

图像增强

图像增强是数字图像处理的基本内容之一,其目的是突出图像中的有用信息,扩大图像中不同物体特征之间的差别,为图像的信息提取及其他图像分析技术奠定良好的基础。由于没有衡量图像增强质量的通用标准,图像增强往往和具体应用背景有较大的相关性。图像增强主要目的有两个:一是通过增强有用信息,抑制无用信息,从而改善图像的视觉效果;二是有利于人工和机器分析。图像增强包括内容广泛,如去掉图像的噪声、抽取图像某些目标的轮廓、图像的勾边处理、提取图像中的特征以及把黑白图像映射成为彩色图像等技术。图像增强可分为空域增强法和频域增强法两大类。空域增强直接对图像像素进行运算,频域增强则是在图像的某种变换域内,对图像的变换系数值进行运算,即作某种修正,然后通过逆变换获得增强了的图像。本章从空域增强和频域增强两个方面全面讨论图像增强方法。

3.1 概述

图像增强是数字图像处理的基本内容之一。一般情况下,经过增强处理后,图像的视觉效果会得到改善,某些特定信息得到增强。也就是说,图像增强处理只是突出了某些信息,增强了对某些信息的辨识能力,其他信息则被压缩了。因此,图像增强处理并不是一种无损处理,更不能增加原图像的信息,而是通过某种技术手段有选择地突出对某一具体应用有用的信息,削弱或抑制一些无用信息。例如,图像的平滑处理中经常采用低通滤波的增强方法。通过低通滤波,虽然消除了图像的噪声,但图像的空间纹理特征却被削弱了,图像从整体上显得比较模糊,换句话说,图像噪声的消除是以纹理信息的减弱为代价而实现的。

根据图像增强处理过程所在的空间不同,可分为基于空间域的增强方法和基于频率域的增强方法两类(图 3-1)。前者直接在图像所在的二维空间进行处理,即直接对每一像素的灰度值进行处理;后者则是首先经过傅里叶变换将图像从空间域变换到频率域,然后在频率域对频谱进行操作和处理,再将其反变换到空间域,从而得到增强后的图像。

基于空间域的增强方法按照所采用的技术不同可分为灰度变换和空域滤波两种方法。灰度变换是基于点操作的增强方法,它将每一个像素的灰度值按照一定的数学变换公式转换为一个新的灰度值,如增强处理中常用的对比度增强、直方图均衡化等方法。空域滤波是

基于邻域处理的增强方法,它应用某一模板对每个像素及其周围邻域的所有像素进行某种数学运算,得到该像素的新的灰度值(即输出值),输出值的大小不仅与该像素的灰度值有关,而且还与其邻域内像素的灰度值有关,常用的图像平滑与锐化技术就属于空域滤波的范畴。

图像增强技术按所处理的对象不同还可分为灰度图像增强和彩色图像增强,按增强目的还可分为光谱信息增强、空间纹理信息增强和时间信息增强等。

对图像增强效果的评价可以从定性和定量两方面进行。定性主要从人的主观感觉出发,依靠图像的视觉效果进行评价。一般从图像的清晰度、色调、纹理等几个方面进行主观评价;对图像增强的定量分析,目前并没有统一的评价标准。一般可以从图像的信息量、标准差、均值、纹理度量值和具体研究对象的光谱特征等几方面与原始图像进行比较评价。定性分析尽管具有主观性,但却可以从一幅图像中有选择地对感兴趣的特定研究对象进行重点比较和评价,因此定性分析可以对图像的局部或具体研究目标进行评价。定量分析虽然比较客观公正,但通常是对一幅图像从整体上进行统计分析,很难对图像的局部或具体对象进行评价。而且由于定量分析是对整体图像的分析,容易受到噪声等因素的影响。因此,对图像增强效果的评价一般以定性分析为主。从根本上讲,图像增强效果的好坏除与具体算法有一定关系外,还与待增强图像的数据特征有直接关系。因此,一个对某一图像效果好的增强算法不一定适合于另一个图像。一般情况下,为了得到满意的图像增强效果,常常需要同时挑选几种合适的增强算法进行相当数量的试验,从中选出视觉效果比较好的、计算量相对小的、又满足要求的最优算法。

