计算机视觉与模式识别

苏远歧

新型计算机研究所

第四章 数字图像处理基础

数字图像处理是借助于数字计算机来处理数字图像。它有两个主要应用领域：(1)改善图示信息以便人们解释；(2)为存储、传输和表示而对数字图像进行处理，以便于机器自动理解。

一、几何变换：旋转、缩放、平移、……

二、灰度变换：图像反转、伽马矫正、直方图均衡、……

三、空间滤波：模板运算、平滑和锐化、……

四、频域滤波：离散傅里叶变换、离散余弦变换、……

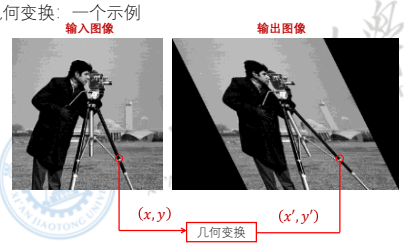
一、几何变换

4.1.1 几何变换的定义

几何变换

- A spatial transformation (also known as a geometric operation) modifies the spatial relationship between pixels in an image, mapping pixel locations in an input image to new locations in an output image.
- 几何变换改变一幅图像像素间的几何关系，将输入图像的像素位置映射到输出图像的相应位置。

4.1.2 几何变换：一个示例



4.1.3 几何变换的数学描述

几何变换

- 几何变换可以用一个函数描述：

$$(x', y') = \mathbf{G}(x, y) = (Gx(x, y), Gy(x, y))$$

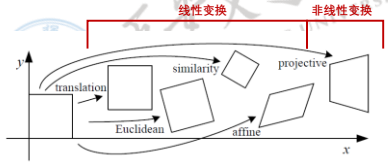
- 假设输入图像为I，输出图像为O，则有：

$$O(x', y') = I(x, y)$$

- 几何变换$\mathbf{G}(x, y)$按照形式不同可以划分成：
  - 线性
  - 非线性

4.1.4 常用的几何变换

- 常用的几何变换
  - 平移、欧几里得变换、相似变换和彷射变换
  - 透视变换



4.1.5 线性几何变换

线性几何变换

- 线性几何变换通常描述为：

$$\begin{cases} x' = Gx(x, y) = a_{11}x + a_{12}y + b_1 \\ y' = Gy(x, y) = a_{21}x + a_{22}y + b_2 \end{cases}$$

- 矩阵形式：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = A_{2\times 2} \begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{b}_{2\times 1}$$

- 逆变换：

$$\begin{bmatrix} x \\ y \end{bmatrix} = A^{-1}\left(\begin{bmatrix} x' \\ y' \end{bmatrix} - \mathbf{b}_{2\times 1} \right)$$

4.1.6 典型的线性几何变换

- 平移(b_1, b_2)：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

4.1.6 典型的线性几何变换

- 绕(x_0, y_0)点旋转θ角度：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \right) + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

4.1.6 典型的线性几何变换

- 放大s倍：

$$\begin{bmatrix} x \\ y \end{bmatrix} = s \begin{bmatrix} x' \\ y' \end{bmatrix}$$

4.1.6 典型的线性几何变换

- 欧几里得变换：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

- 相似变换：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

- 仿射变换：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = A_{2\times 2} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

4.1.7 前向变换

- 给定输入的数字图像：

$$I(x,y)$$

  - 其中$x = 1,2,\cdots,X$；$y = 1,2,\cdots,Y$；
- 前向变换：
  - 遍历输入图像I的像素(x,y)；
  - 根据几何变换，生成相应像素点(x',y')；
  - 填充输出图像O的相应位置的像素值：

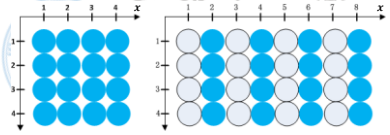
$$O(x',y') = I(x,y)$$

4.1.8 前向变换的不足

产生空洞

- 以一个简单的变换为例

$$\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}2 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$$

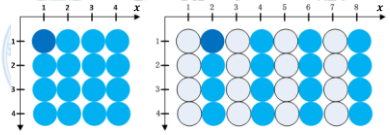


4.1.8 前向变换的不足

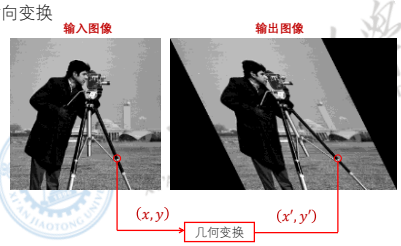
产生空洞

- 以一个简单的变换为例

$$\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}2 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$$



4.1.9 后向变换



4.1.9 后向变换

4.1.9 后向变换

- 给定输入的数字图像：

$$I(x,y)$$

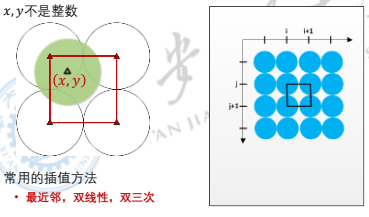
  - 其中$x=1,2,\cdots,X$；$y=1,2,\cdots,Y$；
- 后向变换：
  - 确定输出图像O的区域$[1\cdots X']\times[1\cdots Y']$
  - 遍历输出图像O的像素(x',y')；
  - 根据逆变换，寻找输入图像中的相应像素位置(x,y)；
  - 填充输出图像O的相应位置的像素值：

$$O(x',y')=I(x,y)$$

4.1.10 后向变换中的插值

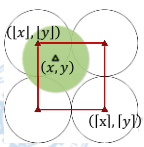
逆变换得到的像素位置(x,y)不在网格点上！

- x,y不是整数



- 常用的插值方法
  - 最近邻，双线性，双三次

4.1.10 后向变换中的插值: 双线性插值

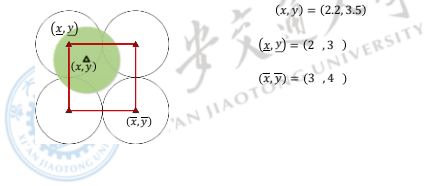


$(x,y)=(2.2,3.5)$

$([x],[y])=(2\ ,3\ )$

$([x],[y])=(3\ ,4\ )$

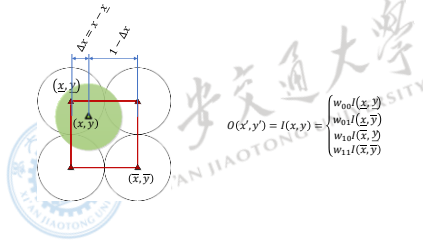
4.1.10 后向变换中的插值: 双线性插值



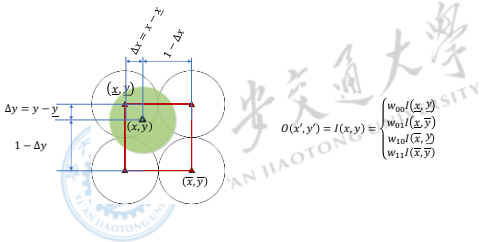
$(x,y) = (2.2, 3.5)$

$(\underline{x},\underline{y}) = (2\ ,3\ )$

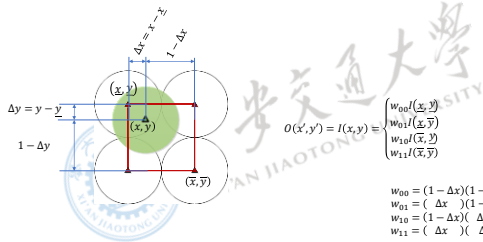
$(\overline{x},\overline{y}) = (3\ ,4\ )$



