

数字图像与视频处理

卢官明 唐贵进 崔子冠 编著

机械工业出版社
CHINA MACHINE PRESS



“十三五”江苏省高等学校重点教材 (编号: 2017-2-029)
高等院校通信与信息专业规划教材

数字图像 与视频处理

DIGITAL IMAGE AND VIDEO PROCESSING



卢官明 唐贵进 崔子冠 编著

机械工业出版社
CHINA MACHINE PRESS

教学资源下载网站
<http://www.cmpedu.com>



第2章 图像增强

熊健
南邮通信学院



本章学习目标

- 掌握数字图像增强的基本方法和技术。
- 掌握数字图像灰度的线性与非线性变换的方法及应用。
- 熟悉直方图均衡化、直方图规定化的步骤。
- 掌握图像平滑的基本方法，如邻域平均法、中值滤波法、低通滤波。
- 了解基于非局部相似性的图像去噪、基于稀疏表示的图像去噪算法。
- 掌握图像锐化的基本方法，如梯度运算、索贝尔（**Sobel**）算子、拉普拉斯算子、高通滤波。
- 了解图像的同态滤波、基于**Retinex**理论的图像增强方法。
- 了解伪彩色增强、假彩色增强的基本方法。



第2章 图像增强

- 2.1 引言
- 2.2 图像的灰度变换
- 2.3 图像平滑与去噪
- 2.4 图像锐化
- 2.5 图像的同态滤波
- 2.6 基于Retinex理论的图像增强
- 2.7 彩色增强
- 2.8 MATLAB编程实例

2.1 引言

在图像的形成、存储、传输等过程中，由于多种因素的影响，会导致图像质量的下降。



对比度下降



2.1 引言

改善降质图像（退化图像）的方法：

- **图像增强:** 不考虑图像降质的原因，并不要求改善后的图像去逼近原始图像，而是根据一定的要求将图像中感兴趣的部分加以处理或突出有用的图像特征（如边缘、轮廓、对比度等），抑制不需要的信息，以改善图像的主观视觉效果或便于后续的图像分析和识别。
- **图像复原:** 针对图像降质的具体原因，设法补偿降质因素，从而使改善后的图像尽可能地逼近原始图像。



图像增强算法分类

(1) 空间域法：是在空间域内直接对图像的像素值进行运算操作。空间域法又分为点运算处理法和邻域运算处理法。

- **点运算处理法：**是指直接对图像的各像素点逐一进行灰度变换的处理方法。例如，图像的**灰度变换**、**直方图修正**等都采用点运算处理法。
- **邻域运算处理法：**是对图像像素的某一邻域进行处理的方法。例如，**图像平滑**、**图像锐化**等都采用邻域运算处理法。

(2) 频率域法：在频率域上对图像的变换系数进行处理，增强感兴趣的频率分量，然后再进行反变换到空间域，得到增强后的图像。常用的方法包括**低通滤波**、**高通滤波**以及**同态滤波**等。



图像增强算法分类

图像增强

空域法 (图像域)

点运算处理

灰度变换

直方图修正

邻域运算处理

图像平滑

图像锐化

彩色增强

伪彩色增强

假彩色增强

真彩色增强

频域法

(变换域)

低通滤波

高通滤波

同态滤波



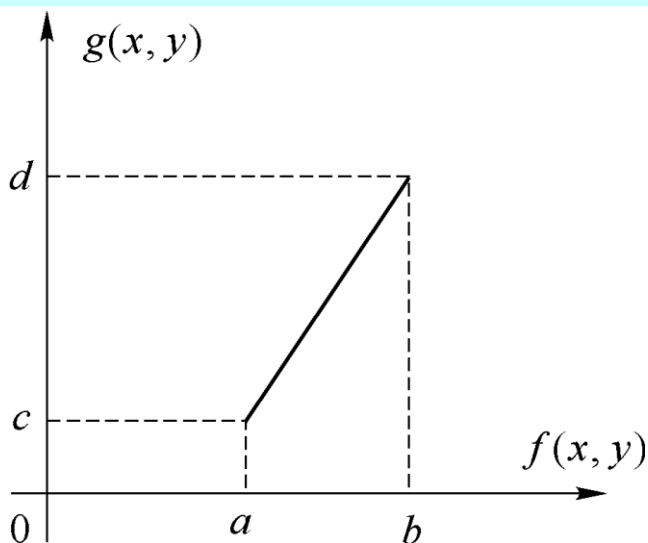
第2章 图像增强

- 2.1 引言
- **2.2 图像的灰度变换**
- 2.3 图像平滑与去噪
- 2.4 图像锐化
- 2.5 图像的同态滤波
- 2.6 彩色增强
- 2.7 MATLAB编程实例

2.2.1 灰度的线性变换

假定原图像 $f(x,y)$ 的灰度范围为 $[a,b]$ ，变换后图像 $g(x,y)$ 的灰度范围扩为 $[c,d]$ ，则采用线性变换：

$$g(x, y) = \frac{d - c}{b - a} [f(x, y) - a] + c$$



(a)原图像



(b)变换后的图像

图2-2 灰度的线性变换



2.2.1 灰度的线性变换

若图像灰度在 $[0, M_f]$ 范围内，其中大部分像素的灰度级分布在区间 $[a, b]$ ，很小部分的灰度级超出了此区间，为改善增强的效果，可令

$$g(x, y) = \begin{cases} c & 0 \leq f(x, y) \leq a \\ \frac{d - c}{b - a} [f(x, y) - a] + c & a \leq f(x, y) \leq b \\ d & b \leq f(x, y) \leq M_f \end{cases}$$

2.2.1 灰度的线性变换

为了突出感兴趣的目标或灰度区间，相对抑制那些不感兴趣的灰度区域，可采用分段线性变换，如常用的三段线性变换法。

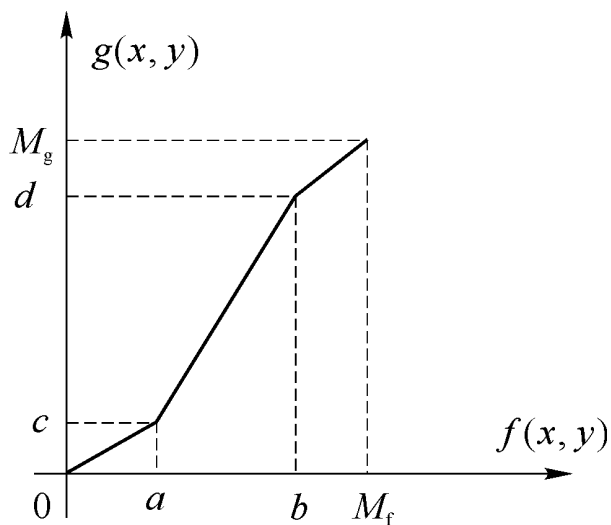


图2-4 分段线性变换

$$g(x, y) = \begin{cases} \frac{c}{a} f(x, y) & 0 \leq f(x, y) < a \\ \frac{d-c}{b-a} [f(x, y) - a] + c & a \leq f(x, y) < b \\ \frac{M_g - d}{M_f - b} [f(x, y) - b] + d & b \leq f(x, y) < M_f \end{cases}$$



2.2.2 灰度的非线性变换

- 采用非线性变换函数（例如对数函数、幂指数函数等）

- 对数变换式
$$g(x, y) = a + \frac{\ln[f(x, y) + 1]}{b \cdot \ln c}$$

a 、 b 、 c 是调整曲线的位置和形状的参数。

- 指数变换式
$$g(x, y) = b^{c[f(x, y) - a]} - 1$$

a 、 b 、 c 是调整曲线的位置和形状的参数。



2.2.3 直方图修正

1.直方图的概念

如果将图像中像素亮度（灰度级）看成是一个随机变量，则其分布情况就反映了图像的**统计特性**。灰度直方图是灰度级的函数，它表示图像中具有某种灰度级的像素的个数，反映了图像中每种灰度级出现的概率，如图2-10所示。

2.直方图的概念

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 6 | 4 | 3 | 1 | 2 | 1 |
| 1 | 6 | 6 | 4 | 1 | 6 |
| 3 | 4 | 5 | 6 | 6 | 6 |
| 1 | 4 | 6 | 6 | 2 | 4 |
| 1 | 3 | 6 | 4 | 6 | 6 |

图像的灰度级表示

| | | | | | | |
|------|---|---|---|---|---|----|
| 灰度级 | 1 | 2 | 3 | 4 | 5 | 6 |
| 出现频率 | 7 | 4 | 3 | 7 | 2 | 13 |

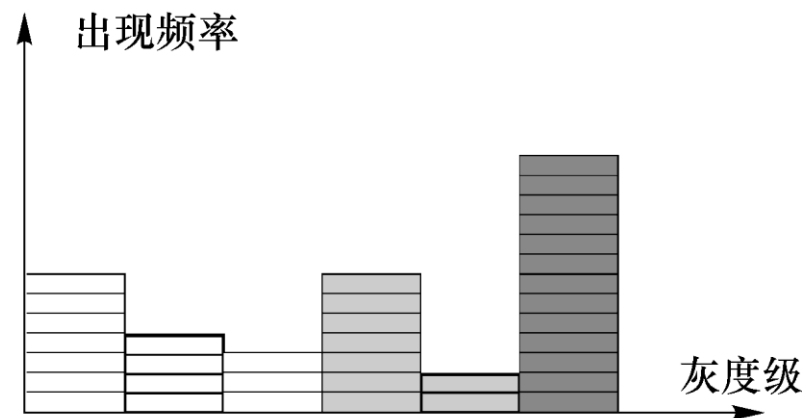


图2-10 图像的灰度直方图



2.灰度直方图的定义

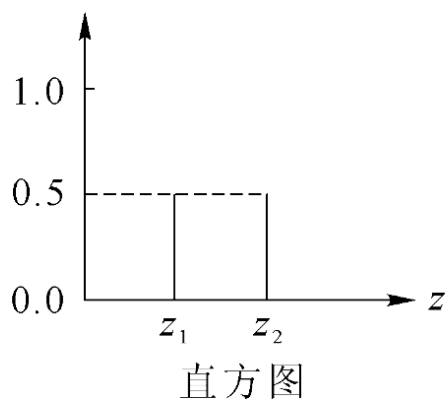
设图像总像素个数为 n ，共有 L 级灰度， r_k 为图像的第 k 级灰度值，并且具有灰度级 r_k 的像素数为 n_k ，则：

$$p(r_k) = \frac{n_k}{n} \quad (k = 0, 1, 2, \dots, L-1)$$



3.直方图的性质

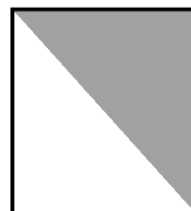
- (1) 直方图是一幅图像中各像素灰度值出现的频数的统计结果，它只反映该图像中不同灰度值出现的次数，而未反映某一灰度值像素所在的位置。**
- (2) 任一幅图像，都能唯一地确定出一幅与它对应的直方图，但不同的图像，可能有相同的直方图。**
- (3) 如果一幅图像由两个不连续的区域组成，并且每个区域的直方图已知，则整幅图像的直方图是这两个区域的直方图之和。**