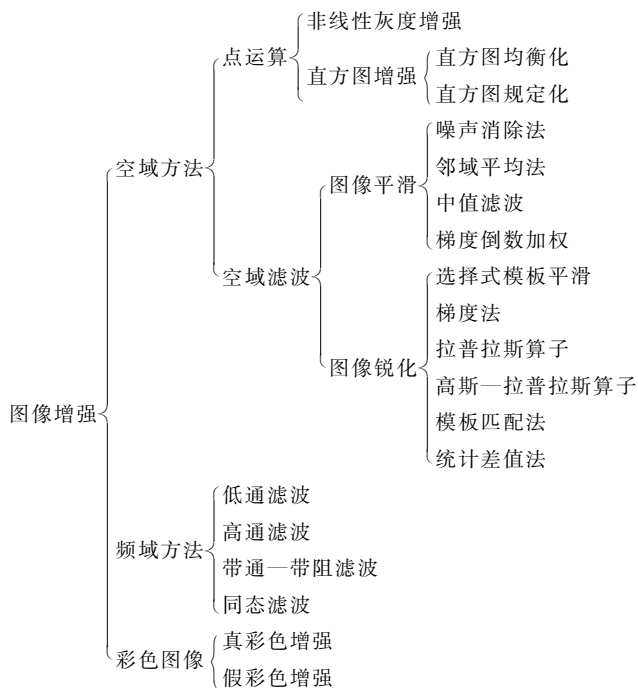


图 3-1 图像增强的方法概况

3.2 空域增强

空间域增强是指直接在图像所在的二维空间进行增强处理,即增强构成图像的像素。空间域增强方法主要有灰度变换增强、直方图变换增强、空间平滑滤波增强和图像锐化等。

3.2.1 灰度变换增强

灰度变换可使图像对比度扩展,图像清晰,特征明显。它是图像增强的重要手段。在图像空间所进行的灰度变换是一种点处理方法,它将输入图像中每个像素 (x, y) 的灰度值 $f(x, y)$,通过映射函数 $T(\cdot)$,变换成输出图像中的灰度 $g(x, y)$,即

$$g(x, y) = T[f(x, y)] \quad (3-1)$$

根据不同的应用要求,可以选择不同的变换函数,如正比函数和指数函数等。根据函数的性质,灰度变换的方法有以下几种:

- (1) 线性灰度变换。
- (2) 分段线性灰度变换。
- (3) 非线性灰度变换。

对于线性灰度变换和非线性灰度变换,是直接应用确定的变换公式依次对每个像素进行处理,也称为直接灰度变换。

1. 线性灰度变换

在曝光不足或曝光过度,或景物本身灰度就比较小的情况下,图像灰度可能会局限在某一个很小的范围内,致使图像中细节分辨不清。正如灰色钮扣掉在与钮扣颜色相似的地毯上,很难找寻,原因就是它们亮度太接近。但是,如果白色钮扣掉在黑色的地毯上,很快就能找寻出来,原因就是它们亮度反差大。同样,对于灰度局限在某一个很小范围内的数字图像,如果采用线性函数对图像的每一个像素作线性扩展,扩大像素的对比度,将有效地改善图像视觉效果。该方法也称为线性拉伸,即将输入图像(原始图像)灰度值的动态范围按线性关系公式拉伸扩展至指定范围或整个动态范围。对于常见的8位灰度图像而言,其动态范围是 $[0, 255]$ 。线性拉伸采用的变换公式一般为

$$g(x, y) = f(x, y) \cdot C + R \quad (3-2)$$

C, R 的值由输出图像的灰度值动态范围决定。

假定原始输入图像的灰度取值范围为 $[f_{\min}, f_{\max}]$,输出图像的灰度取值范围为 $[g_{\min}, g_{\max}]$,其变换公式为

$$g(x, y) = \frac{f(x, y) - f_{\min}}{f_{\max} - f_{\min}}(g_{\max} - g_{\min}) + g_{\min} \quad (3-3)$$

一般要求 $g_{\min} < f_{\min}, g_{\max} > f_{\max}$ 。对于8位灰度图像,则有

$$g(x, y) = \frac{f(x, y) - f_{\min}}{f_{\max} - f_{\min}} \times 255 \quad (3-4)$$

这一线性变换很容易具体实现,只需将原图像在 (x, y) 处的像素灰度值 $f(x, y)$ 代入式(3-3),即可得到增强后的图像在对应位置 (x, y) 处灰度值 $g(x, y)$ 。逐一扫描整幅图像,并进行同样的计算,就可以得到增强后的数字图像。对于8位灰度图像,则采用式(3-4)进

行计算。图 3-2 给出了线性拉伸的示意图,它表示将原始输入图像的灰度范围不加区别地扩展。

从图 3-2 可以看出,进行线性拉伸前,图像灰度集中在 $[a, b]$ 之间,而进行线性拉伸后,图像灰度集中在 $[a', b']$ 之间。可见,灰度范围得到了拉伸。图 3-3 给出了图像灰度变换前后效果对比。

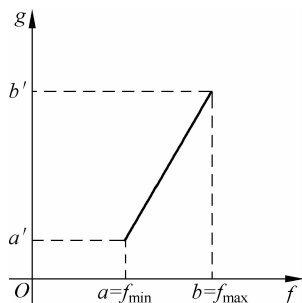


图 3-2 与式(3-3)对应的线性拉伸示意图



(a) 变换前



(b) 变换后

图 3-3 线性灰度变换

另一种情况,图像中大部分像素的灰度级在 $[f_{\min}, f_{\max}]$ 范围内,少部分像素分布在小于 $f_{\min}$ 和大于 $f_{\max}$ 的区间内,此时可用式(3-5)作变换,即

$$g(x, y) = \begin{cases} g_{\min}, & f(x, y) < f_{\min} \\ \frac{f(x, y) - f_{\min}}{f_{\max} - f_{\min}}(g_{\max} - g_{\min}) + g_{\min}, & f_{\min} \leq f(x, y) < f_{\max} \\ g_{\max}, & f(x, y) \geq f_{\max} \end{cases} \quad (3-5)$$

这种两端截取的变换将会造成一小部分信息丢失。不过有时为了某种应用,做这种牺牲是值得的。例如,利用遥感资料分析降水时,在预处理中去掉非气象信息图,既可以减少运算量,又可以提高分析精度。

