned}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



# 透视变换

- 透视变换是将图片投影到一个新的视平面，也称作投影映射。它是二维  $(x, y)$  到三维  $(X, Y, Z)$ ，再到另一个二维  $(x', y')$  空间的映射。
- 相对于仿射变换，它提供了更大的灵活性，将一个四边形区域映射到另一个四边形区域（不一定是平行四边形）。透视变换可用于图像形状校正。

The proposed technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. The pattern can be printed on a laser printer and attached to a "reasonable" planar surface (e.g., a hard book cover). Either the camera or the planar pattern can be moved by hand. The motion need not be known. The proposed approach lies between the photogrammetric calibration and self-calibration, because we use 2D metric information rather than 3D or purely implicit one. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques, the proposed technique is considerably more flexible. Compared with self-calibration, it gains considerable degree of robustness. We believe the new technique advances 3D computer vision one step from laboratory environments to the real world.

Note that Bill Triggs [22] recently developed a self-calibration technique from at least 5 views of a planar scene. His technique is more flexible than ours, but has difficulty to initialize. Liebowitz and Zisserman [14] described a technique of metric rectification for perspective images of planes using

The proposed technique only requires the camera to observe a planar pattern shown at a few (at least two) different orientations. The pattern can be printed on a laser printer and attached to a "reasonable" planar surface (e.g., a hard book cover). Either the camera or the planar pattern can be moved by hand. The motion need not be known. The proposed approach lies between the photogrammetric calibration and self-calibration, because we use 2D metric information rather than 3D or purely implicit one. Both computer simulation and real data have been used to test the proposed technique, and very good results have been obtained. Compared with classical techniques, the proposed technique is considerably more flexible. Compared with self-calibration, it gains considerable degree of robustness. We believe the new technique advances 3D computer vision one step from laboratory environments to the real world.

Note that Bill Triggs [22] recently developed a self-calibration technique from at least 5 views of a planar scene. His technique is more flexible than ours, but has difficulty to initialize. Liebowitz and Zisserman [14] described a technique of metric rectification for perspective images of planes using



# 算数运算

---



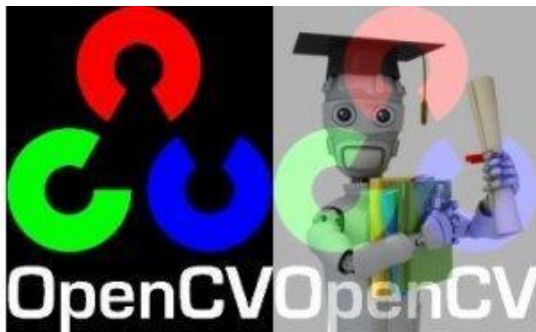
# 图像加法

相加图片像素尺寸必须一样，相加可能导致图像偏白

- 图像加法可以用于多幅图像平均去除噪声：

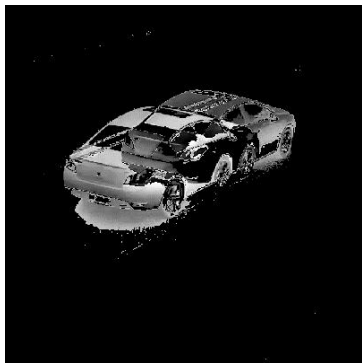
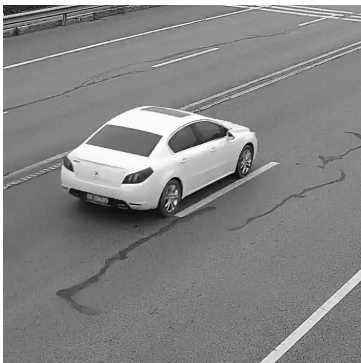


- 图像加法实现水印的叠加：



# 图像减法

- 图像减法是找出两幅图像的差异，可以在连续图像中可以实现消除背景 and 运动检测：



# 图像缩放

---

# 缩放

- 图像缩放 ( image scaling ) 是指对数字图像的大小进行调整的过程。将分辨率为(w,h)的图像, 缩放为(w', h')的图像, 水平方向系数为 $S_x = w' / w$ , 垂直方向缩放系数为 $S_y = h' / h$ 。缩放变换矩阵为:

$$A = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 示例:



原始图像



放大图像

# 图像缩小

- 图像缩小可以通过删除矩阵中的元素来实现，例如：下面的例子进行隔行、隔列删除后，高度、宽度均减小为原来的一半

1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8



1	1	1	1
3	3	3	3
5	5	5	5
7	7	7	7

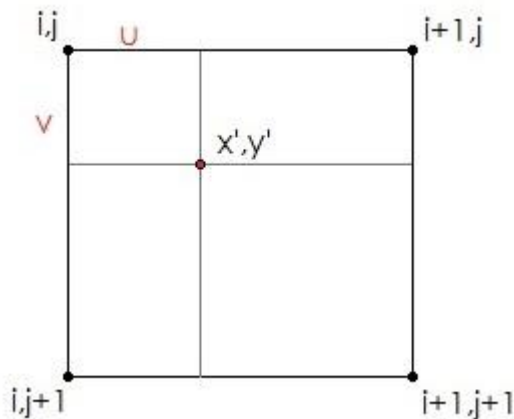


# 图像放大

• 图像放大需要进行像素插入，常用的插值法有最邻近插值法和双线性插值法

– **最邻近插值法**：直接使用新的像素点  $(x', y')$  最近的整数坐标灰度值作为该点的值，该方法计算量小，但精确度不高，并且可能破坏图像中的线性关系

– **双线性插值法**：使用新的像素点  $(x', y')$  最邻近的四个像素值进行插值计算，假设为  $(i, j)$ ,  $(i+1, j)$ ,  $(i, j+1)$ ,  $(i+1, j+1)$ , 则  $u = x' - i$ ,  $v = y' - j$ .



# 图像放大（续）

- 下图是最邻近插值法和双线性插值法的效果对比



最邻近插值



双线性插值

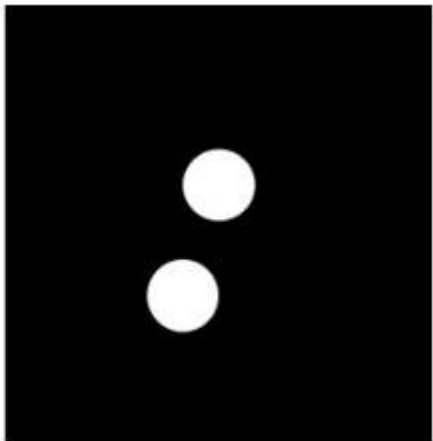
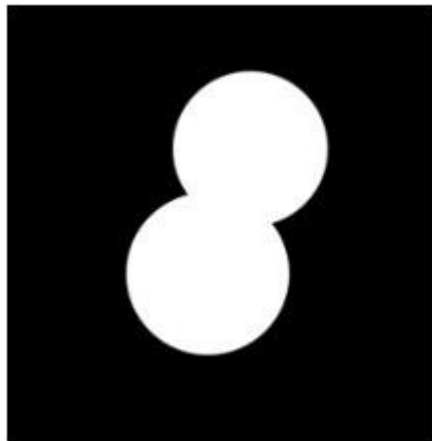
# 腐蚀与膨胀

---



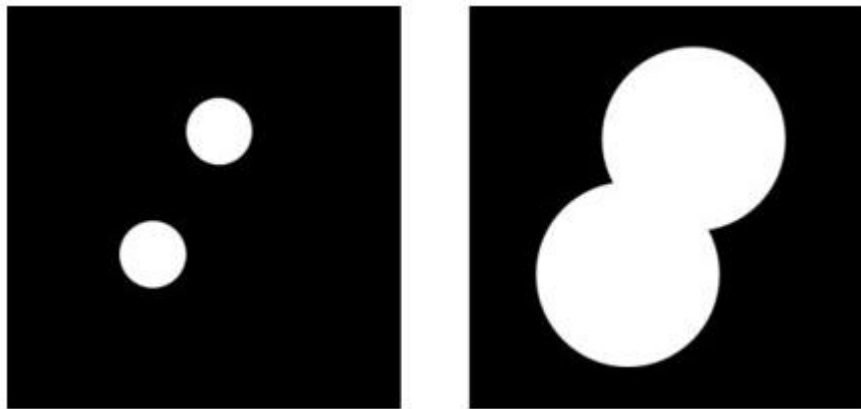
# 图像腐蚀

- 腐蚀是最基本的形态学操作之一，它能够将图像的边界点消除，使图像沿着边界向内收缩，也可以将小于指定结构体元素的部分去除。腐蚀用来“收缩”或者“细化”二值图像中的前景，借此实现去除噪声、元素分割等功能。



# 图像膨胀

- 图像膨胀(dilate)是指根据原图像的形状，向外进行扩充。如果图像内两个对象的距离较近，那么在膨胀的过程中，两个对象可能会连通在一起。膨胀操作对填补图像分割后图像内所存在的空白相当有帮助。