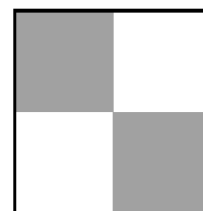
图像 a



图像 b



图像 c



图像 d

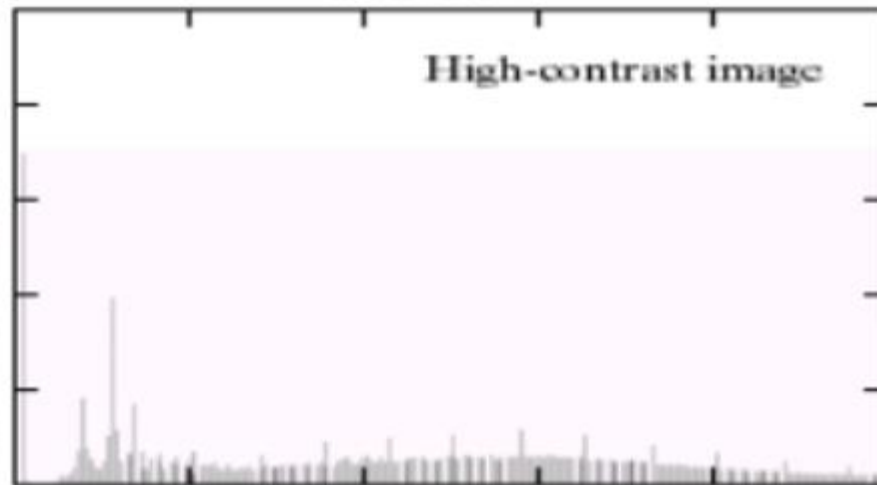
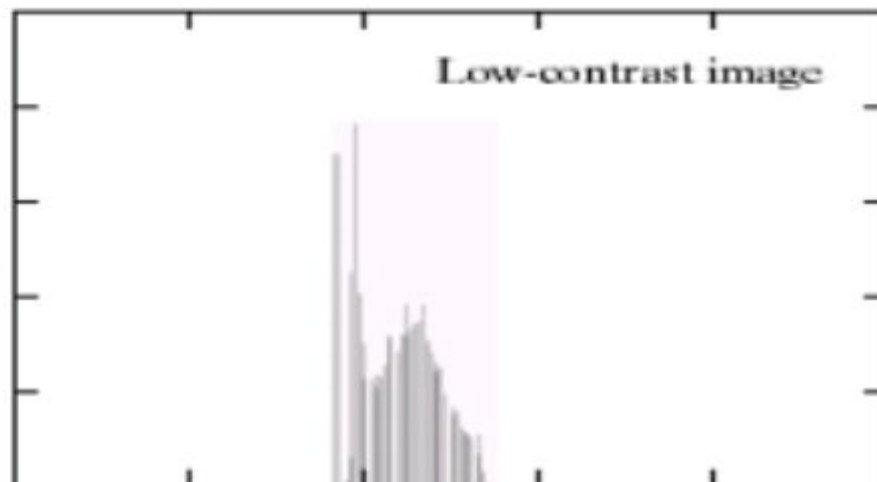
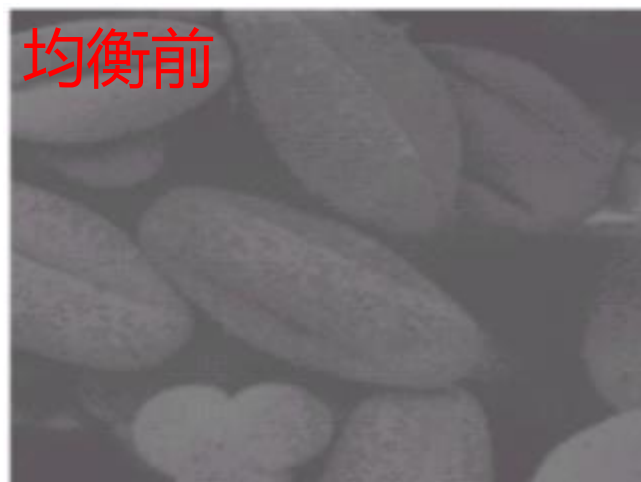
图2-11 图像与直方图间的多对一关系

4.直方图均衡化



图像整体偏暗

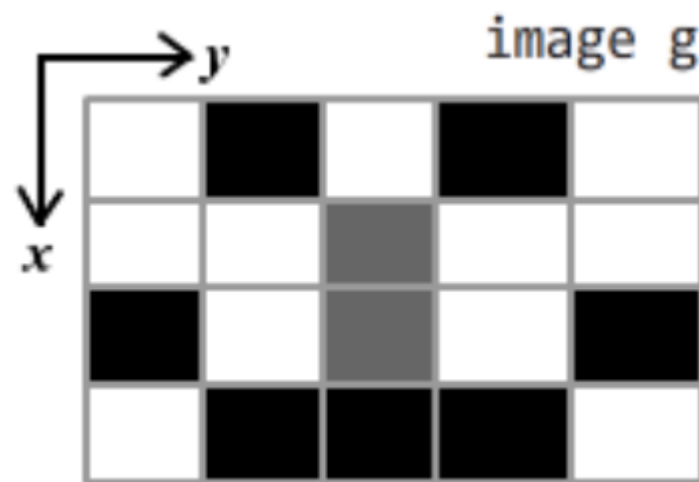
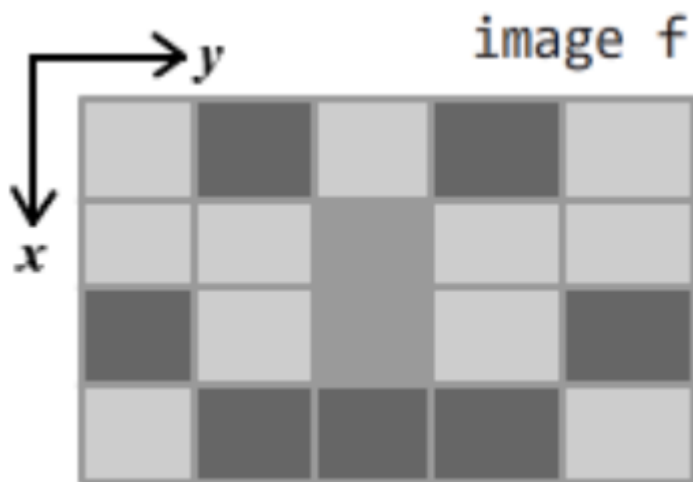
4.直方图均衡化



直方图均衡化方法

用累计分布函数(Cumulative Distribution Function, CDF) 作为灰度变换函数 $s = T(r)$, 从而将原始图像的关于灰度 r 的分布直方图, 转换为关于灰度 s 的均匀分布。

设输入图像为二元函数 $f(x, y)$, 输出图像为二元函数 $g(x, y)$





直方图均衡化方法

设任意灰度值 t 在 f 中出现的概率为函数 $p_f(t)$ ，在 g 中出现的概率为函数 $p_g(t)$ 。这两个函数均可以直接由图像统计出来。定义另两个函数（分别表示 f 和 g 中灰度值小于 n 的概率）

$$S_f(n) = \int_1^n p_f(t) dt \quad S_g(n) = \int_1^n p_g(t) dt$$

必须保证：原本比 r 暗的灰度，在变换后依然比 s 暗；原本比 r 亮的灰度，在变换后依然比 s 亮。否则图像会黑白颠倒。

$$S_f(r) = S_g[T(r)] \Leftrightarrow S_f(r) = S_g(s)$$

对上式进行微分就得到：

$$p_f(r) \cdot dr = p_g(s) \cdot ds \quad \dots\dots (2)$$



直方图均衡化方法

再接下来，如果我们令变换 $T(r) = L \cdot S_f(r)$ ，那么：

$$\begin{aligned} s &= T(r) \\ &= L \cdot S_f(r) \\ &= L \cdot \int_1^r p_f(t) dt \\ \Rightarrow \frac{ds}{dr} &= L \cdot p_f(r) \dots\dots\dots (3) \end{aligned}$$

由 (2)(3) 得

$$p_g(s) = 1/L$$

g 中各灰度出现概率相等，为常数1/L。也就是说，各灰度被完全均摊了



例2-1 给定一幅图像的灰度级概率密度函数为

$$P_r(r) = \begin{cases} -2r + 2 & 0 \leq r \leq 1 \\ 0 & \text{其他} \end{cases}$$

求其直方图的均衡化，计算出变换函数 $T(r)$ 。

解：为使其变换为一幅灰度级均匀分布的图像，即直方图均匀化处理，必须求出变换函数 $T(r)$ 。由式（2-12）得

$$s = T(r) = \int_0^r P_r(x) dx = \int_0^r (-2x + 2) dx = -r^2 + 2r$$

均衡化前后的直方图如图2-14所示。

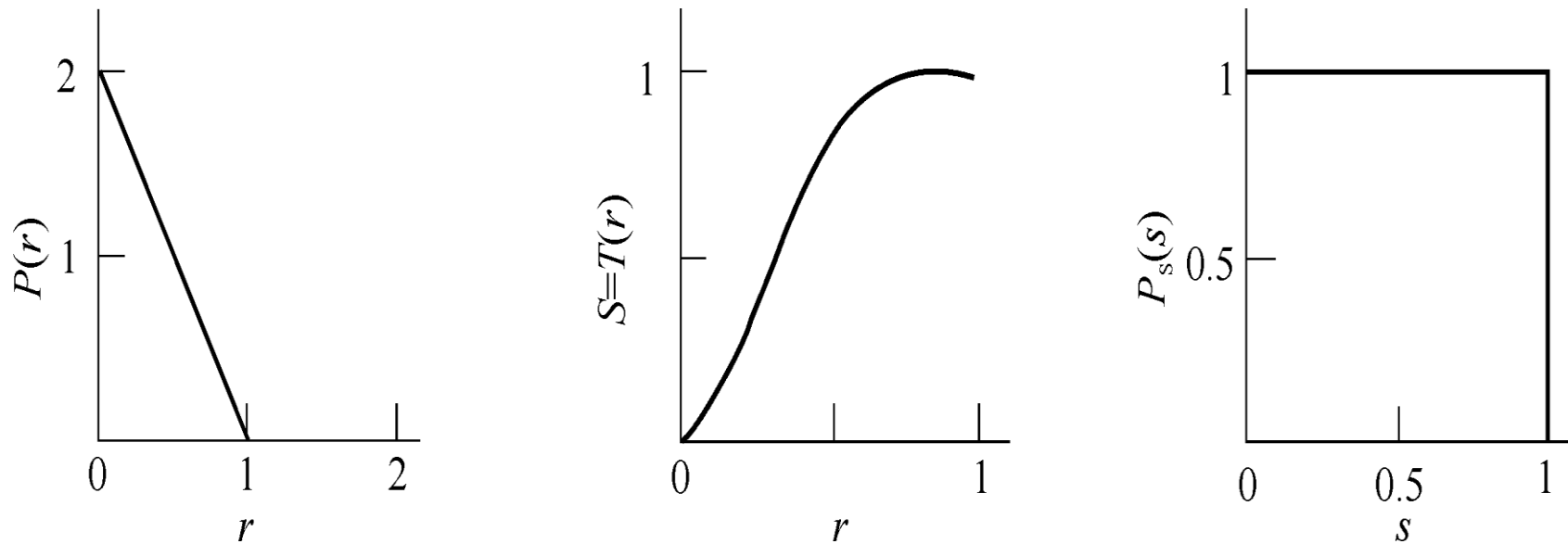
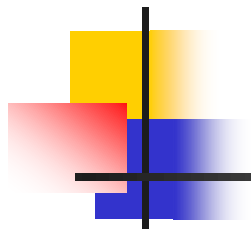
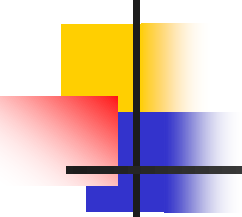