$$O(x',y') = I(x,y) = \begin{cases} w_{00}I(\underline{x},\underline{y}) \\ w_{01}I(\underline{x},\overline{y}) \\ w_{10}I(\overline{x},\underline{y}) \\ w_{11}I(\overline{x},\overline{y}) \end{cases}$$

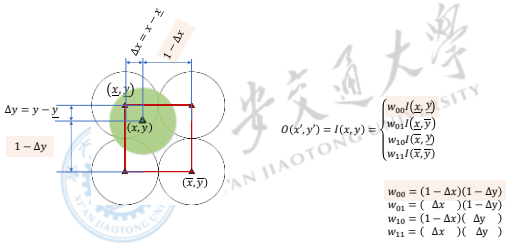


$$O(x',y') = I(x,y) = \begin{cases} w_{00}I(\underline{x},\underline{y}) \\ w_{01}I(\underline{x},\overline{y}) \\ w_{10}I(\overline{x},\underline{y}) \\ w_{11}I(\overline{x},\overline{y}) \end{cases}$$



$$O(x',y') = I(x,y) = \begin{cases} w_{00}I(\underline{x},\underline{y}) \\ w_{01}I(\underline{x},\overline{y}) \\ w_{10}I(\overline{x},\underline{y}) \\ w_{11}I(\overline{x},\overline{y}) \end{cases}$$

$w_{00} = (1 - \Delta x)(1 - \Delta y)$
$w_{01} = (\ \Delta x\ )(1 - \Delta y)$
$w_{10} = (1 - \Delta x)(\ \Delta y\ )$
$w_{11} = (\ \Delta x\ )(\ \Delta y\ )$



$$O(x', y') = I(x, y) = \begin{cases} w_{00} I(\underline{x}, \underline{y}) \\ w_{01} I(\underline{x}, \bar{y}) \\ w_{10} I(\bar{x}, \underline{y}) \\ w_{11} I(\bar{x}, \bar{y}) \end{cases}$$

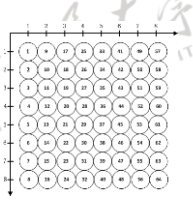
$$\begin{aligned} w_{00} &= (1 - \Delta x)(1 - \Delta y) \\ w_{01} &= (\ \Delta x\ )(1 - \Delta y) \\ w_{10} &= (1 - \Delta x)(\ \Delta y\ ) \\ w_{11} &= (\ \Delta x\ )(\ \Delta y\ ) \end{aligned}$$

4.1.11 图像金字塔中的几何变换

- 高斯图像金字塔

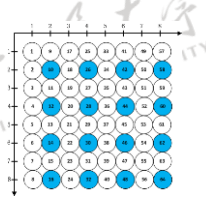
4.1.12 降采样

- 降采样对应的几何变换描述：$\begin{bmatrix} x' \\ y' \end{bmatrix} = 0.5 \begin{bmatrix} x \\ y \end{bmatrix}$；
- 逆变换为：$\begin{bmatrix} x \\ y \end{bmatrix} = 2 \begin{bmatrix} x' \\ y' \end{bmatrix}$；
- 后向变换：
  - 确定输出图像O的区域$[1 \cdots 4] \times [1 \cdots 4]$
  - 遍历输出图像O的像素(x', y')；
  - 根据逆变换，寻找输入图像中的相应像素位置(x, y)；
  - 填充输出图像O的相应位置的像素值：$O(x', y') = I(x, y)$



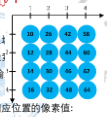
4.1.12 降采样

- 降采样对应的几何变换描述：$\begin{bmatrix} x' \\ y' \end{bmatrix} = 0.5 \begin{bmatrix} x \\ y \end{bmatrix}$；
- 逆变换为：$\begin{bmatrix} x \\ y \end{bmatrix} = 2 \begin{bmatrix} x' \\ y' \end{bmatrix}$；
- 前向变换：
  - 确定输出图像O的区域$[1 \cdots 4] \times [1 \cdots 4]$
  - 遍历输出图像O的像素(x', y')；
  - 根据逆变换，寻找输入图像中的相应像素位置(x, y)；
  - 填充输出图像O的相应位置的像素值：$O(x', y') = I(x, y)$



4.1.12 降采样

- 降采样对应的几何变换描述：$\begin{bmatrix} x' \\ y' \end{bmatrix} = 0.5 \begin{bmatrix} x \\ y \end{bmatrix}$；
- 逆变换为：$\begin{bmatrix} x \\ y \end{bmatrix} = 2 \begin{bmatrix} x' \\ y' \end{bmatrix}$；
- 前向变换：
  - 确定输出图像O的区域；
  - 遍历输出图像O的像素；
  - 根据逆变换，寻找输入图像中的相应像素位置(x, y)；
  - 填充输出图像O的相应位置的像素值：$O(x', y') = I(x, y)$



- 降采样对应的几何变换描述：$\begin{bmatrix} x' \\ y' \end{bmatrix} = 0.5 \begin{bmatrix} x \\ y \end{bmatrix}$；
- 逆变换为：$\begin{bmatrix} x \\ y \end{bmatrix} = 2 \begin{bmatrix} x' \\ y' \end{bmatrix}$；
- 前向变换：
  - 确定输出图像O的区域$[1 \cdots 4] \times [1 \cdots 4]$
  - 遍历输出图像O的像素(x', y')
  - 根据逆变换，寻找输入图像中的相应像素位置(x, y)
  - 填充输出图像O的相应位置的像素值$O(x', y') = I(x, y)$

二、灰度变换

4.2.1 灰度变换的定义

灰度变换

- 灰度变换是所有图像处理中最为简单的技术之一；
- 它把单个像素的值$I(x, y)$映射到另外一个值；
- 令$s = I(x, y)$，这个过程可以描述为：

$$t = T(s)$$

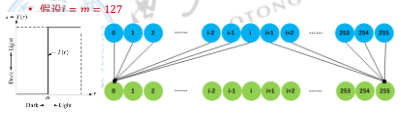
- 常用的灰度变换：
  - 阈值处理函数，图像反转，对比度拉伸函数，对数变换；
  - 分段线性变换、比特平面分层；
  - **伽马校正；**
  - **直方图均衡；**