2. 分段线性灰度变换

线性拉伸可以将原始输入图像中的灰度值不加区别地扩展。在实际应用中,为了突出图像中感兴趣的研究对象,常常要求局部扩展拉伸某一范围的灰度值,或对不同范围的灰度值进行不同的拉伸处理,即分段线性拉伸。例如,某一幅图像,其亮的部分和暗的部分都是背景,中间灰度值是感兴趣的区域图像,此时就可以对灰度值大和小的两部分同时进行灰度压缩,而对中间灰度值进行拉伸。即将图像灰度区间分成两段乃至多段,分别作线性变换。

一般来讲,分段线性拉伸实际上是仅将某一范围的灰度值进行拉伸,而其余范围的灰度值实际上是被压缩了。图 3-4 给出了常用的几种分段线性拉伸的示意图,其中图 3-4(a)对应的变换公式如式(3-6)所示。

分段线性变换的优点是可以根据用户的需要,拉伸特征物体的灰度细节,相对抑制不感兴趣的灰度级。

$$g(x, y) = \begin{cases} \frac{a'}{a} \cdot f(x, y), & 0 \leq f(x, y) < a \\ \frac{b' - a'}{b - a} \cdot (f(x, y) - a) + a', & a \leq f(x, y) < b \\ \frac{M' - a'}{M - a} \cdot (f(x, y) - b) + b', & b \leq f(x, y) < M \end{cases} \quad (3-6)$$

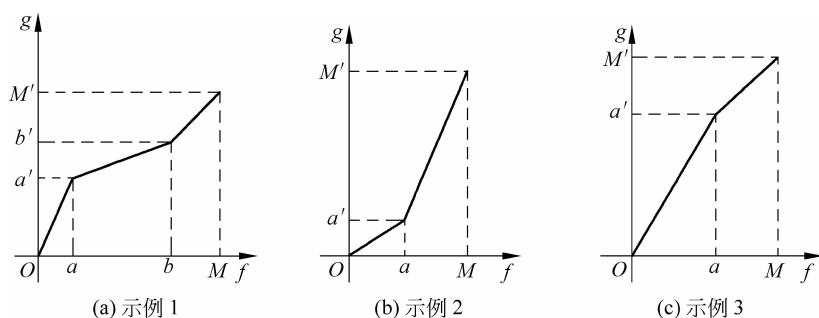


图 3-4 分段线性拉伸示例

3. 非线性灰度变换

非线性拉伸不是对图像的整个灰度范围进行扩展,而是有选择地对某一灰度值范围进行扩展,其他范围的灰度值则有可能被压缩。与分段线性拉伸不同的是,非线性拉伸不是通过在不同灰度值区间选择不同的线性方程来实现对不同灰度值区间的扩展与压缩的,非线性拉伸在整个灰度值范围内采用统一的变换函数,利用变换函数的数学性质实现对不同灰度值区间的扩展与压缩。

下面介绍常用的 3 种非线性扩展方法。

(1) 对数扩展

对数扩展的基本形式为

$$g(x, y) = \log[f(x, y)] \quad (3-7)$$

对数的底一般根据需要可以灵活选择,在实际应用中,一般取自然对数变换,具体形式为

$$g(x, y) = a + \frac{\ln[f(x, y) + 1]}{b \ln c} \quad (3-8)$$

式中, a 、 b 、 c 都是可选择的参数; $f(x, y) + 1$ 是为了避免对零求对数,确保 $\ln[f(x, y) + 1] \geq 0$ 。 $\ln[f(x, y) + 1] = 0$, $y = a$, 则 a 为 y 轴上的截距,确定了变换曲线的初始位置变换关系, b 和 c 两个参数确定变换曲线的变换速率。对数扩展可以将图像的低亮度(灰度值)区进行大幅拉伸,但是高亮度区则被压缩了。其变换函数曲线如图 3-5(a)所示。

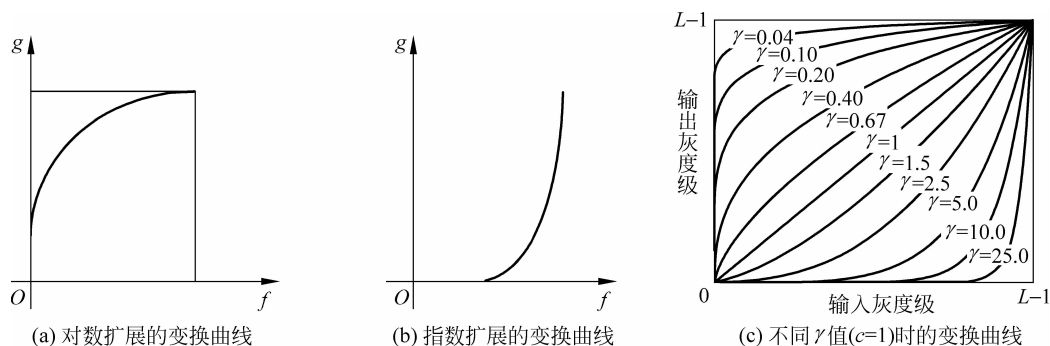


图 3-5 对数、指数、幂次扩展的变换曲线

(2) 指数扩展

指数扩展的基本形式为

$$g(x, y) = b^{f(x, y)} \quad (3-9)$$

在实际应用中,为了增加变换的动态范围,一般需要加入一些调制参数。具体形式为

$$g(x, y) = b^{[f(x, y) - a]} - 1 \quad (3-10)$$

式中,参数 a 、 b 、 c 用于调整曲线的位置和形状。其中参数 a 可以改变曲线的起始位置,参数 c 可以改变曲线的变化速率,指数扩展可以对图像的高亮度区进行大幅扩展,其变换函数曲线如图 3.5(b)所示。