图2-14 将非均匀概率密度函数变换成均匀概率密度函数



对于离散图像，假定数字图像中的总像素为 n ，灰度级总数为 L 个，第 k 个灰度级的值为 r_k ，图像中具有灰度级 r_k 的像素数目为 n_k ，则该图像中灰度级 r_k 像素出现的概率（或称为频数）为

$$p_r(r_k) = \frac{n_k}{n} \quad (0 \leq r_k \leq 1; k = 0, 1, \dots, L-1)$$

对其进行直方图均衡化处理的变换函数为

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n}$$

相应的逆变换函数为

$$r_k = T^{-1}(s_k) \quad (0 \leq s_k \leq 1)$$



直方图均衡化的实现步骤

- **1. 统计原始图像的直方图：**

$$p_r(r_k) = \frac{n_k}{n}$$

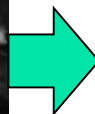
其中， r_k 是归一化的输入图像灰度级。

- **2. 计算直方图累积分布曲线**

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n}$$

- **3. 用累积分布函数作变换函数进行图像灰度变换：**根据计算得到的累积分布函数，建立输入图像与输出图像灰度级之间的对应关系，即重新定位累积分布函数 s_k （与归一化灰度等级 r_k 比较，寻找最接近的一个作为原灰度级 k 变换后的新灰度级）。

彩色图像-直方图均衡化的实现步骤



Hist eq



2.2.4 直方图规定化

- 假设 $P_r(r)$ 是原始图像灰度分布的概率密度函数， $P_z(z)$ 是希望得到的图像的灰度分布概率密度函数。将灰度直方图从 $P_r(r)$ 变换到 $P_z(z)$ 的处理，称为直方图规定化处理。如何建立 $P_r(r)$ 和 $P_z(z)$ 之间的联系是直方图规定化处理的关键。



直方图规定化的步骤

首先对原始图像进行直方图均衡化处理，则有

$$s = T(r) = \int_0^r P_r(x) dx$$

假定已经得到了所希望的图像，并且它的概率密度函数是 $P_z(z)$ ，对这幅图像也做均衡化处理，即

$$u = G(z) = \int_0^z P_z(x) dx$$

其反变换函数为 $z = G^{-1}(u)$

因为对于两幅图像（这两幅图像只是灰度分布概率密度不同）同样做了均衡化处理，所以， $P_s(s) = P_u(u) = 1$ 。



直方图规定化的步骤

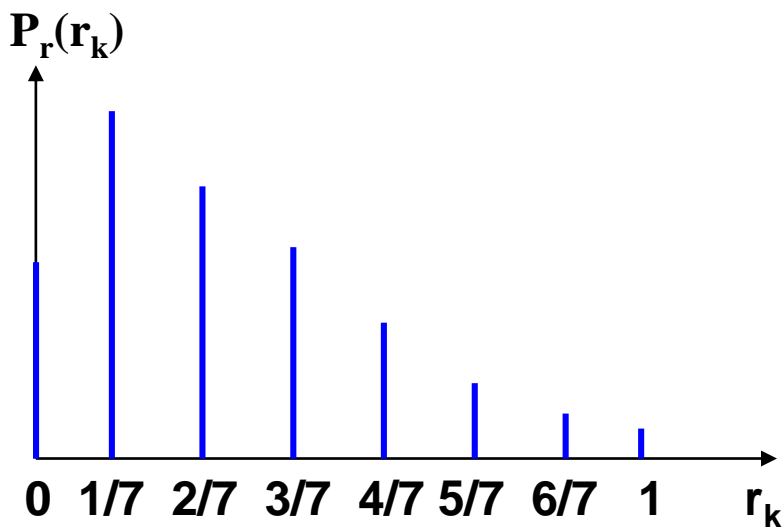
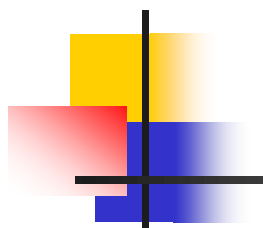
这样，如果用从原始图像中得到的均匀灰度级 s 代替式(2-18)中的 u ，即 $z = G^{-1}(u) = G^{-1}(s)$

则得到的灰度级 z 便是所希望的规定化后的图像的灰度级。

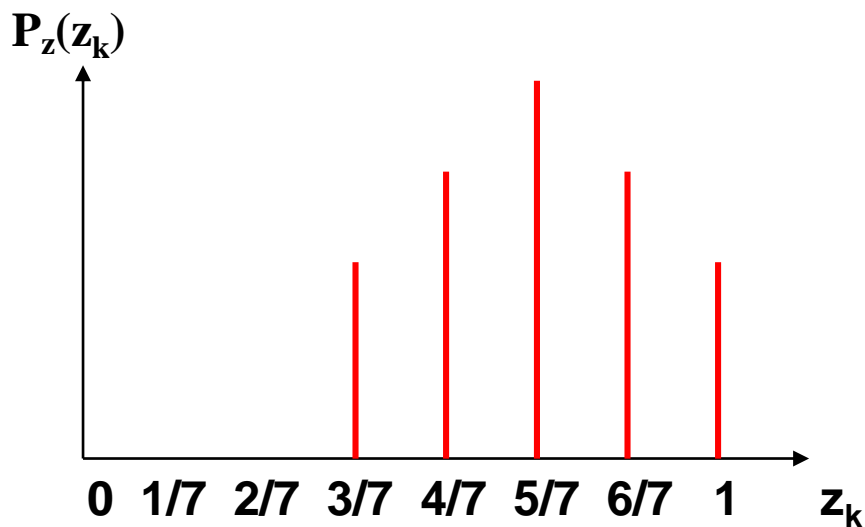
根据以上思路，可以总结出直方图规定化处理的步骤如下。

- (1) 对原图像进行直方图均衡化处理。
- (2) 规定希望的灰度概率密度函数 $P_z(z)$ ，并用式 (2-17) 求得变换函数 $G(z)$ 。
- (3) 将步骤 (1) 中所得到的灰度级 s 用到逆变换函数

$$z = G^{-1}(s) = G^{-1}[T(r)]$$



a原始直方图



b用户希望得到的直方图

我们需要获得从直方图a变换到直方图b的一个映射

首先，将直方图a均衡化，得到均匀化灰度级 s_k

然后，将直方图b也均衡化，得到均匀化灰度级 u_k

将直方图a和b均衡化后，所得到的2个均匀化的直方图对应一幅图像，则
 $s_k = u_k$ 。

直方图规定化的步骤

1. 对原始直方图 $p_r(r_k)$ 进行均衡化。

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j)$$

2. 对希望的直方图 $p_z(z_k)$ 进行均衡化。

$$u_k = G(z_k) = \sum_{j=0}^k p_z(z_j)$$

3. 用 s_k 与 u_k 理论上相等的关系，通过公式：

$$u_k = s_k = G^{-1}(z_k), \text{ 得到 } s_k \text{ 与 } z_k \text{ 的对应关系。}$$

4. 利用步骤1所得到的 s_k 与 r_k 的对应关系，和步骤3所得到的 s_k 与 z_k 的对应关系，可以得到 r_k 与 z_k 的对应关系，即得到了 z_k 所对应的像素数目。

5. 根据步骤4中所得到的像素数目除以总像素数目得到进行规定化直方图处理后图像的直方图。

直方图规定化的步骤

原图像



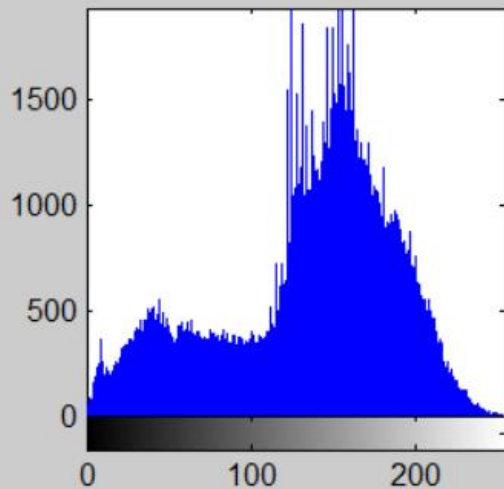
模板图像



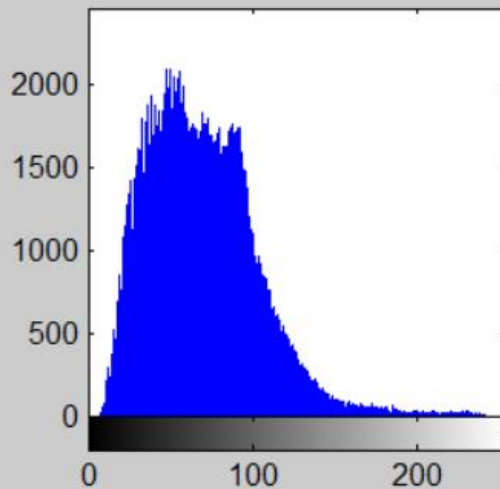
规定化后的图像



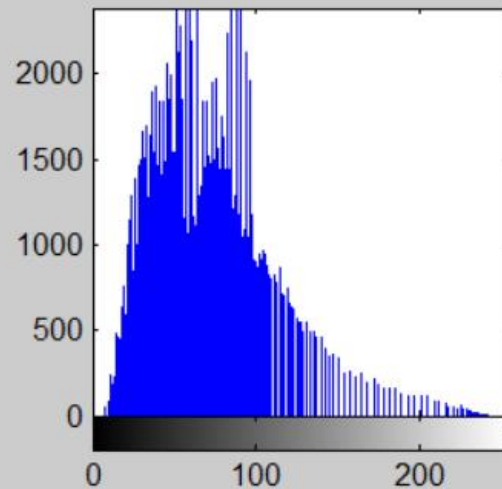
原图像的直方图



模板图像的直方图



规定化后的图像的直方图





第2章 图像增强

- 2.1 引言
- 2.2 图像的灰度变换
- **2.3 图像平滑与去噪**
- 2.4 图像锐化
- 2.5 图像的同态滤波
- 2.6 基于Retinex理论的图像增强
- 2.7 彩色增强
- 2.8 MATLAB编程实例