4.2.2 灰度变换的实现形式

- 灰度变换常用查找表的形式实现；
- 以像素深度为8个比特的灰度图像为例，像素值有256种可能性：

$$s \in \{0, 1, 2, \cdots, 255\}$$

- 灰度变换$t = T(s)$则可以转化为查找表：
  - 假设$t = m = 127$



4.2.2 灰度变换的实现形式

- 灰度变换常用查找表的形式实现；
- 以像素深度为8个比特的灰度图像为例，像素值有256种可能性：

$$s \in \{0,1,2,\cdots,255\}$$

- 灰度变换$t = T(s)$则可以转化为查找表：
  - 假设$t = m = 127$

4.2.3 常用灰度变换：图像反转

图像反转

- **图像反转的形式为：**

$$t = T(s) = 255 - s$$

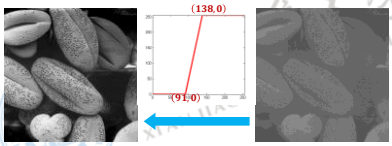
4.2.4 灰度变换在图像增强中的应用

图像增强

- 视觉上最具吸引力的图像应用之一
- 图像增强的意义：
  - 图像中的细节分辨不清；
  - 由对比度的不足带来的；
    - 图像成像时曝光不足或过度
    - 由于成像设备的非线性
    - 图象记录设备动态范围太窄等因素
- 灰度变换实现图像增强的原理：
  - 按需扩展/压缩图像的灰度等级；
  - 将感兴趣的灰度范围扩展，相对抑制不感兴趣的灰度区域。

4.2.4 灰度变换在图像增强中的应用

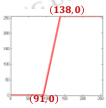
图像增强



4.2.5 灰度等级的扩展/压缩图像

图像等级的扩展/压缩

- 给定连续的灰度变换函数$t = T(s)$，针对灰度区间$[s_1, s_2]$
  - 如果$|T(s_2) - T(s_1)| > |s_2 - s_1|$，该区间的灰度等级被**扩展**
  - 如果$|T(s_2) - T(s_1)| < |s_2 - s_1|$，该区间的灰度等级被**压缩**
- 以刚才的灰度变换函数为例：
  - 在区间[0,91]，灰度等级被**压缩**
  - 在区间[91,138]，灰度等级被**扩展**
  - 在区间[138,255]，灰度等级被**压缩**

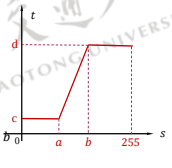


4.2.6 基于分段线性函数的灰度变换

分段线性函数

- 假设输入图像感兴趣的灰度范围为$[a, b]$，我们将它映射到输出图像为$[c, d]$
- 设计原则
  - 将感兴趣的灰度范围线性扩展，
  - 相对抑制不感兴趣的灰度区域。

$$t = \begin{cases} d & s > b \\ \frac{d-c}{b-a}(s-a) + c & a \le s \le b \\ c & s < a \end{cases}$$



4.2.7 分段线性灰度变换的示例

分段线性函数

- 假设输入图像感兴趣的灰度范围为$[a, b]$，我们将它映射到输出图像为$[c, d]$

$$t = \begin{cases} \frac{255-d}{255-b}(s-b) & s > b \\ \frac{d-c}{b-a}(s-a) & a \le s \le b \\ \frac{c}{a}s & s < b \end{cases}$$

4.2.8 分段线性灰度变换的特点

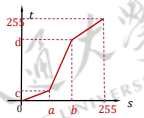
- 分段线性灰度变换的特点：
  - 形式简单，灵活度高；
  - 可以设计成连续的函数，但无法光滑；
  - 需要人工干预；
- 非线性的灰度变换
  - 对数变换

$$t = c\ln(1+s),\ t = a + \frac{\ln(1+s)}{b\ln c}$$

  - 指数变换

$$t = e^{c(s-a)} - 1$$

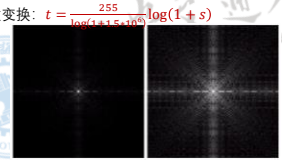
  - 其中，a, b, c是可变参数，根据实际需要调节



4.2.9 对数变换

给定一个傅里叶的频谱，它的幅度$s \in [0,1.5*10^6]$

- 如果采用线性变换：$t = \frac{255}{1.5*10^6}s$
- 对数变换：$t = \frac{255}{\log(1+1.5*10^6)}\log(1+s)$

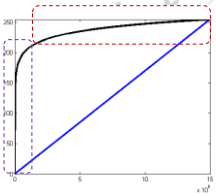


线性变换　　对数变换

4.2.10 对数变换的特点

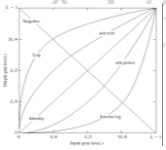
对数变换：

- **低灰度区（暗部）扩展**
- **高灰度区（亮部）压缩**



4.2.11 一些灰度变换函数的总结

- 灰度变换
  - 相等，图像反转
  - 对数，指数，n次幂，n次根
- 从扩张/压缩的角度
  - 相等和反转函数不改变原始图像的对比度
  - 对数、n次根在低灰度区扩展，高灰度区压缩
  - 指数、n次幂则相反



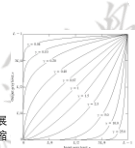
4.2.12 幂律变换

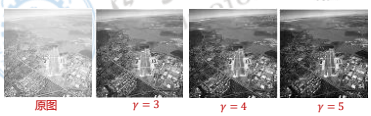
幂律变换

- 幂律变换的形式：

$$t = s^{\gamma}$$

  - 其中s,t均归一化到$[0,1]$之间
  - $\gamma = 1$，$t = s$
  - $\gamma > 1$，整体变暗，低灰度区压缩，高灰度区扩展
  - $\gamma < 1$，整体变亮，低灰度区扩展，高灰度区压缩





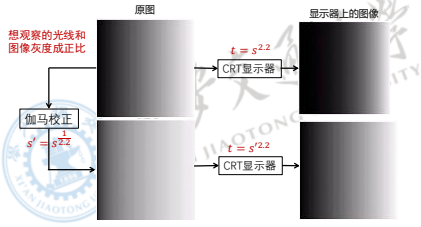
原图　　$\gamma = 3$　　$\gamma = 4$　　$\gamma = 5$

4.2.13 幂律变换的应用——伽马校正

伽马校正（Gamma Correction）

- 图像文件中的RGB数值被转换成模拟信号并且驱动阴极射线管（**C**athode **R**ay **T**ube）中的电子枪；
- CRT产生的光线与电压的指数成正比，这个指数成为伽马（**Gamma**），符号为γ；
- γ值大概在2.2左右；
- 灰度矫正的必要性
  - 设想一下，灰度图像的值$I(x,y)$正比于我们看到的光线
  - 如果我们想让电视机正确显示图像，应该怎么做？