(3) 幂次变换

幂次变换的基本形式为

$$g(x, y) = c [f(x, y)]^\gamma \quad (3-11)$$

式中, c 、 γ 是正常数。不同的 γ 系数对灰度变换具有不同的响应。若 $\gamma < 1$,它对低灰度的放大程度大于高灰度的放大程度,导致图像的低灰度范围得以扩展而高灰度范围得以压缩,使得图像的整体亮度提高;若 $\gamma > 1$,则相反。图 3.5(c)给出了不同 γ 值($c=1$)时的变换曲线。

图像获取、打印和显示等设备的输入/输出响应通常为非线性的,满足幂次关系。为了得到正确的输出结果,采取这种幂次关系进行校正的过程就称为 γ 校正。例如,阴极射线管显示器的输入强度与输出电压之间具有幂次关系,其 γ 值为 $1.8 \sim 2.5$,它显示的图像往往比期望的图像更暗。为了消除这种非线性变换的影响,可以在显示之前对输入图像进行相反的幂次变换,即若 $\gamma=2.5$ 且 $c=1$,则以进行校正 $\hat{g}(x, y) = [f(x, y)]^{1/2.5}$ 。于是,校正后的输入图像经显示器显示后其输出与期望输出相符,即 $g(x, y) = \hat{g}(x, y)^{2.5} = f(x, y)$ 。

幂次变换与对数变换都可以扩展与压缩图像的动态范围。相比而言,幂次变换更具有灵活性,它只需改变 γ 值就可以达到不同的增强效果。但是,对数变换在压缩动态范围方面更有效。

3.2.2 直方图变换增强

直方图变换的图像增强技术是以概率统计学理论为基础的,常用的方法有直方图均衡化技术和直方图规定化(匹配)技术。

1. 灰度直方图

灰度直方图是灰度值的函数,它描述了图像中各灰度值的像素个数。通常用横坐标表示像素的灰度级别,纵坐标表示对应的灰度级出现的频率(像素的个数)。频率的计算式为

$$p(r) = n_r \quad (3-12)$$

式中, n_r 为图像中灰度为 r 的像素数。图 3-6 是一幅原始图像和对应的灰度直方图。

常用的直方图是规格化和离散化的,即纵坐标用相对值表示。设图像总像素为 N ,某一级灰度像素数为 n_r ,则直方图为

$$p(r) = n_r / N \quad (3-13)$$

灰度直方图是图像的重要特性,它反映了一幅图像的灰度分布情况,是图像处理中最常用的统计图之一,图像的明暗状况和对比度等特征都可以通过直方图反映出来,因此,可以通过修改直方图的方法来调整图像的灰度分布情况。例如,从图 3-7(a)、(b)的两个灰度密

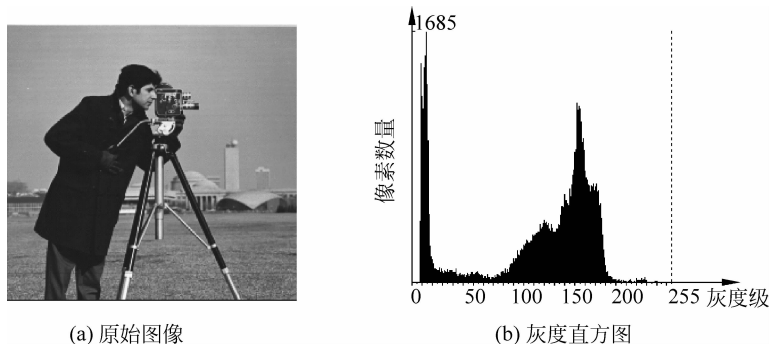


图 3-6 原始图像和对应的灰度直方图

度分布函数中可以看出,图 3-7(a)所对应的图像中大多数像素灰度值取在较暗的区域,所以这幅图像肯定较暗,一般在摄影过程中曝光过强就会造成这种结果。而图 3-7(b)所对应的图像中像素灰度值集中在亮区,因此,图 3-7(b)所对应的图像特性将偏亮,一般在摄影中曝光太弱将导致这种结果。从两幅图像的灰度分布来看图像的质量均不理想,可通过修改直方图的方法来调整图像的灰度分布。

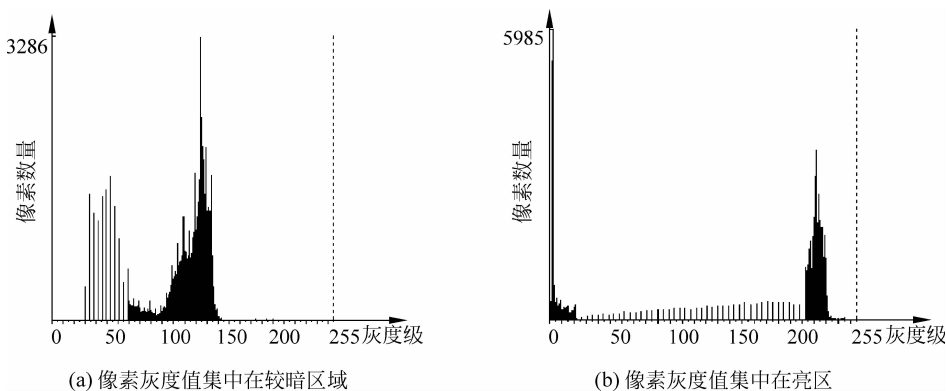


图 3-7 两幅不同的原始图像所对应的灰度直方图

2. 直方图的性质

直方图反映的是一幅图像的灰度值的概率统计特征,一幅图像的直方图基本上可以描述一幅图像的概貌,它具备以下性质:

(1) 直方图没有位置信息

由于直方图反映的是图像灰度值的概率统计特征,因此,一幅图像对应的直方图描述了该图像灰度的总体分布,无法体现像素灰度分布的位置信息。如在基于内容的图像检索中,如果两幅图像的直方图相同,只能认为这两幅图像可能是同一幅图像,但不能说肯定是同一幅图像,因为直方图反映的是整体特性,只能说这两幅图像灰度分布在整体上是相同的,但其位置分布是否一致无法确定。因此,可以利用直方图来初选,如果要进一步判断是否相同,还需考虑灰度位置分布是否相同。

(2) 直方图具有可叠加性

如果将一幅图像分成若干个区域,则每个区域都可以分别作直方图,而原图像的总直方

图为各区直方图之和。各区的形状、大小可以随意选择。例如,如果两幅图像具有相同的直方图,可以进一步划分成9个矩形区域,并分别作出直方图,如果两幅图像在9个区域对应的直方图也相同,则这两幅图像相同的概率大大增加了。

(3) 直方图的统计特征

常用的直方图的统计特征主要有以下几个:

① 矩。

$$m_i = \sum_{r=0}^{L-1} r^i p(r) \quad (3-14)$$

式中, r 为灰度级, $r=0,1,\dots,L-1$; i 为矩的阶数。

② 绝对矩。

$$m_{ia} = \sum_{r=0}^{L-1} |r|^i p(r) \quad (3-15)$$

式中, r 为灰度级, $r=0,1,\dots,L-1$; i 为矩的阶数。

③ 中心矩。

$$\mu_i = \sum_{r=0}^{L-1} (r - m_1)^i p(r) \quad (3-16)$$

式中, r 为灰度级, $r=0,1,\dots,L-1$; i 为矩的阶数。

④ 绝对中心矩。

$$\mu_{ia} = \sum_{r=0}^{L-1} |r - m_1|^i p(r) \quad (3-17)$$

式中, r 为灰度级, $r=0,1,\dots,L-1$; i 为矩的阶数。

⑤ 熵。

$$H_r = \sum_{r=0}^{L-1} p(r) \log_2 p(r) \quad (3-18)$$

式中, r 为灰度级, $r=0,1,\dots,L-1$; i 为矩的阶数。

充分利用直方图的性质,可使图像灰度直方图在图像分析中发挥更大的作用。

3. 直方图均衡化

为了改变图像整体偏暗或整体偏亮,灰度层次不丰富的情况,可以将原图像的直方图通过变换函数修正为均匀的直方图,使直方图不再偏于低端,也不再偏于高端,而是变成比较均匀的分布,这种技术叫直方图均衡化。

设 r 和 s 分别表示原始图像灰度级和经过直方图均衡化以后的图像灰度级。为便于讨论,对 r 和 s 进行归一化,使

$$0 \leq r, \quad s \leq 1$$

归一化后,对于一幅给定的图像,灰度级分布在 $0 \leq r \leq 1$ 范围内。可以对 $[0,1]$ 区间内的任一个 r 值进行以下变换,即

$$s = T(r) \quad (3-19)$$

也就是说,通过上述变换,每个原始图像的像素灰度值 r 都对应产生一个 s 值。变换函数 $s = T(r)$ 应满足下列条件:

① 在 $0 \leq r \leq 1$ 的区间内, $T(r)$ 单值单调增加。

② 对于 $0 \leq r \leq 1$, 有 $0 \leq T(r) \leq 1$ 。

这里的第①个条件保证了图像的灰度级从白到黑的次序不变。第②个条件则保证了映射变换后的像素灰度值在允许的范围内。满足这两个条件的变换函数的一个例子如图 3-8 所示。

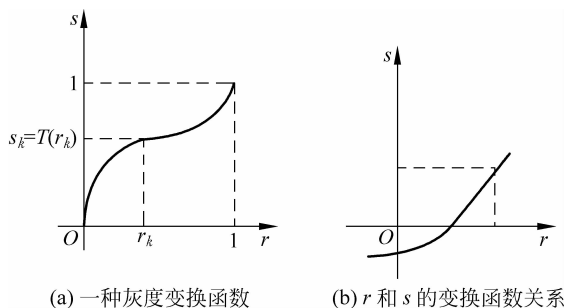


图 3-8 灰度变换示意图

从 s 到 r 的反变换可用式(3-20)表示,即

$$r = T^{-1}(s) \quad (3-20)$$

由概率理论可知,如果已知随机变量 r 的概率密度为 $P_r(r)$,而随机变量 s 是 r 的函数,即 $s = T(r)$,则 s 的概率密度为 $P_s(s)$ 可以由 $P_r(r)$ 求出。

由于直方图的物理意义是属于某灰度 r 所包含的像素总数,某一段直方图也表示该段灰度范围所包含像素总数与经单调增函数的一对一映射变换到新的一段 s 灰度后所包含像素总数应相等,因此有

$$P_s(s)ds = P_r(r)dr \quad (3-21)$$

等式两边对 s 求导,有

$$P_s(s) = \frac{d}{ds} \left[\int_{-\infty}^r P_r(r)dr \right] = P_r \frac{dr}{ds} = P_r \frac{d}{ds} [T^{-1}(s)] \quad (3-22)$$