2.3 图像平滑与去噪

2.3.1 模板操作和卷积运算

模板操作实现了一种邻域运算，即某个像素点的运算结果不仅与本像素灰度有关，而且与其邻域点的值有关。模板操作的数学含义是卷积（或互相关）运算。

常用的模板有：

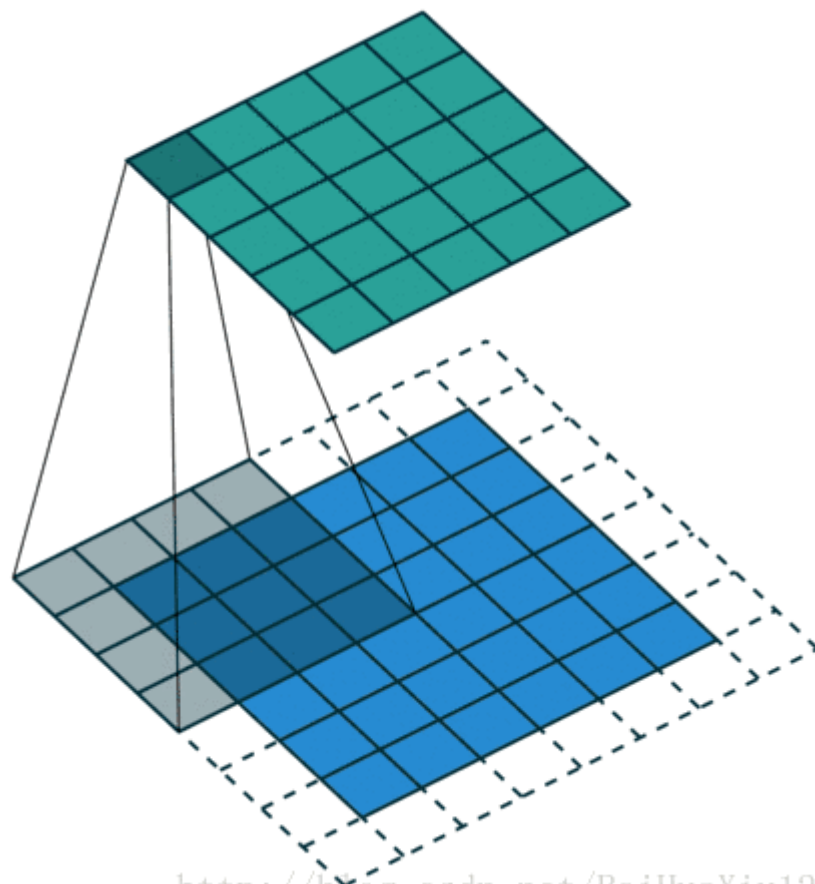
$$H_1 = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$H_2 = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$H_3 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

2.3 图像平滑与去噪

2.3.1 模板操作和卷积运算



2.3 图像平滑与

2.3.1 卷积运算 (sobel)

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$



2.3.2 邻域平均法



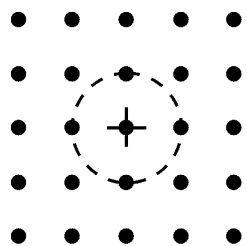
2.3.2 邻域平均法

邻域平均法是一种局部空间域处理的算法。

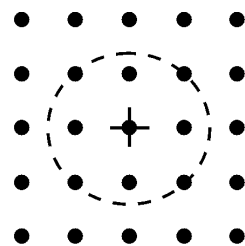
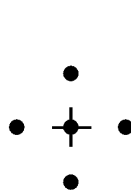
基本思想：用邻域像素灰度的平均值代替每个像素的灰度值。假定有一幅 $N \times N$ 像素的图像 $f(x,y)$ ，平滑处理后得到一幅图像 $g(x,y)$ ：

$$g(x,y) = \frac{1}{M} \sum_{(i,j) \in S} f(i,j)$$

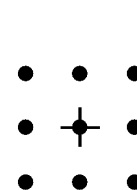
式中， $x,y=0, 1,2,\dots,N-1$ ； S 是以点 (x,y) 为中心的邻域的集合，但不包括点 (x,y) ； M 是集合内坐标点的总数。

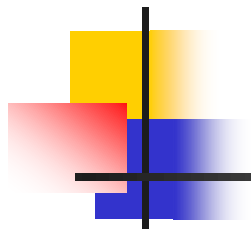


(a) 4邻域



(b) 8邻域





➤ 邻域平均法

- 优点：算法简单，计算速度快。
- 缺点：在降低噪声的同时容易模糊图像边沿和细节处。

➤ 改进：采用阈值法

$$g(x, y) = \begin{cases} \frac{1}{M} \sum_{(m,n) \in S} f(m, n) & \text{若 } \left| f(x, y) - \frac{1}{M} \sum_{(m,n) \in S} f(m, n) \right| > T \\ f(x, y) & \text{其它} \end{cases}$$

式中 T 是一个非负阈值，当一些点和它们邻值的差值小于 T 时，保留这些点的像素灰度值。



(a) 原始图像



(b) 邻域平均后的结果

图2-20 采用邻域平均法的效果



2.3.3 中值滤波

中值滤波是对一个滑动窗口内的诸像素灰度值排序，用中值代替窗口中心像素的原来灰度值，因此它是一种**非线性**的图像平滑法。

在一定的条件下，中值滤波可以克服线性滤波器所带来的图像细节模糊，而且对滤除**脉冲干扰及图像椒盐噪声非常有效**；但是，对一些**细节多，特别是点、线、尖顶细节较多的图像**则不宜采用中值滤波的方法。中值滤波的作用是在保护图像边缘的同时，去除噪声。

2.3.3 中值滤波

- **中值滤波的依据**：噪声以孤立点的形式出现，这些点对应的像素数很少，而图像则是由像素数较多、面积较大的块构成。中值滤波的目的就是要把这些孤立的点去除掉。
- **中值滤波方法**：选一个含有奇数点的窗口 W ，将这个窗口在图像上移动，把该窗口中所含的像素点按灰度值进行升（或降）序排列，取位于中间的灰度值，来代替该点的灰度值。将原图像中所有的像素点都执行上述操作后就得到中值滤波的结果图像。

例：有一个序列为 $\{0, 3, 4, 0, 7\}$ ，当窗口 $m = 5$ 时，试求出采用中值滤波的结果。

解：该序列重新排列后为 $\{0, 0, 3, 4, 7\}$ 则中值滤波的结果

$$M\{0, 0, 3, 4, 7\} = 3$$

2.3.4 基于非局部相似性去噪

- 图像的边缘、图像具有周期模式区域以及具有重复性特征的纹理



图2-24 具有非局部相似性的图像块

- 基于此，若能够有效利用具有相似结构的图像块来衡量像素与像素之间的联系，则像素点灰度值估值会更接近真实值。



2.3.4 基于非局部相似性去噪

- 因此，提出了非局部（Non-Local）滤波的概念。
- 之所以称之为非局部的方法，主要是由于这些具有相似结构的图像块都是位于图像中的不同位置。
- 可以看出，局部相似性主要考虑的是当前像素和其邻近像素的相似程度，而非局部相似性不仅考虑像素值相似度，还要兼顾像素周围的结构是否相似。



2.3.4 基于非局部相似性去噪

1、NLM (Non-Local Means)算法

$$w(i, j) = \exp\left(-\frac{d(i, j)}{h^2}\right)$$

$$d(i, j) = \|N(i) - N(j)\|_{2, \alpha}^2$$

$$\hat{f}(i) = \frac{\sum_{j \in \Phi} w(i, j) f(j)}{\sum_{j \in \Phi} w(i, j)}$$

- $\hat{f}(i)$ 为滤波结果， $d(i, j)$ 为高斯加权距离， $w(i, j)$ 为权重。



2.3.4 基于非局部相似性去噪

2、BM3D (Block-Matching and 3D filtering)算法

BM3D是一种基于块处理的去噪方法，它将非局部方法和变换域滤波有效地结合起来。分为两个步骤：

- 步骤1：生成含噪图像的基础估计（即初步去噪结果）
- 步骤2：对步骤1生成的基础估计再次去噪，形成最终估计。

2.3.4 基于非局部相似性去噪

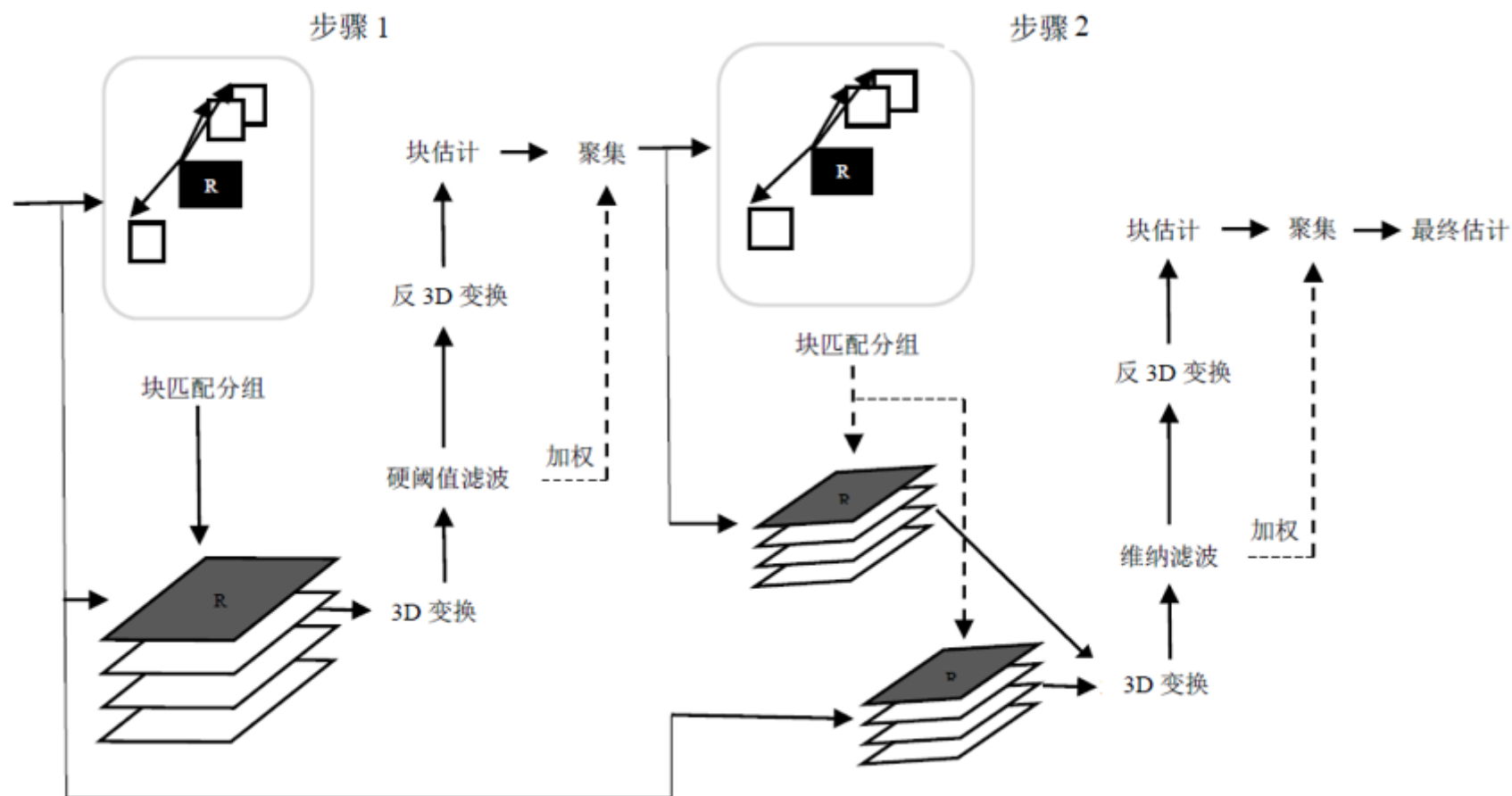


图2-25 BM3D算法流程示意图

2.3.4 基于非局部相似性去噪

3、WNNM(Weighted Nuclear Norm Minimization)算法

- 利用非局部特性，对图像进行分块匹配后得到相似块矩阵（由要降噪的图像块和图像中与其相似的图像块组成），然后将该相似块矩阵分解为干净数据矩阵和噪声数据矩阵之和，其中干净数据矩阵因为图像数据之间的相似性具有低秩特性，而噪声数据因其自身离散随机的特点而呈现稀疏性。
- 由此，通过求解最小化矩阵的秩这个优化问题来复原原始图像数据。

$$\mathbf{X}_i = \operatorname{argmin}_{\mathbf{X}_i} \|\mathbf{Y}_i - \mathbf{X}_i\|_F^2 + \lambda \|\mathbf{X}_i\|_*$$