4.2.14 伽马校正对CRT显示的影响



不做伽马矫正，则图像整体显示偏黑，低灰度区域对比度降低

4.2.15 伽马校正的影响

伽马矫正作为一种图像处理方法

- 可以改善图像质量；
- 调整此功能可使画面中较暗的部分层次分明、细节清晰可辨；
- [illegible]果。

伽马校正校正前图像　　伽马校正校正后图像

4.2.16 图像灰度直方图

直方图

- 数字图像处理中最简单且有效的工具；
- 对图像的分析、观察直至处理，均离不开直方图；
- **灰度直方图**是灰度等级的函数，
  - 描述的是图像中该灰度等级的**像素个数**
  - 该灰度等级的**像素出现频率**；
  - 横坐标表示灰度等级，纵坐标表示图像中该灰度等级出现的个数或该灰度级像素出现的频率
- 反映了图像灰度分布的情况；

4.2.17 图像灰度直方图：一个示例

构建直方图

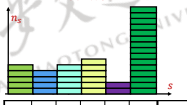
- 遍历图像，统计每个灰度等级的像素个数

输入图像

灰度等级：1,2,3,4,5,6

1	2	3	4	5	6
6	4	3	2	2	1
1	6	6	4	6	6
3	4	5	6	6	6
1	4	6	6	2	3
1	3	6	4	6	6

灰度直方图



1	2	3	4	5	6	灰度等级
5	4	5	6	2	14	像素个数

4.2.17 图像灰度直方图：一个示例

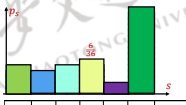
计算像素频率：$p_s = \frac{n_s}{\sum_s n_s}$

输入图像

灰度等级：1,2,3,4,5,6

1	2	3	4	5	6
6	4	3	2	2	1
1	6	6	4	6	6
3	4	5	6	6	6
1	4	6	6	2	3
1	3	6	4	6	6

灰度直方图



1	2	3	4	5	6	灰度等级
$\frac{5}{36}$	$\frac{4}{36}$	$\frac{5}{36}$	$\frac{6}{36}$	$\frac{2}{36}$	$\frac{14}{36}$	像素频率

4.2.18 图像灰度直方图的性质

灰度直方图

- 灰度直方图只能反映图像的灰度分布情况，而不能反映图像像素的位置，即所有的空间信息全部丢失；
- 一幅图像对应唯一的灰度直方图，反之不成立；不同的图像可对应相同的直方图；

4.2.18 图像灰度直方图的性质

灰度直方图的可加性

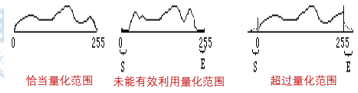
- 当灰度直方图统计的是像素个数的时候，它具有可加性；

图像的直方图H =区域Ⅰ的直方图H_1 + 区域Ⅱ的直方图H_2

4.2.19 图像灰度直方图的应用

灰度直方图的应用

- **判断量化是否恰当**：直方图给出了简单可见的指示，用来判断一幅图象是否合理的利用了全部被允许的灰度级范围
  - 一般一幅图应该利用全部或几乎全部可能的灰度级，否则等于增加了量化间隔。丢失的信息将不能恢复。

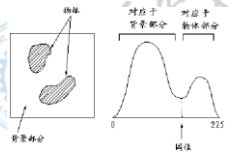


恰当量化范围　未能有效利用量化范围　超过量化范围

4.2.20 图像灰度直方图的应用

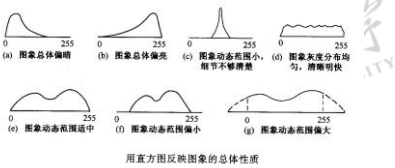
灰度直方图的应用

- 边界阈值选取（确定图像二值化的阈值）
  - 假设某图象的灰度直方图具有二峰性，则表明这个图象的较亮的区域和较暗的区域可以较好地分离，以这一点为阈值点，可以得到好的２值处理的效果。



$$O(x,y)=\begin{cases}0 & I(x,y)\le T\\ 255 & I(x,y)>T\end{cases}$$

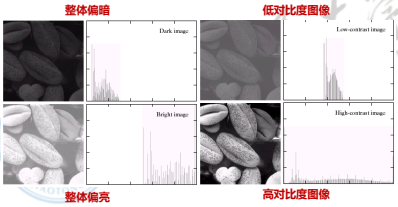
4.2.21 看懂灰度直方图



用直方图反映图象的总体性质

- 直方图反映的总体性质:
  明暗程度、细节是否清晰、动态范围大小等

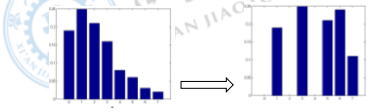
4.2.22 四类典型的直方图



4.2.23 直方图均衡

直方图均衡

- 图像处理领域中利用图像直方图对对比度进行调整的方法
- 这种方法通常用来增加许多图像的全局对比度，尤其是当图像的有用数据的对比度相当接近的时候
- 通过这种方法，亮度可以更好地在直方图上分布



4.2.24 直方图均衡算法

直方图均衡

- 给定灰度图像，让表示p_s灰度s，$s = 0,1,\cdots,255$的像素的出现的频率
- 累计概率：$c_s = \sum_{i=0}^{s} p_s, s = 0,1,\cdots,255$
  - 累计概率总是位于0到1之间
- 灰度变换函数：

$$t = T(s) = \text{Round}(255 \times c_s)$$

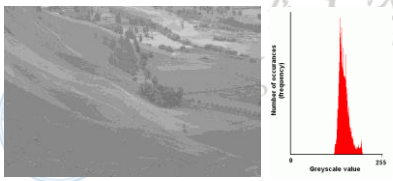
- 通常采用：

$$t = T(s) = \text{Round}\big((\max - \min) \times c_s + \min\big)$$

  - 其中max和min是原始图像的最大和最小像素值

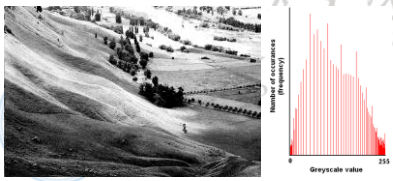
4.2.25 直方图均衡示例

未经直方图均衡的原始图像



4.2.25 直方图均衡示例

直方图均衡后的图像



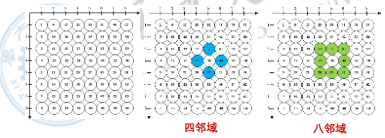
三、空间滤波

空间滤波（空域滤波）

- 按其特点划分：线性和非线性；
  - 线性滤波器的设计常基于对傅里叶变换的分析
  - 非线性空间滤波器常对图像的邻域进行直接操作

空间滤波（空域滤波）

- 按其特点划分：线性和非线性；
  - 线性滤波器的设计常基于对傅里叶变换的分析
  - 非线性空间滤波器常对图像的**邻域**进行直接操作



空间滤波（空域滤波）

- 按其特点划分：线性和非线性；
  - 线性滤波器的设计常基于对傅里叶变换的分析
  - 非线性空间滤波器常对图像的邻域进行直接操作
- 按其功能划分：平滑和锐化
  - 平滑用低通滤波器实现
  - 锐化用高通滤波器实现
- 空间滤波器的工作原理可以借助于傅里叶变换进行分析
  - 傅里叶空间的某个范围的分量受到抑制，而其他分量不受影响
  - 改变输出图的频率分布