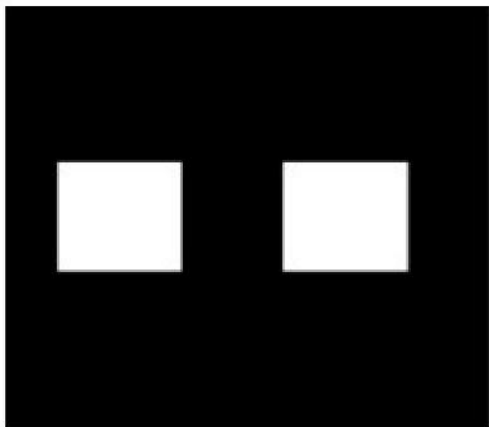
# 图像开运算

- 开运算进行的操作是先将图像腐蚀，再对腐蚀的结果进行膨胀。开运算可以用于去噪、计数等。



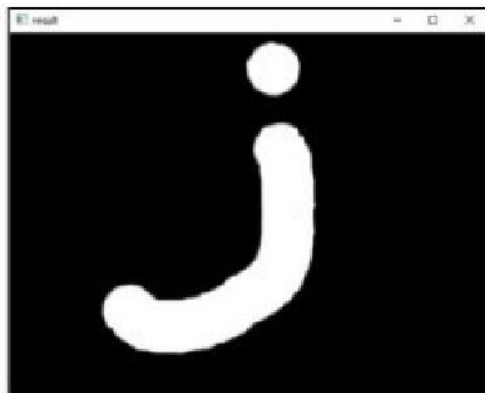
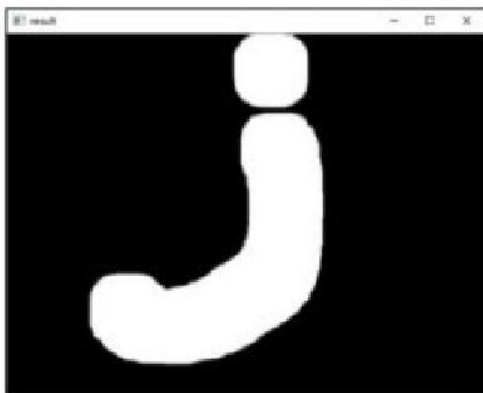
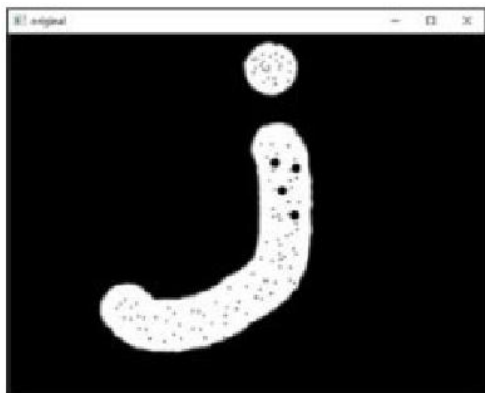
# 图像开运算（续）