F范数: $\|\mathbf{X}\|_F \equiv \sqrt{\sum_{i=1}^m \sum_{j=1}^n |x_{ij}|^2}$

核范数: $\|\mathbf{X}\|_* = \operatorname{tr}(\sqrt{\mathbf{X}^T \mathbf{X}}) = \operatorname{tr}(\boldsymbol{\Sigma})$

2.3.4 基于非局部相似性去噪



(a) 原图



(b) 含噪图像



(c) NLM



(d) BM3D



(e) WNNM

图2-26 噪声标准差为50时，不同非局部去噪算法效果



2.3.5 频率域低通滤波

- 图像的平滑除了在空间域中进行外，也可以在频率域中进行。
- 卷积理论是频率域技术的基础。
- 设函数 $f(x, y)$ 与算子 $h(x, y)$ 的卷积结果是 $g(x, y)$ ，即 $g(x, y) = h(x, y) * f(x, y)$ ，那么根据卷积定理，在频率域有：

$$G(u, v) = H(u, v)F(u, v)$$

- 其中 $G(u, v)$, $H(u, v)$, $F(u, v)$ 分别是 $g(x, y)$, $h(x, y)$, $f(x, y)$ 的傅立叶(或其它)变换
- $H(u, v)$ 是传递函数

2.3.5 频率域低通滤波

- 由于**噪声主要集中在高频部分**，为去除噪声改善图像质量，可采用低通滤波器 $H(u,v)$ ，来抑制 $F(u,v)$ 的高频分量，滤波得到 $G(u,v)$ ，然后再进行傅立叶反变换获得滤波图像，就可达到平滑图像的目的。

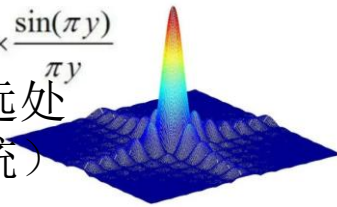
$$G(u, v) = H(u, v)F(u, v)$$

$$g(x, y) = \mathcal{T}^{-1}[H(u, v)F(u, v)]$$



$$\sin c(x, y) = \frac{\sin(\pi x)}{\pi x} \times \frac{\sin(\pi y)}{\pi y}$$

延伸到无穷远处
(非因果系统)

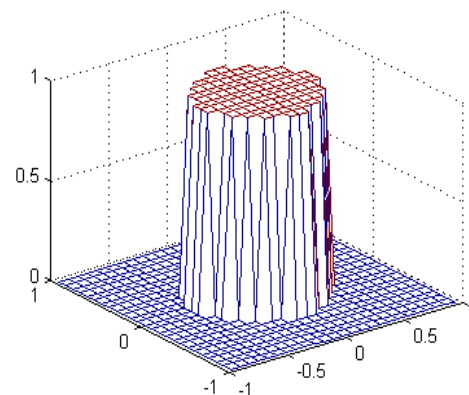


1. 理想低通滤波器

- 设傅立叶平面上理想低通滤波器离开原点的截止频率为 D_0 ，则理想低通滤波器（ILPF）的传递函数为：

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases}$$

其中 $D(u, v) = \sqrt{u^2 + v^2}$



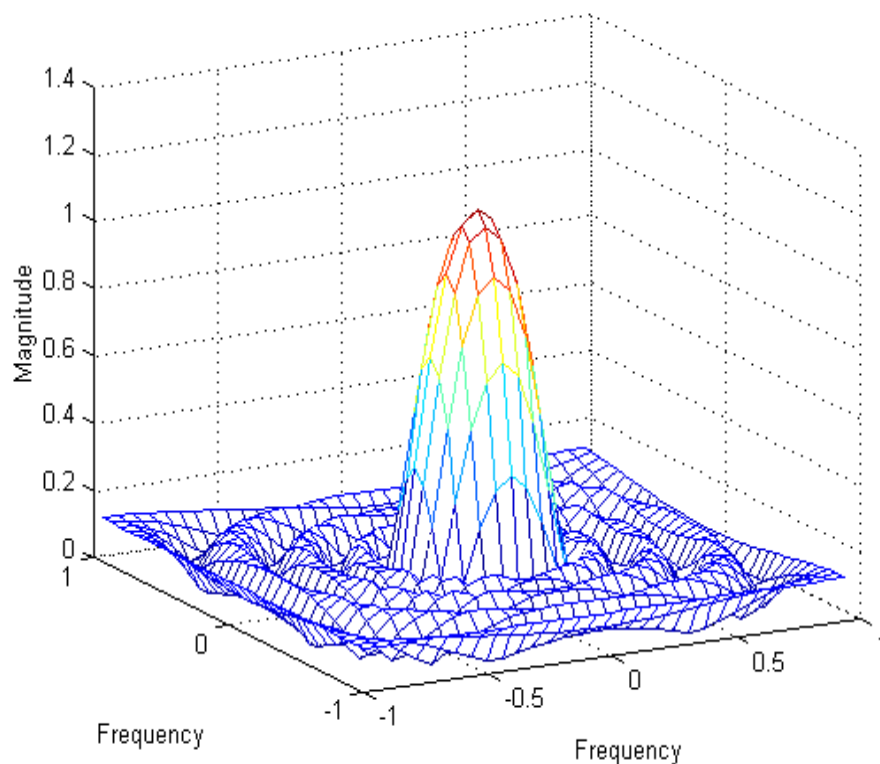
- 含义：以 D_0 为半径的圆内所有频率分量无损的通过，圆外的所有频率分量完全衰减。
- 由于高频成分包含有大量的边缘信息，因此，采用该滤波器在去噪声的同时将会导致边缘信息损失而使图像边缘模糊。

2. 巴特沃兹(Butterworth)低通滤波器

n 阶Butterworth滤波器的传递函数为:

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0} \right]^{2n}}$$

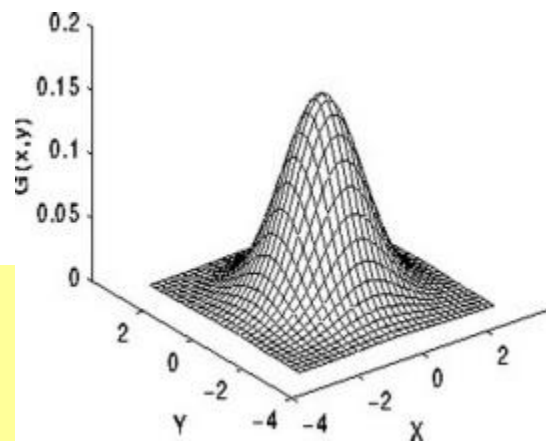
它的特性是连续性衰减，而不象理想滤波器那样陡峭变化，尾部包含大量的高频分量。因此采用该滤波器滤波在抑制噪声的同时，图像边缘的模糊程度大大减小，**没有振铃效应**产生；但计算量大于理想低通滤波器。



3. 高斯低通滤波器

Gauss滤波器的传递函数为：

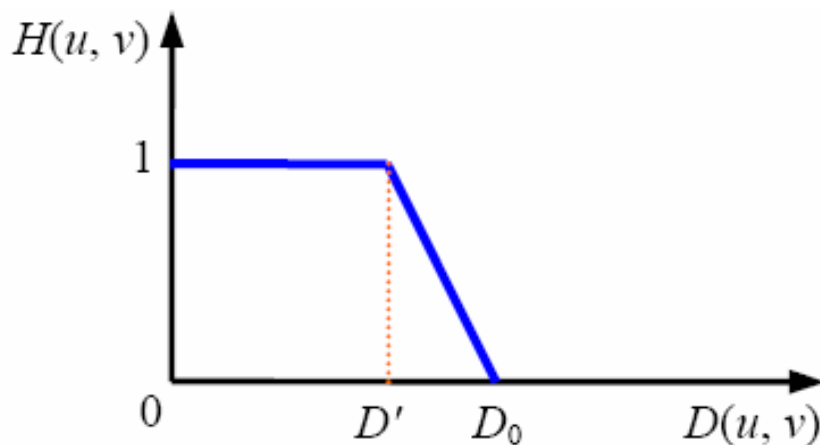
$$H(u, v) = e^{-D^2(u, v) / 2\sigma^2}$$



高斯低通滤波器的**傅立叶反变换也是高斯的**，这意味着反变换后高斯滤波器将**没有振铃现象**产生。

4. 梯形低通滤波器

$$H(u, v) = \begin{cases} 1 & \text{如 } D(u, v) \leq D' \\ \frac{D(u, v) - D_0}{D' - D_0} & \text{如 } D' < D(u, v) < D_0 \\ 0 & \text{如 } D(u, v) > D_0 \end{cases}$$



梯形低通滤波器转移函数的剖面示意图

性能介于理想低通滤波器和具有平滑过渡带的滤波器之间，滤波后的图像具有一定的模糊，也有一定的振铃现象。



2.3.6 基于稀疏表示的图像去噪

- 在对信号进行分析时，如何对信号进行有效表示是一个重要的问题。
- 传统的信号表示理论往往是基于正交变换，比如离散余弦变换。自然界的图像本身是存在自相关性的，从某种程度上来讲，大多数的自然图像都是可压缩的。
- 对数字图像而言，用二维函数 $f(i, j)$ 来表示，就存在着大量信息的冗余，可以有其他简练的表达式，比如将图像转换到其他的域中进行表示。从数学角度分析，图像的稀疏表示实质上就是对图像数据进行稀疏分解。

2.3.6 基于稀疏表示的图像去噪

- 建立一个数据库 $D = [d_1, d_2, \dots, d_M] \in R^{N \times M}$ ($M > N$)
(D 称为“字典”，其每一个元素 d_i 都是一个 N 维列向量)
不要求字典 D 中的原子（也称为“基函数”）相互正交
- 如将图像（大小为 $\sqrt{N} \times \sqrt{N}$ ）看成一维向量 $X \in R^N$ ，
可以用下式进行线性表示：

$$X = D\alpha$$

其中， $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_M]^T$ 称为图像 X 在字典 D 上的分解系数， α 中非零元素的个数称为 l_0 范数。当 l_0 范数为 k ($k \ll M$) 的时候，就称图像 X 在字典 D 下是 k -稀疏的。

2.3.6 基于稀疏表示的图像去噪

- 对图像信号最简单而且直接的稀疏测度为求图像信号的 l_0 范数，即计算信号中非零元素的个数，则稀疏表示问题可以描述为：

$$\min \| \alpha \|_0 \quad s.t. \quad X=D\alpha$$

- 上式问题的求解属于NP-hard问题（组合搜索）。松弛为 l_1 范数（即稀疏向量 中所有元素的绝对值之和）进行求解：

$$\min \| \alpha \|_1 \quad s.t. \quad X=D\alpha$$

上式为等式约束问题。



2.3.6 基于稀疏表示的图像去噪

- 当图像中存在噪声的时候，一般情况下，不需要完全准确对图像进行重构，上述优化问题可以变为以下的不等式约束（即稀疏逼近）问题：

$$\min \|\alpha\|_1 \quad s.t. \quad \|X - D\alpha\|_2^2 < \varepsilon$$

其中， ε 表示允许的误差。当 $\varepsilon = 0$ 时，即为稀疏表示问题。



2.3.6 基于稀疏表示的图像去噪

- 不含噪声干净图像一般具有一定的规律，往往可以找到某种字典，使得原子信号的结构和图像信号的结构很相似，即使得其主要分布在过完备字典中有限的原子上，相反，噪声是随机的，相互独立的，没有结构信息，分解后分布较分散，系数较小。
- 只要设置合适的逼近误差，如字典表示后舍去大量的小系数，保留少量的大系数，就可以滤除大部分噪声，实现图像的去噪。