4.3.2 空间滤波的作用

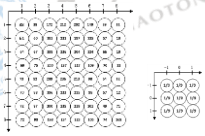
空间滤波（空域滤波）

- 图像增强/图像恢复
- 提取图像的结构

4.3.3 线性空间滤波

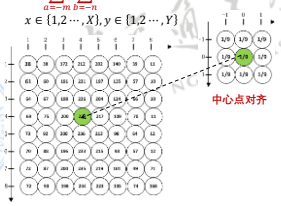
模板运算

- 大小为$X \times Y$的图像I，
  $I(x,y), x \in \{1,2 \cdots, X\}, y \in \{1,2 \cdots, Y\}$
- 大小为$(2m+1) \times (2n+1)$的模板f
  $f(a,b), a \in \{-m \cdots 0 \cdots m\}, b \in \{-n \cdots 0 \cdots n\}$



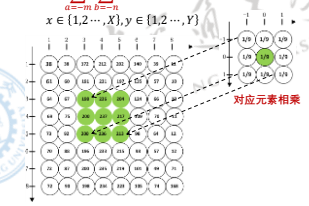
4.3.3 线性的空间滤波

$$O(x,y) = \sum_{a=-m}^{m} \sum_{b=-n}^{n} I(x+a, y+b) f(a,b)$$
$$x \in \{1,2 \cdots, X\}, y \in \{1,2 \cdots, Y\}$$



4.3.3 线性的空间滤波

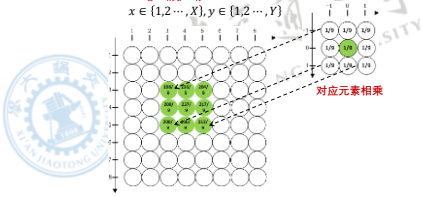
$$O(x,y) = \sum_{a=-m}^{m} \sum_{b=-n}^{n} I(x+a, y+b) f(a,b)$$
$$x \in \{1,2 \cdots, X\}, y \in \{1,2 \cdots, Y\}$$



4.3.3 线性的空间滤波

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x+a,y+b)f(a,b)$$

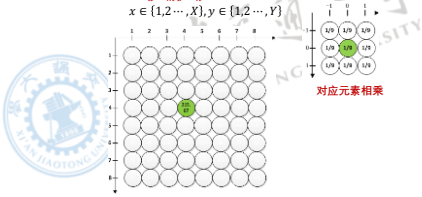
$$x\in\{1,2\cdots,X\},y\in\{1,2\cdots,Y\}$$



4.3.3 线性的空间滤波

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x+a,y+b)f(a,b)$$

$$x\in\{1,2\cdots,X\},y\in\{1,2\cdots,Y\}$$



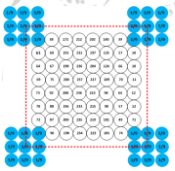
4.3.3 线性的空间滤波

模板运算

- 主要步骤：
  - 将模板在图中漫游，将模板中心与图中某个像素位置重合；
  - 将模板上系数与模板下对应像素相乘；
  - 将所有乘积相加；
  - 将和（模板的输出响应）赋给图中对应模板中心位置的像素。
- 边界条件
  - 不过当模板的边界超出图像的边界时，要注意边界问题的处理，最常用的方法是填充（padding），但其会影响图像的边界，影响程度随模板尺寸的增大而增加。

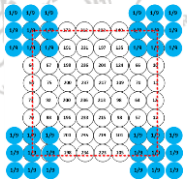
4.3.4 边界的处理

- 假设图像的尺寸是8×8的，模板的尺寸是3×3，模板运算后的图像尺寸常见的有三种：
  - **全尺寸：** 10×10
  - 同等尺寸： 8×8
  - 有效尺寸： 6×6



4.3.4 边界的处理

- 假设图像的尺寸是8×8的，模板的尺寸是3×3，模板运算后的图像尺寸常见的有三种：
  - 全尺寸： 10×10
  - **同等尺寸： 8×8**
  - 有效尺寸： 6×6

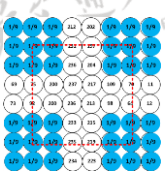


4.3.4 边界的处理

- 假设图像的尺寸是8×8的，模板的尺寸是3×3，模板运算后的图像尺寸常见的有三种：
  - 全尺寸： 10×10
  - 同等尺寸： 8×8
  - **有效尺寸： 6×6**

filter2(g, f, shape) in MATLAB

shape='full', 'same', 'valid'

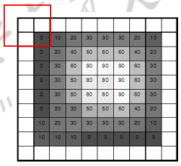


4.3.4 边界的处理

- 如果给定一幅$M \times N$大小的图像和$(2a+1) \times (2b+1)$的模板，请问模版运算后：
  - 全尺寸的图像大小：
    $(M+2a) \times (N+2b)$
  - 同等尺寸的图像大小：
    $M \times N$
  - 有效尺寸的图像大小
    $(M-2a) \times (N-2b)$

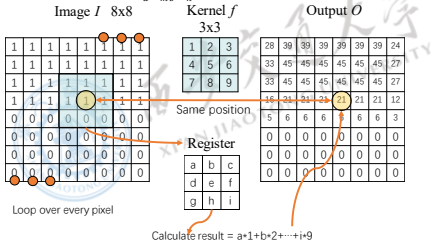
4.3.4 边界的处理

- 全尺寸和相同尺寸都涉及到图像区域之外的像，需要进行插值
- 常用方法：
  - 图像之外的区域全黑
  - 拷贝图像的边缘
  - 反射插值
  - 循环插值