均衡化直方图要求 $P_s(s)$ 为常数,可令

$$P_s(s) = 1$$

由式(3-22)有

$$ds = P_r(r)dr \quad (3-23)$$

两边积分,得

$$s = T(r) = \int_0^r P_r(r)dr \quad (3-24)$$

式(3-24)右边为 $P_r(r)$ 的累积分布函数。它表明当变换函数为 r 的累积分布函数时,能达到直方图均衡化的目的。

上面的修正方法是以连续随机变量为基础进行讨论的。为了对图像数字进行处理,必须引入离散形式的公式。当灰度级是离散值的时候,可用频数近似代替概率值。下面将上述结论推广到离散数字图像。

设一幅图像的像元数为 n ,共有 l 个灰度级, n_k 代表灰度级为 r_k 的像元的数目,则第 k 个灰度级出现的概率可表示为

$$P_r(r_k) = \frac{n_k}{n}, \quad 0 \leq r_k \leq 1 \quad k = 0, 1, \dots, l-1 \quad (3-25)$$

式(3-24)所表示的变换函数 $T(r)$ 可改写为

$$s_k = T(r_k) = \sum_{j=0}^k P_r(r_j) = \sum_{j=0}^k \frac{n_j}{n} \quad (3-26)$$

式中, $0 \leq r_k \leq 1, k=0, 1, \dots, l-1$ 。

可见,均衡化后各像素的灰度值 s_k 可直接由原图像的直方图算出。

4. 直方图均衡化的计算步骤及实例

下面通过两个具体的例子来说明如何对一幅图像进行直方图均衡化处理以及图像经过直方图均衡化增强后的效果。

例 3-1 假设有一幅大小为 64×64 的灰度图像,共有 8 个灰度级,其灰度级分布见表 3-1,现要求对其进行均衡化处理。

表 3-1 图像的灰度级分布表

原始直方图数据			均衡化后的直方图数据		
r_k	n_k	n_k/n	s_k	n_k	n_k/n
$r_0=0$	790	0.19	0	0	0.00
$r_1=1/7$	1023	0.25	$s_0=1/7$	790	0.19
$r_2=2/7$	850	0.21	0	0	0.00
$r_3=3/7$	656	0.16	$s_1=3/7$	1023	0.25
$r_4=4/7$	329	0.08	0	0	0.00
$r_5=5/7$	245	0.06	$s_2=5/7$	850	0.21
$r_6=6/7$	122	0.03	$s_3=6/7$	985	0.24
$r_7=1$	81	0.02	$s_4=1$	448	0.11

处理过程如下:

(1) 根据式(3-26)计算各灰度级的 s_k :

$$s_0 = T(r_0) = \sum_{j=0}^0 P_r(r_j) = P_r(r_0) = 0.19$$

$$s_1 = T(r_1) = \sum_{j=0}^1 P_r(r_j) = P_r(r_0) + P_r(r_1) = 0.19 + 0.25 = 0.44$$

依此类推,可计算得: $s_2=0.65$; $s_3=0.81$; $s_4=0.89$; $s_5=0.95$; $s_6=0.98$; $s_7=1$; 图 3-9(b)给出了 s_k 和 r_k 之间的阶梯状关系,即转换函数。

(2) 对 s_k 进行舍入处理,由于原图像的灰度级只有 8 级,因此上述各 s_k 需用 $1/7$ 为量化单位进行舍入运算,得到以下结果:

$$\begin{aligned} s_{0\text{舍入}} &= 1/7 & s_{1\text{舍入}} &= 3/7 & s_{2\text{舍入}} &= 5/7 & s_{3\text{舍入}} &= 6/7 \\ s_{4\text{舍入}} &= 1 & s_{5\text{舍入}} &= 1/7 & s_{6\text{舍入}} &= 1/7 & s_{7\text{舍入}} &= 1/7 \end{aligned}$$

(3) s_k 的最终确定,由 s_k 的舍入结果可见,均衡化后的灰度级仅有 5 个级别,分别是 $s_0=1/7$ 、 $s_1=3/7$ 、 $s_2=5/7$ 、 $s_3=6/7$ 、 $s_4=1$ 。

(4) 计算对应每个 s_k 的像素数目,因为 $r_0=0$ 映射到 $s_0=1/7$,所以有 790 个像素取 s_0 这个灰度值;同样 r_1 映射到 $s_1=3/7$,因此有 1023 个像素取值 $s_1=3/7$;同理有 850 个像素取值 $s_2=5/7$;又因为 r_3 和 r_4 都映射到 $s_3=6/7$,所以有 $656+329=985$ 个像素取此灰度

值,同样有 $245+122+81=448$ 个像素取 $s_4=1$ 的灰度值,如表 3-1 所示。

均衡化后的直方图见图 3-9(c),可以看出,在离散情况下,直方图仅能接近于均匀概率密度函数,图 3-9(c)所示的结果虽然并不是理想的均衡化结果,但与原始直方图相比已有很大改善,原始图像灰度值偏低,图像整体上偏暗,直方图均衡化后,其亮度得到了较大的提升,灰度值分布比较均衡。