2.3.6 基于稀疏表示的图像去噪

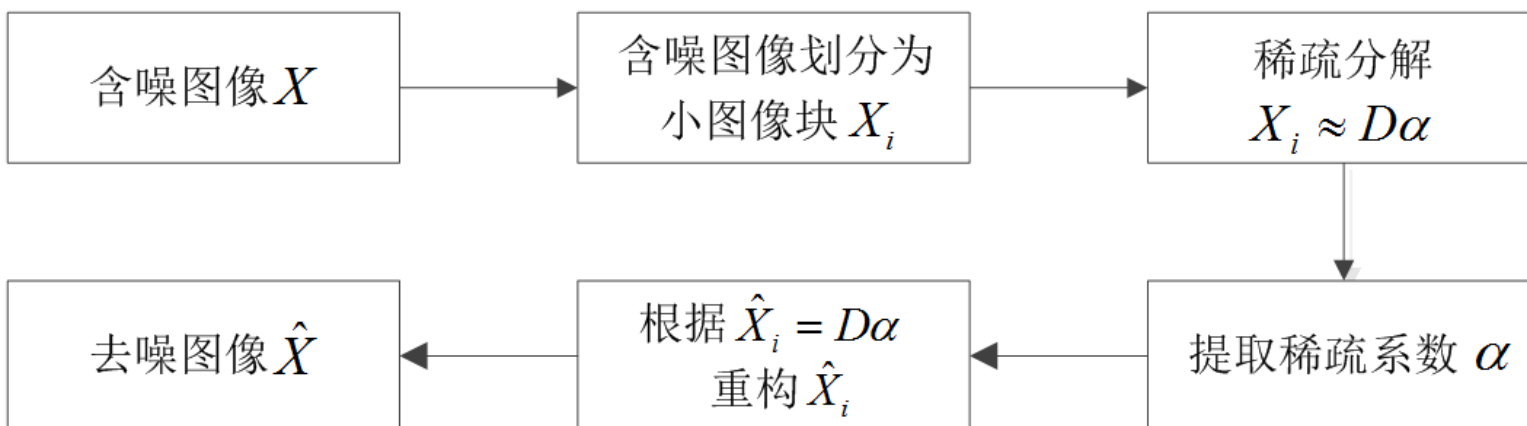


图2-31 基于稀疏表示的图像去噪的流程框图



2.3.6 基于稀疏表示的图像去噪

- 在迄今出现的稀疏表示去噪算法中，以 K -奇异值分解（ K -Singular Value Decomposition）算法最具代表性。
- **K-SVD**去噪算法主要包括两个部分：
 - 稀疏求解
 - 字典更新



第2章 图像增强

- 2.1 引言
- 2.2 图像的灰度变换
- 2.3 图像平滑与去噪
- **2.4 图像锐化**
- 2.5 图像的同态滤波
- 2.6 基于Retinex理论的图像增强
- 2.7 彩色增强
- 2.8 MATLAB编程实例



2.4.1 梯度运算（算子）

对于图像 $f(x,y)$ ，在点 (x,y) 处的梯度定义为一个矢量：

$$G[f(x,y)] = \begin{pmatrix} f'_x \\ f'_y \end{pmatrix} = \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix}$$

- **梯度**的两个重要性质是：
- **(1)** 梯度的方向在函数 $f(x,y)$ 最大变化率的方向上。
- **(2)** 梯度的幅度用 $|G[f(x,y)]|$ 表示，并由下式算出：

$$|G[f(x,y)]| = \sqrt{f'^2_x + f'^2_y} = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

2.4.1 梯度运算（算子）

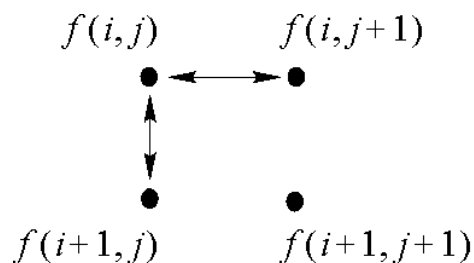
对于数字图像而言, 有两种二维离散梯度的计算方法:

➤ 水平垂直差分法

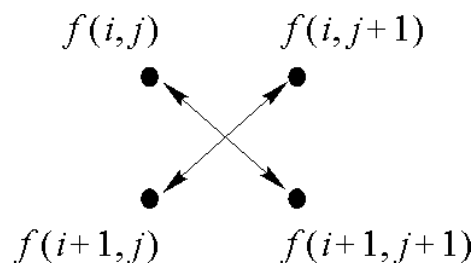
$$|G[f(i, j)]| = |f(i, j) - f(i+1, j)| + |f(i, j) - f(i, j+1)|$$

➤ 罗伯茨梯度法 (Roberts Gradient): 交叉差分法

$$|G[f(i, j)]| = |f(i, j) - f(i+1, j+1)| + |f(i+1, j) - f(i, j+1)|$$



(a)



(b)

图2-32 求梯度的两种差分算法



2.4.1 梯度运算（算子）

由梯度的计算可知：

- ① 在灰度变化平缓的区域其梯度值较小，
- ② 图像中灰度变化较大的边缘区域其梯度值大，
- ③ 而在灰度均匀区域其梯度值为零。

注意： 以上两种梯度近似算法在图像的最后一行和最后一列的各像素的梯度无法求得，一般就用前一行和前一系列的梯度值近似代替。

2.4.2 Sobel算子

- 采用梯度运算对图像进行锐化处理，同时会使噪声（平均）、条纹（隔行）等得到增强，Sobel算子则在一定程度上克服了这个问题。
- Soble算子所用的 3×3 像素窗口如图2-30所示。

$$\begin{array}{ccccc} f(i-1, j-1) & f(i-1, j) & f(i-1, j+1) & & \\ & \bullet & \bullet & \bullet & \\ f(i, j-1) & f(i, j) & f(i, j+1) & & \\ & \bullet & \bullet & \bullet & \\ & \bullet & \bullet & \bullet & \\ f(i+1, j-1) & f(i+1, j) & f(i+1, j+1) & & \end{array}$$

图2-34 Soble算子所用的 3×3 像素窗口



2.4.2 Sobel算子

- 锐化后图像 $f(i,j)$ 的灰度值为:

$$g = \sqrt{G_x^2 + G_y^2}$$

可简化为:

$$g = |G_x| + |G_y|$$

其中:

$$G_x = [f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1)] - [f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1)]$$

$$G_y = [f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1)] - [f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1)]$$



2.4.3 Laplacian算子

拉普拉斯运算也是偏导数运算的线性组合运算。

$f(x,y)$ 的拉普拉斯运算定义为：

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



二阶微分算子



2.4.3 Laplacian算子

对数字图像来讲, $f(x, y)$ 的二阶偏导数可表示为

$$\begin{aligned}\frac{\partial^2 f(x, y)}{\partial x^2} &= \nabla_x f(i+1, j) - \nabla_x f(i, j) \\ &= [f(i+1, j) - f(i, j)] - [f(i, j) - f(i-1, j)] \\ &= f(i+1, j) + f(i-1, j) - 2f(i, j)\end{aligned}$$

$$\frac{\partial^2 f(x, y)}{\partial y^2} = f(i, j+1) + f(i, j-1) - 2f(i, j)$$



2.4.3 Laplacian算子

为此，拉普拉斯算子 $\nabla^2 f$ 为

$$\begin{aligned}\nabla^2 f &= \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \\ &= f(i+1, j) + f(i-1, j) + f(i, j+1) + f(i, j-1) - 4f(i, j)\end{aligned}$$

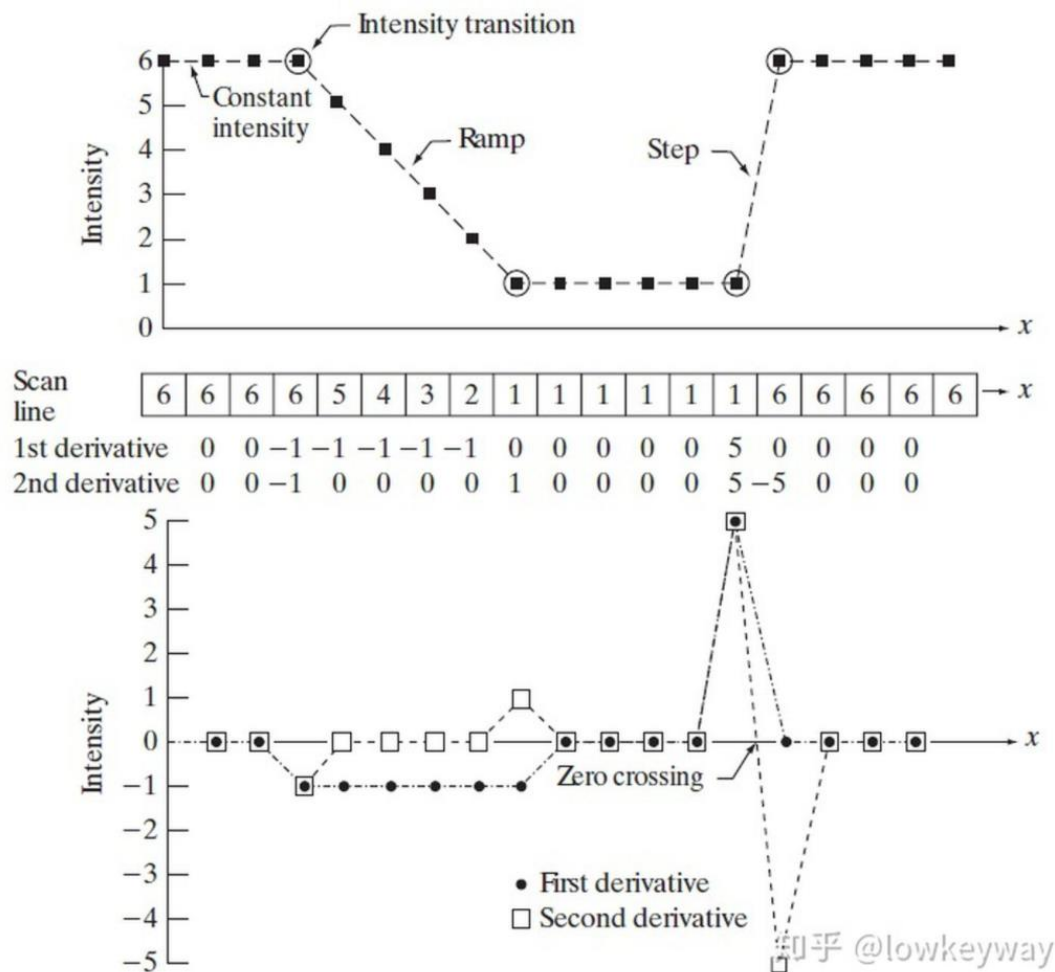
| | | |
|---|----|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

拉普拉斯算子模板

可见，数字图像在 (i, j) 点的拉普拉斯算子，可以由 (i, j) 点灰度值减去该点邻域平均灰度值来求得。

2.4.3 Laplacian算子

一阶偏导与二阶偏导的区别



一阶微分对于变化较缓的地方，一阶微分会给出一个比较长的序列，对应到图像上就是轮廓比较“粗”，二阶微分只识别跳变的边缘，对应到图像上就是比较“细”。**细节增强二阶微分更好。**