- 开运算可用于取出主题图像之间细小的连接



# 图像闭运算

- 闭运算是先膨胀、后腐蚀的运算，它有助于关闭前景物体内部的小孔，或去除物体上的小黑点，还可以将不同的前景图像进行连接。

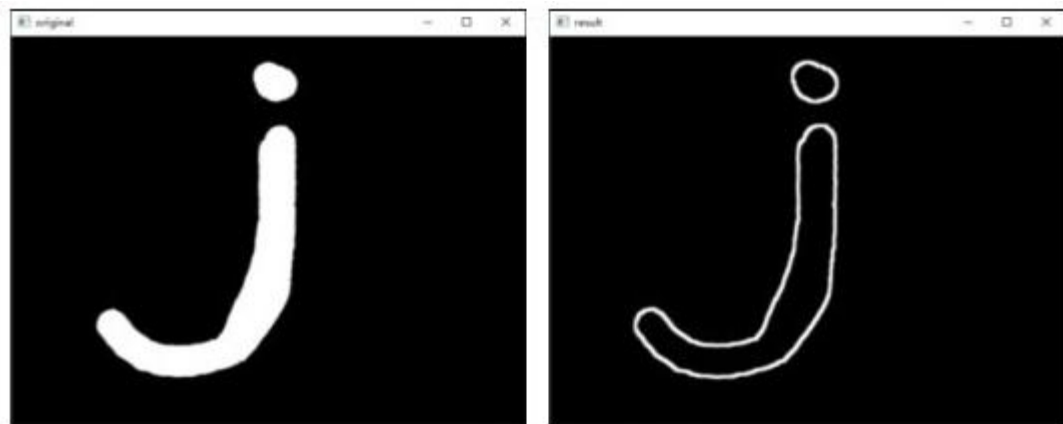


# 图像闭运算（续）



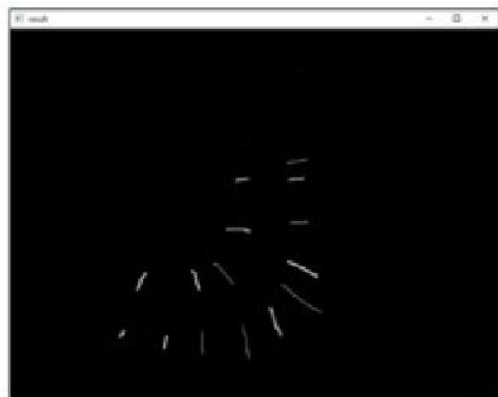
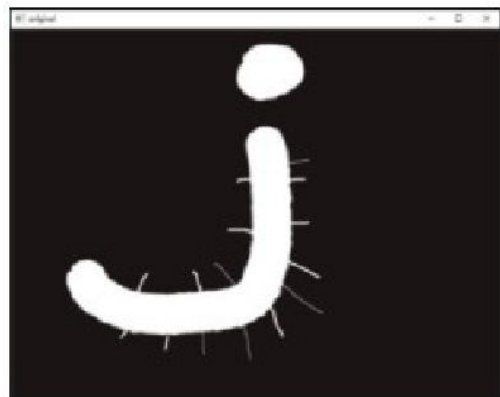
# 形态学梯度

- 形态学梯度运算是用图像的膨胀图像减腐蚀图像的操作，该操作可以获取原始图像中前景图像的边缘。



# 礼帽运算

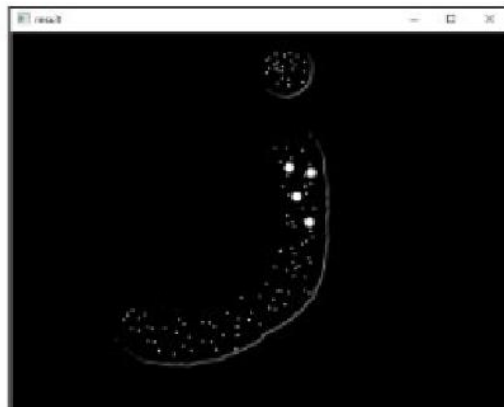
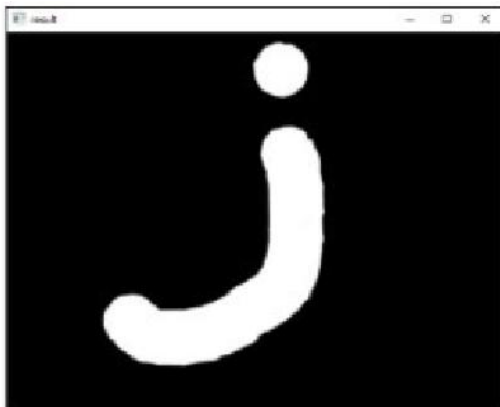
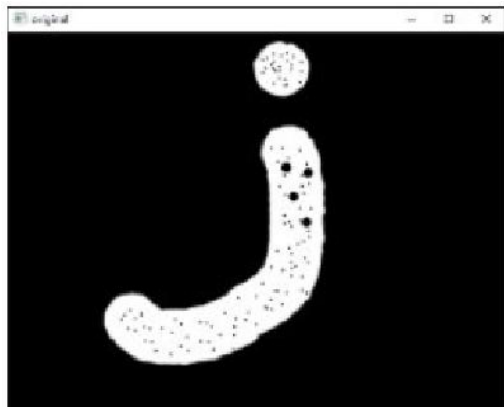
- 礼帽运算是用原始图像减去其开运算图像的操作。礼帽运算能够获取图像的噪声信息，或者得到比原始图像的边缘更亮的边缘信息。





# 黑帽运算

- 黑帽运算是用闭运算图像减去原始图像的操作。黑帽运算能够获取图像内部的小孔，或前景色中的小黑点，或者得到比原始图像的边缘更暗的边缘部分。



## 今日总结

- 图像处理基本原理
- 图像色彩处理
- 图像形态操作