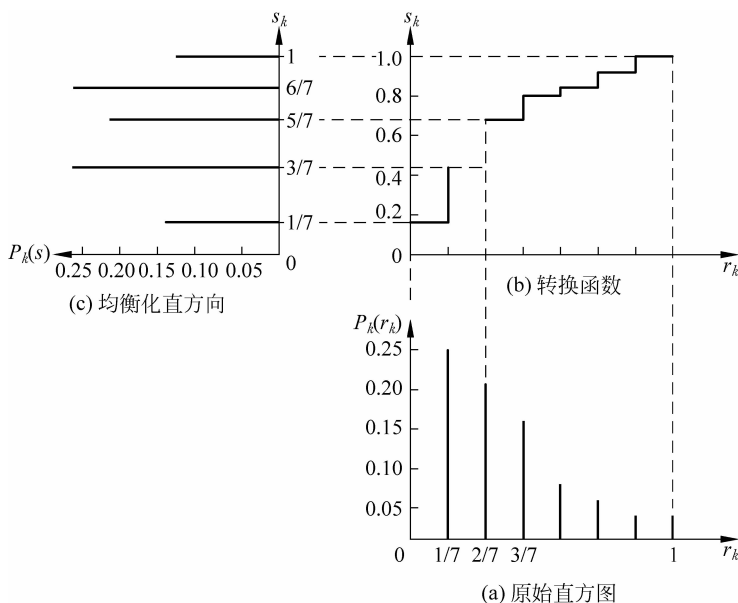


图 3-9 直方图均衡化

直方图均衡化一般会使原始图像的灰度等级减少,这是由于均衡化过程中要进行近似舍入所造成的。在例 3-2 中由 8 个灰度级缩减成了 5 个,被舍入合并的灰度级是原始图像上出现频率较低的灰度级。若这些灰度级构成的图像细节比较重要,则可以采用局部自适应的直方图均衡化技术,也可以采用增加像素位数的方法来减少由于灰度级合并所造成的灰度层次的损失。

例 3-2 直方图均衡化效果示例。图 3-10 给出了一幅图像的直方图均衡化的效果。图 3-10(a)和图 3-10(b)分别是原始图像和其直方图。由于原始图像的灰度值分布在较窄的区间,而且其亮度偏低,所以图像显得非常模糊,其相应的直方图则表现为动态范围较小且靠近坐标轴原点。经过均衡化增强后,图像的灰度值动态范围明显增加,图像的亮度也得到了提升,图像从整体上给人一种清晰的感觉(图 3-10(c)和图 3-10(d))。

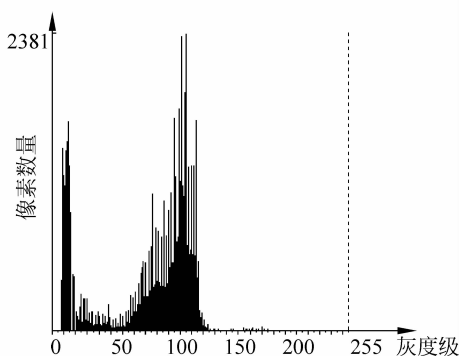
由上面的例子可见,利用累积分布函数作为灰度变换函数,经变换后得到的新灰度的直方图虽然不很平坦,但毕竟比原始图像的直方图平坦得多,而且其动态范围也大大扩展了。因此这种方法对于对比度较弱的图像进行处理是很有效的。

5. 直方图规定化

直方图均衡化的优点是能增强整个图像的对比度,提升图像的亮度,从而得到的直方图是在整个灰度级动态范围内近似均匀分布的直方图。直方图均衡化处理方法是行之有效的增强方法之一。但是,由于它的变换函数采用的是累积分布函数,因此它只能产生近似均匀



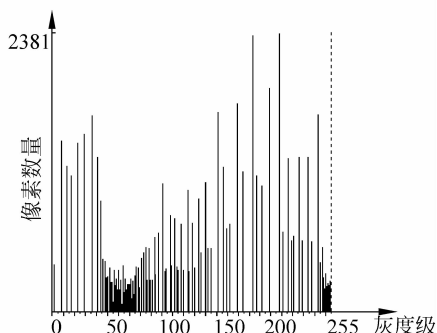
(a) 原始图像



(b) 原始图像对应的直方图



(c) 均衡化后的图像



(d) 均衡化后的图像对应的直方图

图 3-10 直方图均衡化图像

的直方图的结果,这样就必然会限制它的效能。也就是说,在实际应用中,并不是总需要具有均匀直方图的图像,有时需要具有特定直方图的图像,以便能够有目的地对图像中的某些灰度级分布范围内的图像加以增强。换句话说,希望可以人为地改变直方图的形状,使之成为某个特定的形状。直方图规定化方法就是针对上述思想提出来的一种直方图修正增强方法,它可以按照预先设定的某个形状来调整图像的直方图。下面仍然从研究连续灰度级的概率密度函数出发来讨论直方图规定化的基本思想。

假设 $P_r(r)$ 和 $P_z(z)$ 分别表示原始图像和目标图像(即希望得到的图像)灰度分布的概率密度函数,直方图规定化的目的就是调整图像的直方图,使之具有 $P_z(z)$ 所表示的形状。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。