而且对像素的陡变的地方，二阶微分会出现有“零交叉”的两个点，这种点对边缘定位非常有作用



2.4.4 频率域高通滤波

- 图像中的边缘或线条等细节部分与图像频谱的高频分量相对应。

- 采用高通滤波使图像的边缘或线条等细节变得清楚，实现图像的锐化。

- 频率域高通滤波的实现，有3种常见的滤波器：

1. 理想高通滤波器

2. 巴特沃斯高通滤波器

3. 高斯高通滤波器

理想高通滤波器

$$H(u, v) = \begin{cases} 0 & D(u, v) \leq D_0 \\ 1 & D(u, v) > D_0 \end{cases}$$

巴特沃斯高通滤波器

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}}$$

高斯高通滤波器

$$H(u, v) = 1 - e^{-D^2(u, v)/2D_0^2}$$

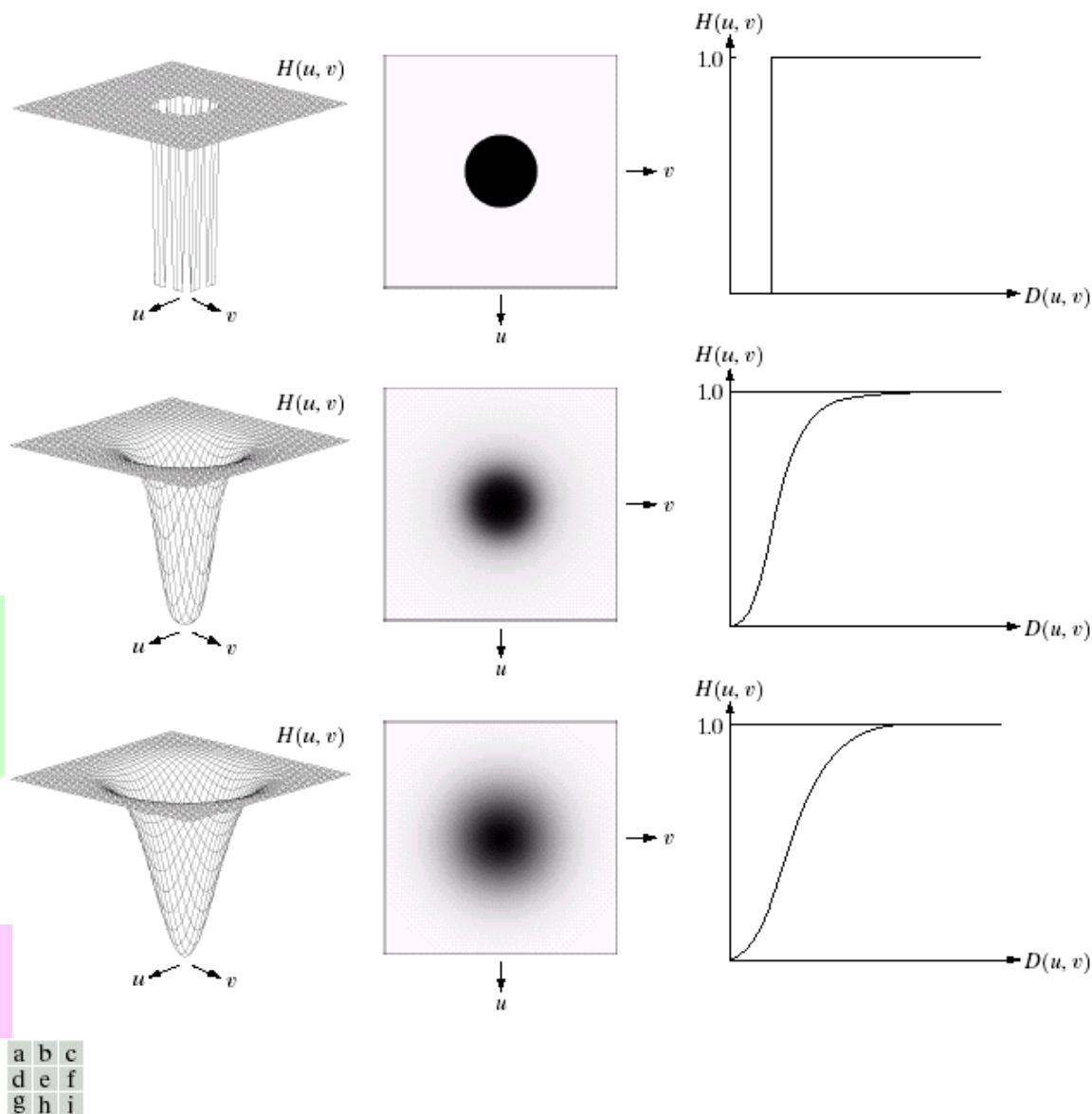


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.



第2章 图像增强

- 2.1 引言
- 2.2 图像的灰度变换
- 2.3 图像平滑与去噪
- 2.4 图像锐化
- **2.5 图像的同态滤波**
- 2.6 基于Retinex理论的图像增强
- 2.7 彩色增强
- 2.8 MATLAB编程实例



2.5 图像的同态滤波

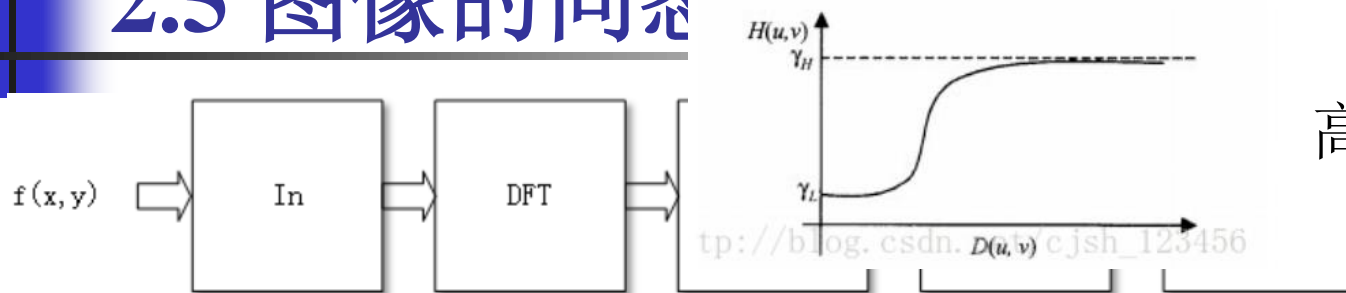
- 拍摄到的图像是光源照射到物体上后的反射特性的记录。图像可被表示为照度分量 $i(x,y)$ 和反射分量 $r(x,y)$ 的乘积。

$$f(x, y) = i(x, y)r(x, y)$$

- 由于光源**照射**的不均匀性总是**渐变**的，所以照度分量的频谱处于**低频**处；而**反射**分量的变化相对而言较为**剧烈**，因此，可粗略的看成**高频**。为使图像中景物更为清晰，应尽量**抑制照度分量**，而**增强反射分量**。
- 同态滤波是一种在频域中同时将图像亮度范围进行压缩和将图像对比度进行增强的方法。

$$H(u, v) = (\gamma_H - \gamma_L) \left[1 - e^{-c[D^2(u, v)/D_0^2]} \right] + \gamma_L$$

2.5 图像的同态



■ 成像模型（照度和反射）： $f(x, y) = i(x, y) r(x, y)$

（1）两边取对数：

$$\ln f(x, y) = \ln i(x, y) + \ln r(x, y)$$

（2）两边取傅立叶变换： $F(u, v) = I(u, v) + R(u, v)$

（3）用一频域函数 $H(u, v)$ 处理 $F(u, v)$ ：

$$H(u, v)F(u, v) = H(u, v)I(u, v) + H(u, v)R(u, v)$$

（4）反变换到空域： $h_f(x, y) = h_i(x, y) + h_r(x, y)$

（5）两边取指数：

$$g(x, y) = e^{|h_f(x, y)|} = e^{|h_i(x, y)|} \bullet e^{|h_r(x, y)|}$$

2.5 图像的同态滤波



$$H_L = 0.5; H_H = 2.0$$



窗内细节变得清晰；
窗外的灰度得到平衡

2.5 图像的同态滤波



原图



结果



第2章 图像增强

- 2.1 引言
- 2.2 图像的灰度变换
- 2.3 图像平滑与去噪
- 2.4 图像锐化
- 2.5 图像的同态滤波
- **2.6 基于Retinex理论的图像增强**
- 2.7 彩色增强
- 2.8 MATLAB编程实例



2.6 基于Retinex理论的增强

- Retinex (Retina Cortex) 理论的基本思想就是光照强度决定了原始图像中所有像素点的动态范围大小，而原始图像的固有属性则是由物体自身的反射系数决定，即假设反射图像和光照图像相乘为原始图像。



2.6 基于Retinex理论的增强

- 假设观察者得到的图像为 $I(x, y)$ ，根据上述理论，它可以表示为：

$$I(x, y) = L(x, y) \times R(x, y)$$

其中， $L(x, y)$ 表示周围光照强度信息的照度分量， $R(x, y)$ 表示物体本身固有性质的反射分量。

- 两边作对数处理：

$$\log(I(x, y)) = \log(L(x, y) \times R(x, y)) = \log(L(x, y)) + \log(R(x, y))$$



2.6 基于Retinex理论的增强

■ 令：

$$i(x, y) = \log(I(x, y)) , \quad l(x, y) = \log(L(x, y)) , \quad r(x, y) = \log(R(x, y)) ,$$

■ 有：

$$i(x, y) = l(x, y) + r(x, y)$$

2.6 基于Retinex理论的增强

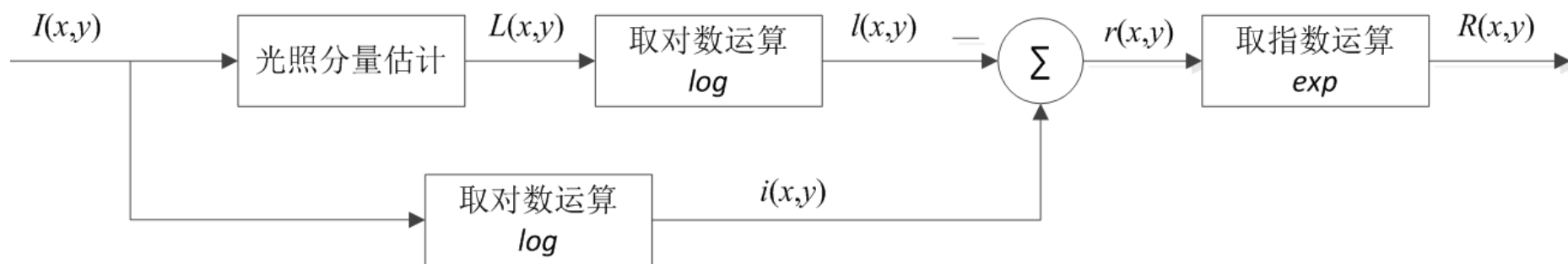


图2-45 基于Retinex理论的图像增强流程框图



2.6 基于Retinex理论的增强

- 根据对光照分量不同的估计算法， **Retinex**增强算法分为：
 - 单尺度**Retinex**（**Single Scale Retinex, SSR**）
 - 多尺度**Retinex**（**Multi-Scale Retinex, MSR**）
 - 带颜色恢复的多尺度**Retinex**（**Multi-Scale Retinex with Color Restoration, MSRCR**）等。