模板运算

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x+a,y+b)f(a,b)$$



4. 3. 5 模板运算

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x+a,y+b)f(a,b)$$

Output Image　Input Image　Kernal Image

4. 3. 5 模板运算

- 模板运算：

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x+a,y+b)f(a,b)$$

- 卷积运算

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x-a,y-b)h(a,b)$$

4. 3. 5 模板运算与卷积

- 模板运算：

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x+a,y+b)f(a,b)$$

- 卷积运算

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x-a,y-b)h(a,b)$$

$$O(x,y)=\sum_{a'=-m}^{m}\sum_{b'=-n}^{n}I(x+a',y+b')h(-a',-b')$$

如果h是f沿着x和y轴的翻转，两者等价！

4. 3. 5 模板运算与卷积

- 模板运算：

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x+a,y+b)f(a,b)$$

- 卷积运算

$$O(x,y)=\sum_{a=-m}^{m}\sum_{b=-n}^{n}I(x-a,y-b)h(a,b)$$

$$O(x,y)=\sum_{a'=-m}^{m}\sum_{b'=-n}^{n}I(x+a',y+b')h(-a',-b')$$

模板不需要进行滤波器翻转！

4. 3. 6 模板运算与卷积的性质

- 任意一个空间相关运算总存在等价的卷积运算，卷积具有的性质可以移植到相关运算。
- 线性：
  - 可加性

$$f\otimes(h_1+h_2)=f\otimes h_1+f\otimes h_2$$

  - 比例关系

$$f\otimes(kh)=k(f\otimes h)$$

- 可交换性：

$$f\otimes h=h\otimes f$$

4. 3. 6 模板运算与卷积的性质

- 结合律：

$$f\otimes h\otimes g=f\otimes(h\otimes g)$$

  - 如果对图像进行一系列的线性滤波

$$((f\otimes h_1)\otimes h_2)\otimes h_3$$

  - 等价于采用一个模板对图像进行滤波

$$f\otimes(h_1\otimes h_2\otimes h_3)$$

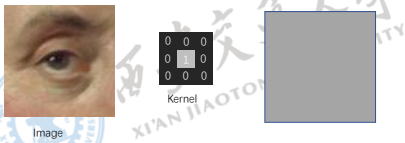
- 微分性质：

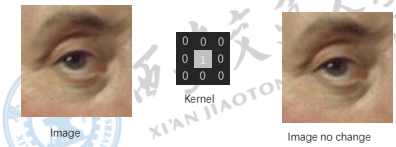
$$\frac{\partial}{\partial x}(f\otimes h)=\frac{\partial}{\partial x}f\otimes h$$

4. 3. 7 常用的空域滤波

- 用于平滑：
  - 高斯滤波器，均值滤波器；
- 用于边缘提取：
  - Sobel滤波器， prewitt算子，拉普拉斯算子
- 运动模糊：
  - 运动模糊算子

常见滤波器：冲激响应



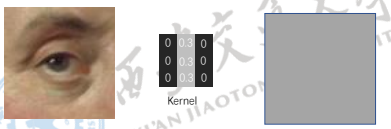


常见滤波器：平移

常见滤波器：运动模糊

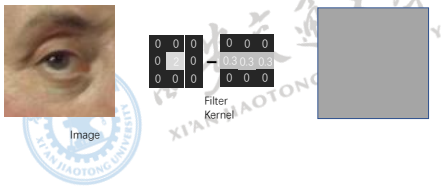
常见滤波器：运动模糊

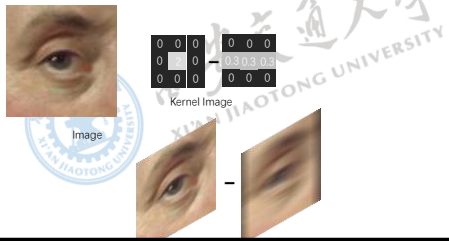
常见滤波器：运动模糊



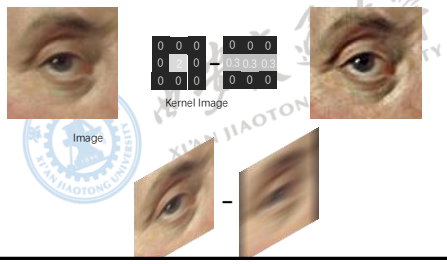
常见滤波器：运动模糊

什么功能？



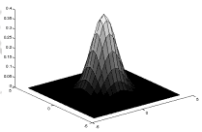


Linear Filter

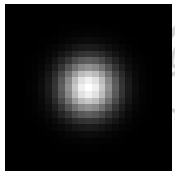


高斯模板

- 旋转对称性
- 中间高两边低
- 靠近当前像素的像素对它作用越大，越远的作用越小



An Isotropic Gaussian

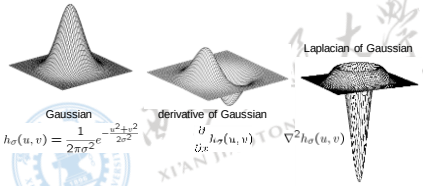


- The picture shows a smoothing kernel proportional to

$$e^{-\frac{x^2+y^2}{2\sigma^2}}$$

- (which is a reasonable model of a circularly symmetric fuzzy blob)

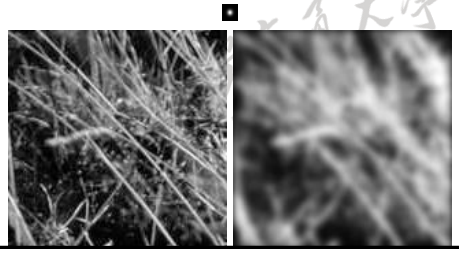
2D filters, more on this later…

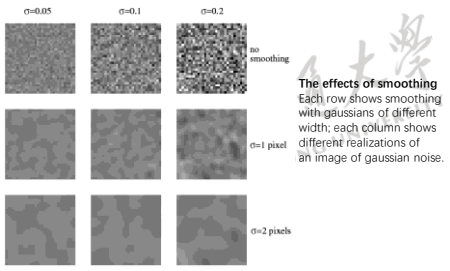


- 拉普拉斯算子:

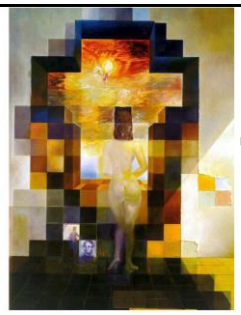
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Smoothing with a Gaussian





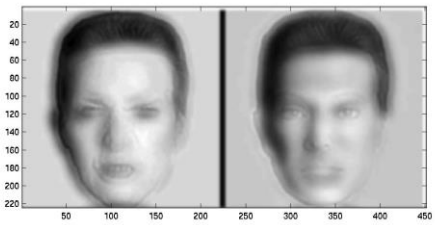
The effects of smoothing
Each row shows smoothing with gaussians of different width; each column shows different realizations of an image of gaussian noise.