2.6 基于Retinex理论的增强

- **SSR**（利用高斯环绕函数对每个色彩通道卷积滤波）

$$r_i(x, y) = \log(R_i(x, y)) = \log\left(\frac{I_i(x, y)}{L_i(x, y)}\right) = \log(I_i(x, y)) - \log(I_i(x, y) * G(x, y))$$

- **MSR**（即执行不同尺度的SSR算法）

$$r_i(x, y) = \sum_{k=1}^N \omega_k (\log(I_i(x, y)) - \log(I_i(x, y) * G_k(x, y)))$$

- **MSRCR**（引入**色彩恢复因子** C 对颜色进行矫正）

$$r_i(x, y) = \sum_{k=1}^N C_i \omega_k (\log(I_i(x, y)) - \log(I_i(x, y) * G_k(x, y)))$$

2.6 基于Retinex理论的增强



(a) 原图



(b) SSR



(c) MSR



(d) MSRCR

图2-46 不同算法的增强效果



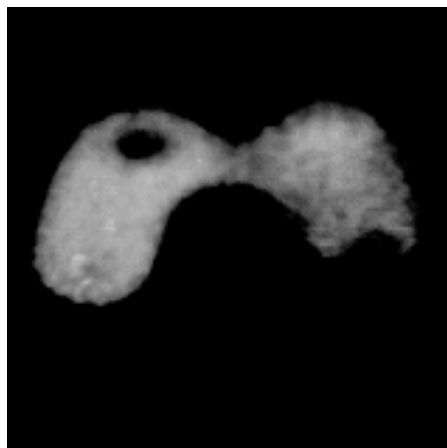
第2章 图像增强

- 2.1 引言
- 2.2 图像的灰度变换
- 2.3 图像平滑与去噪
- 2.4 图像锐化
- 2.5 图像的同态滤波
- 2.6 基于Retinex理论的图像增强
- **2.7 彩色增强**
- 2.8 MATLAB编程实例

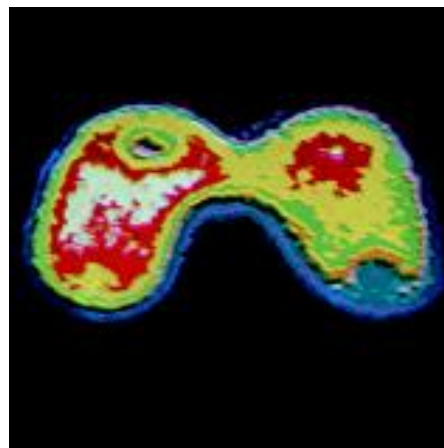
2.7 彩色增强

2.7.1 伪彩色增强

- 伪彩色（**Pseudo color**）增强是针对灰度图像提出的，其目的是把离散灰度图像的不同灰度级按照线性或者非线性关系映射成不同的颜色，得到一幅彩色图像，以改善图像的视觉效果，提高图像内容的可辨识度，使得图像的细节更加突出，目标更容易识别。



(a) 灰度图像



(b) 伪彩色图像

2.7.1 伪彩色增强

1. 灰度分层法

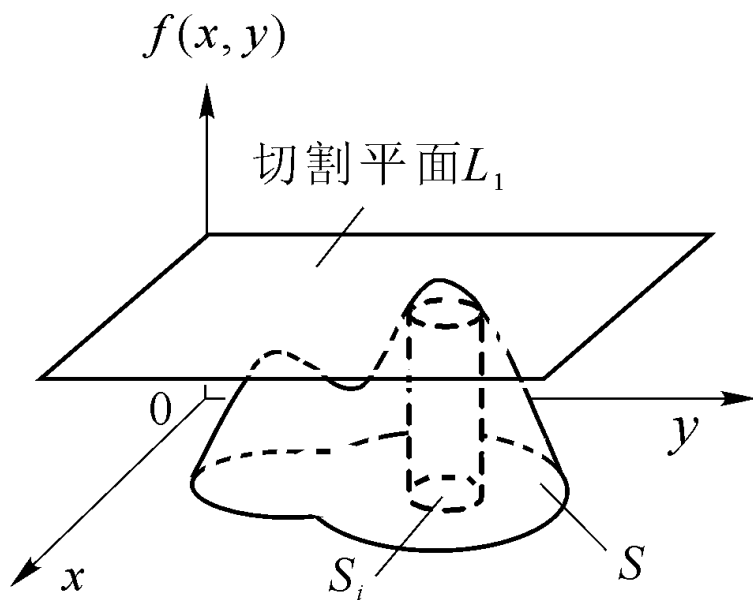


图2-47 灰度分层的切割示意图

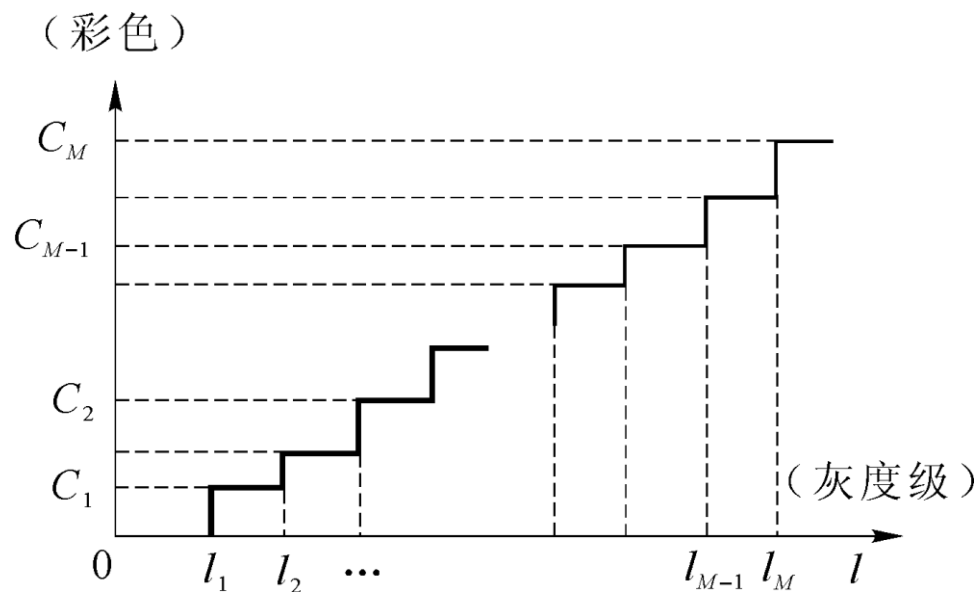


图2-48 多灰度分层的切割示意图

2.7.1 伪彩色增强

2. 灰度级彩色变换

- 将黑白图像变换为具有多种颜色渐变的连续彩色图像。
- 其方法是对输入像素的灰度级执行三个独立的变换，然后，将三个变换结果分别送入彩色电视监视器的红、绿、蓝通道，输出合成图像的彩色内容受变换函数特性调制。

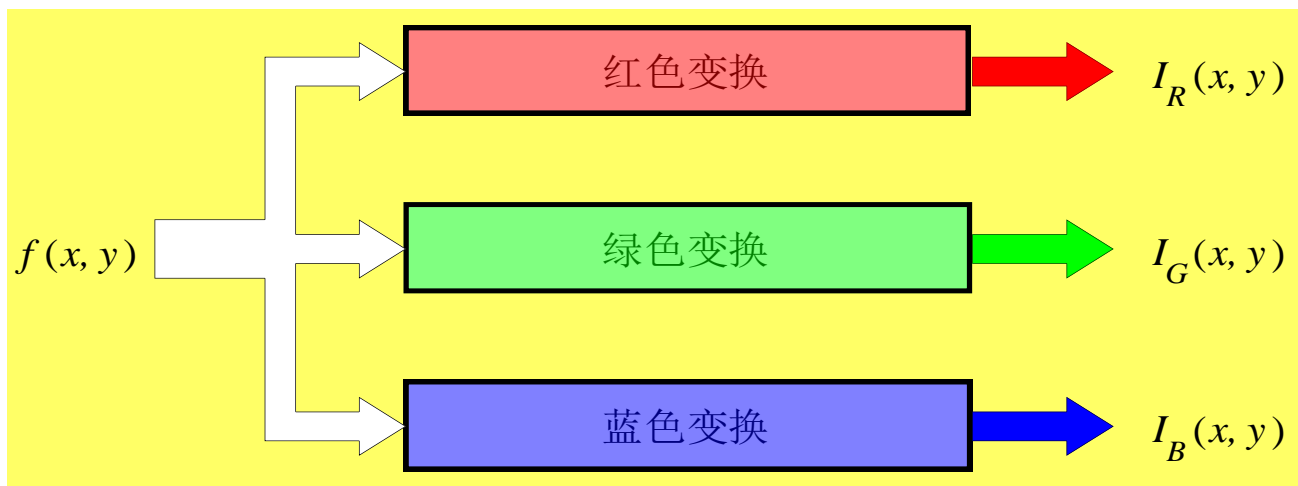


图2-49 灰度级彩色变换原理

2.7.1 伪彩色增强

3. 频率域滤波法

对原来灰度图像中的不同频率分量（可分别借助低通，带通/带阻，高通滤波器获得）赋予不同的颜色。

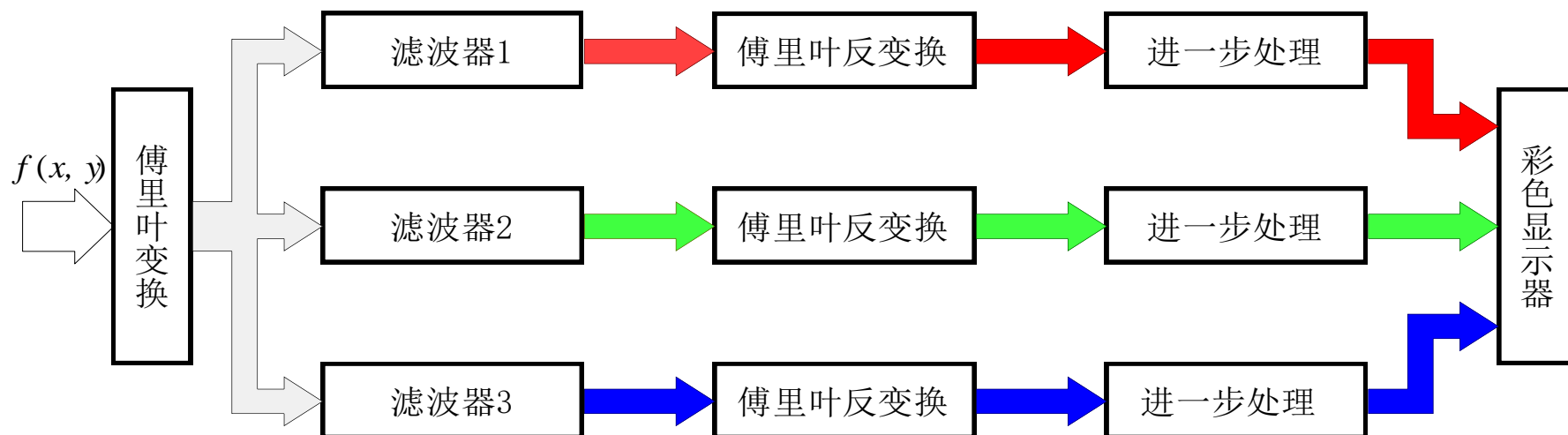


图2-50 频率域伪彩色增强处理