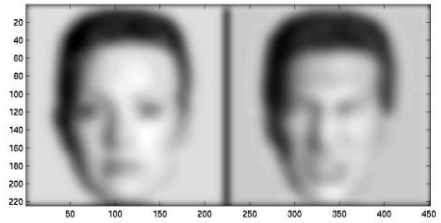


Salvador Dali, *"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln", 1976*

Salvador Dali, *"Gala Contemplating the Mediterranean Sea, which at 30 meters becomes the portrait of Abraham Lincoln", 1976*

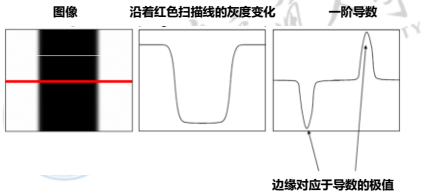
Image smoothing can remove noise, and also …





4. 3. 8 导数与边缘

- 图像的边缘常发生在图像亮度快速变换的区域。



4. 3. 8 导数与边缘

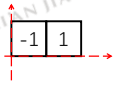
- 二维函数$f[x, y]$的偏导

$$\frac{\partial f}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon, y) - f(x, y)}{\epsilon}$$

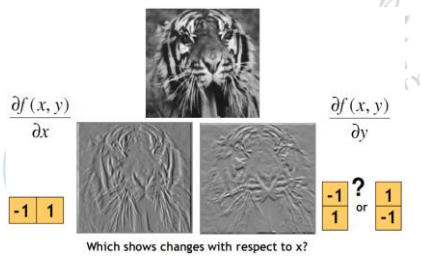
- 当$f[x, y]$是离散函数时，进行有限差分的逼近

$$\frac{\partial f}{\partial x} \approx f(x+1, y) - f(x, y)$$

- 空间相关的模板



4.3.8 导数与边缘

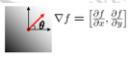


4.3.8 导数与边缘

- 二维图像$f[x,y]$的梯度
- $\nabla f \frac{\partial f}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon,y)-f(x,y)}{\epsilon}$
- 边缘对应于梯度：

$\nabla f = [\frac{\partial f}{\partial x}, 0]$

$\nabla f = [0, \frac{\partial f}{\partial y}]$



$\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$

- 梯度方向：
- 梯度强度：

4. 3. 8 导数与边缘

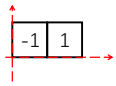
- 二维函数$f[x,y]$的偏导

$$\frac{\partial f}{\partial x} = \lim_{\epsilon \to 0} \frac{f(x+\epsilon,y) - f(x-\epsilon,y)}{2\epsilon}$$

- 当$f[x,y]$是离散函数时，进行有限差分的逼近

$$\frac{\partial f}{\partial x} \approx 0.5\big(f(x+1,y) - f(x-1,y)\big)$$

- 空间相关的模板



四、频率滤波

4.4.1 傅里叶变换的定义

傅里叶变换

- The Fourier transform is a representation of an image as a sum of complex exponentials of varying magnitudes, frequencies, and phases.
- 傅里叶变换将图像表示成一组不同幅度、频率以及相位的调谐函数之和。
- 在数字图像分析中非常重要:
  - 图像增强、图像分析、图像恢复、图像压缩。

4.4.2 离散傅里叶变换

离散傅里叶变换

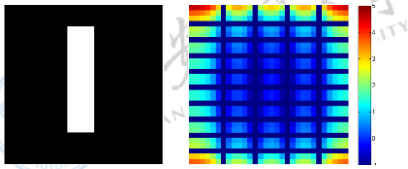
- 数字图像中采用的都是离散傅里叶变换：
  - 输入、输出都是离散的，便于计算机的计算
  - 存在一个快速计算的方法，快速傅里叶变换（**FFT**）
- 离散傅里叶变换的形式：

$$F(u,v)=\sum_{x=1}^{X}\sum_{y=1}^{Y}I(x,y)e^{-2\pi j\frac{ux}{X}}e^{-2\pi j\frac{vy}{Y}}$$

- 反变换的形式：

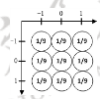
$$I(x,y)=\sum_{u=1}^{X}\sum_{v=1}^{Y}F(u,v)e^{2\pi j\frac{ux}{X}}e^{2\pi j\frac{vy}{Y}}$$

4.4.3 离散傅里叶变换的示例



离散傅里叶变换后的频谱是多少？

```
% construct a 3x3 averaging template
f = 1/9 * ones(3,3);
%  do the 2d Fourier transform
F = fft2(f)
```



离散傅里叶变换后的频谱是多少？

```
% construct a 3x3 averaging template
f = 1/9 * ones(3,3);
%  do the 2d Fourier transform
F = fft2(f)
```

1	0	0
0	0	0
0	0	0

离散傅里叶变换后的频谱是多少？

```
% construct a 3x3 averaging template
f2 = [zeros(5,11); 1/11*ones(1,11); zeros(5,11)];
%  do the 2d Fourier transform
F2 = fft2(f2)
```

0	0	0
0.3	0.3	0.3
0	0	0

离散傅里叶变换后的频谱是多少？

```
% construct a 3x3 averaging template
f1 =zeros(11,11);
f1(6,6) = 2;
%  do the 2d Fourier transform
F1 = fft2(f1)
```

0	0	0
0	2	0
0	0	0

- F1,F2,F1-F2分别是多少？
- 是否和我们先前讲解的例子对应上？
- 如果对应不上，为什么？

4.4.4 离散余弦变换的定义

离散余弦变换

- The discrete cosine transform (DCT) represents an image as a sum of sinusoids of varying magnitudes and frequencies.
- 离散余弦变换将图像表示成一组不同幅度、频率的三角函数之和。
- 在数字图像分析中非常重要：
  - 图像增强、图像分析、图像恢复、图像压缩。
  - 对于一副图像，它的DCT变换仅有少量的系数不为0，特别适合压缩
  - DCT是JPEG压缩标准的核心部分

4.4.5 离散余弦变换的表达式

离散余弦变换

- 离散余弦变换：

$$F(u,v) = \alpha_u \alpha_v \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} I(x,y) \cos\frac{\pi(2x+1)u}{2X} \cos\frac{\pi(2y+1)v}{2Y}$$

- 其中

$$\alpha_u = \begin{cases} 1/\sqrt{X} & u = 0 \\ \sqrt{2}/\sqrt{X} & 1 \le u \le X-1 \end{cases}$$

$$\alpha_v = \begin{cases} 1/\sqrt{Y} & u = 0 \\ \sqrt{2}/\sqrt{Y} & 1 \le v \le Y-1 \end{cases}$$

4.4.6 离散余弦变换逆变换的表达式

离散余弦变换的逆变换

- 离散余弦变换的逆变换：

$$I(x,y)=\sum_{u=0}^{X-1}\sum_{v=0}^{Y-1}\alpha_u\alpha_v F(u,v)\cos\frac{\pi(2x+1)u}{2X}\cos\frac{\pi(2y+1)v}{2Y}$$

- 其中

$$\alpha_u=\begin{cases}1/\sqrt{X} & u=0\\ \sqrt{2}/\sqrt{X} & 1\le u\le X-1\end{cases}$$

$$\alpha_v=\begin{cases}1/\sqrt{Y} & u=0\\ \sqrt{2}/\sqrt{Y} & 1\le v\le Y-1\end{cases}$$

4.4.7 离散余弦变换的系数

- 对于一副图像，它的DCT变换仅有少量的系数不为0，特别适合压缩

4.4.7 离散余弦变换的系数

$$F(u,v)=\alpha_u\alpha_v\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}I(x,y)\cos\frac{\pi(2x+1)u}{2X}\cos\frac{\pi(2y+1)v}{2Y}$$

$$(u=0,v=0)$$

$$\alpha_u=\sqrt{\frac{1}{8}},\alpha_v=\sqrt{\frac{1}{8}}$$

$$\cos\frac{\pi(2x+1)u}{2X}\cos\frac{\pi(2y+1)v}{2Y}=1$$

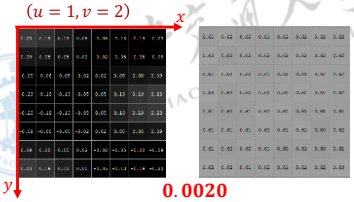
4.92

4.92

4.4.7 离散余弦变换的系数

$$F(u,v)=\alpha_u\alpha_v\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}I(x,y)\cos\frac{\pi(2x+1)u}{2X}\cos\frac{\pi(2y+1)v}{2Y}$$

$$(u=1,v=2)$$



0.0020

4.4.7 离散余弦变换的系数

- 对于一副图像，它的DCT变换仅有少量的系数不为0，特别适合压缩

4.92	-0.01	0.00	0.00	0.01	0.00	0.01	0.00
0.01	0.00	0.00	0.00	-0.01	0.00	-0.01	0.02
0.01	0.00	0.01	-0.01	-0.01	-0.01	0.00	0.00
-0.02	0.00	-0.01	0.00	0.00	0.00	0.00	0.00
0.01	0.00	0.00	0.00	0.00	0.00	-0.01	0.00
-0.01	0.00	0.00	-0.01	-0.01	0.00	0.00	0.01
0.01	0.00	0.00	-0.01	-0.01	-0.02	-0.01	0.00
-0.01	0.00	0.00	-0.01	0.00	0.00	-0.01	0.01

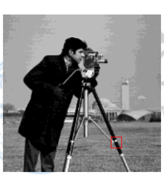
4.4.7 离散余弦变换的系数

- 对于一副图像，它的DCT变换仅有少量的系数不为0，特别适合压缩

3.35	0.00	-0.03	-0.06	0.02	-0.01	0.00	-0.02
0.03	0.00	-0.02	0.02	0.03	0.03	-0.02	-0.02
0.23	-0.03	-0.01	0.01	-0.03	0.01	0.00	0.00
0.03	-0.01	0.07	0.00	-0.01	-0.02	0.01	0.03
-0.08	-0.01	0.00	0.03	0.01	0.00	0.01	0.01
-0.02	0.03	0.02	0.02	-0.03	-0.02	0.01	0.02
0.06	0.02	0.01	0.00	0.00	-0.02	0.00	0.02

4.4.7 离散余弦变换的系数

- 对于一副图像，它的DCT变换仅有少量的系数不为0，特别适合压缩

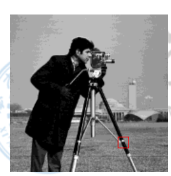


3.35	0.00	-0.03	-0.06	0.02	-0.01	0.00	-0.02
0.03	0.00	-0.02	0.02	0.03	0.03	-0.02	-0.02
0.23	-0.03	-0.01	0.01	-0.03	0.01	0.00	0.00
0.03	-0.01	0.07	0.00	-0.01	-0.02	0.01	0.03
-0.08	-0.01	0.00	0.03	0.01	0.00	0.01	0.01
-0.02	0.03	0.02	0.02	-0.03	-0.02	0.01	0.02
0.06	0.02	0.01	0.00	0.00	-0.02	0.00	0.02

1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

4.4.7 离散余弦变换的系数

- 对于一副图像，它的DCT变换仅有少量的系数不为0，特别适合压缩



0.00	-0.02
0.02	-0.02
0.00	0.00
0.01	0.03
0.01	0.01
0.01	0.02
0.00	0.02

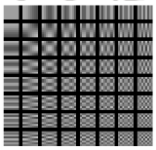
						0	0
						0	0
						0	0
						0	0
						0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

4.4.8 离散余弦变换的解释

离散余弦变换

- 离散余弦变换的矩阵解释：

$$F(u,v)=\sum_{x=1}^{X}\sum_{y=1}^{Y}I(x,y)\left[\alpha_u\alpha_v\cos\frac{\pi(2x-1)u}{2X}\cos\frac{\pi(2y-1)v}{2Y}\right]$$



4.4.9 离散余弦变换的特点

- 本质上是正交变换
- 给定(u,v)，$\alpha_u\alpha_v\cos\frac{\pi(2x-1)u}{2X}\cos\frac{\pi(2y-1)v}{2Y}$，是一组正交基：

$$0=\sum_{x=1}^{X}\sum_{y=1}^{Y}\left[\alpha_u\alpha_v\cos\frac{\pi(2x-1)u'}{2X}\cos\frac{\pi(2y-1)v'}{2Y}\right]\left[\alpha_u\alpha_v\cos\frac{\pi(2x-1)u}{2X}\cos\frac{\pi(2y-1)v}{2Y}\right]$$

$$1=\sum_{x=1}^{X}\sum_{y=1}^{Y}\left[\alpha_u\alpha_v\cos\frac{\pi(2x-1)u}{2X}\cos\frac{\pi(2y-1)v}{2Y}\right]\left[\alpha_u\alpha_v\cos\frac{\pi(2x-1)u}{2X}\cos\frac{\pi(2y-1)v}{2Y}\right]$$

- 低频分量有值，高频分量一般较少的值

4.4.9 离散余弦变换的特点

- 计算的可分离

$$F(u,v)=\sum_{x=1}^{X}\sum_{y=1}^{Y}I(x,y)\left[\alpha_u\alpha_v\cos\frac{\pi(2x-1)u}{2X}\cos\frac{\pi(2y-1)v}{2Y}\right]$$

- 可以转化为先对y进行离散余弦变换，然后再对x进行离散预先变换。

$$F(u,v)$$
$$=\sum_{x=1}^{X}\sum_{y=1}^{Y}I(x,y)\left[\alpha_u\alpha_v\cos\frac{\pi(2x-1)u}{2X}\cos\frac{\pi(2y-1)v}{2Y}\right]$$
$$=\sum_{x=1}^{X}\alpha_u\cos\frac{\pi(2x-1)u}{2X}\sum_{y=1}^{Y}\left[\alpha_v I(x,y)\cos\frac{\pi(2y-1)v}{2Y}\right]$$