首先对原始图像进行直方图均衡化处理,即求变换函数

$$s = T(r) = \int_0^r P_r(\omega) d\omega \quad (3-27)$$

假定已经得到了目标图像,并且它的概率密度函数是 $P_z(z)$ 。对这幅图像也可用同样的变换函数进行均衡化处理,即

$$u = G(z) = \int_0^z P_z(\omega) d\omega \quad (3-28)$$

因为对于两幅图像(注意,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 $P_s(s)$ 和 $P_u(u)$ 具有同样的均匀密度。其中,式(3-28)的逆过程为

$$z = G^{-1}(u) \quad (3-29)$$

这样,如果用从原始图像中得到的均匀灰度级 s 来代替逆过程中的 u ,其结果灰度级将是所要求的概率密度函数 $P_z(z)$ 的灰度级。

$$z = G^{-1}(u) = G^{-1}(s) \quad (3-30)$$

6. 直方图规定化的计算步骤及实例

根据以上的分析,可以总结出直方图规定化增强处理的步骤如下:

- (1) 用直方图均衡化方法将原始图像按式(3-27)做直方图均衡化处理。
- (2) 按照目标图像的灰度级概率密度函数 $P_z(z)$,并用式(3-28)得到变换函数 $G(z)$ 。
- (3) 用(1)中得到的灰度级 s 替代 u ,按式(3-30)做逆变换: $z = G^{-1}(s)$ 。

经过上述3步处理得到的新图像的灰度级将具有事先规定的概率密度 $P_z(z)$ 。在上述直方图规定化方法处理过程中包含 $T(r)$ 和 $G^{-1}(s)$ 两个变换函数,实际应用中可将这两个函数简单地组合成一个函数关系。利用这个函数关系可以从原始图像产生所希望的灰度分布。将 $s = T(r) = \int_0^r P_r(\omega) d\omega$ 代入式(3-30),有

$$z = G^{-1}[T(r)] \quad (3-31)$$

式(3-31)就是用 r 来表示 z 的公式。很显然,从该式可以看出,一幅图像不用直方图均衡化就可以实现直方图规定化,即求出 $T(r)$ 并与 $G^{-1}(s)$ 组合在一起,再对原始图像施以变换即可。如果 $G^{-1}[T(r)] = T(r)$ 时,这个式子就简化为直方图均衡化方法了。

这种方法在连续变量的情况下涉及求反变换函数解析式的问题,在一般情况下这是比较困难的事情。但是由于数字图像处理是对离散变量的处理,因此可用近似的方法绕过这个问题,从而克服该困难。

下面仍然通过两个具体实例来说明直方图规定化的具体过程及实际效果。

例 3-3 采用 64×64 像素的图像,其灰度级仍然是 8 级。其直方图如图 3-11(a)所示,图 3-11(b)是规定的直方图,图 3-11(c)为变换函数,图 3-11(d)为处理后的结果直方图。原始直方图和规定的直方图的数值分别列于表 3-2 和表 3-3 中,经过直方图均衡化处理后的直方图数值列于表 3-4 中。

表 3-2 原始直方图数据

r_k	n_k	$P_k(r_k)$
$r_0 = 0$	790	0.19
$r_1 = 1/7$	1023	0.25
$r_2 = 2/7$	850	0.21
$r_3 = 3/7$	656	0.16
$r_4 = 4/7$	329	0.08
$r_5 = 5/7$	245	0.06
$r_6 = 6/7$	122	0.03
$r_7 = 1$	81	0.02

表 3-3 规定的直方图数据

z_k	$P_k(z_k)$
$z_0 = 0$	0.00
$z_1 = 1/7$	0.00
$z_2 = 2/7$	0.00
$z_3 = 3/7$	0.15
$z_4 = 4/7$	0.20
$z_5 = 5/7$	0.30
$z_6 = 6/7$	0.20
$z_7 = 1$	0.15

表 3-4 均衡化处理后的直方图数据

$r_t \rightarrow s_k$	n_k	$P_t(s_k)$
$r_0 \rightarrow s_0 = 1/7$	790	0.19
$r_1 \rightarrow s_1 = 3/7$	1023	0.25
$r_2 \rightarrow s_2 = 5/7$	850	0.21
$r_3 + r_4 \rightarrow s_3 = 6/7$	985	0.24
$r_5 + r_6 + r_7 \rightarrow s_4 = 1$	448	0.11

具体计算步骤如下:

(1) 对原始图像进行直方图均衡化映射处理的数列于表 3-4 的 n_k 栏内。

(2) 利用式 $u_k = G(z_k) = \sum_{i=0}^k P_z(z_j)$ 计算变换函数。

$$u_0 = G(z_0) = \sum_{j=0}^0 P_z(z_j) = P_z(z_0) = 0.00$$

$$u_1 = G(z_1) = \sum_{j=0}^1 P_z(z_j) = P_z(z_0) + P_z(z_1) = 0.00$$

$$u_2 = G(z_2) = \sum_{j=0}^2 P_z(z_j) = P_z(z_0) + P_z(z_1) + P_z(z_2) = 0.00$$

$$u_3 = G(z_3) = \sum_{j=0}^3 P_z(z_j) = P_z(z_0) + P_z(z_1) + P_z(z_2) + P_z(z_3) = 0.15$$

依此类推,有

$$u_4 = G(z_4) = \sum_{j=0}^4 P_z(z_j) = 0.35$$

$$u_5 = G(z_5) = \sum_{j=0}^5 P_z(z_j) = 0.65$$

$$u_6 = G(z_6) = \sum_{j=0}^6 P_z(z_j) = 0.85$$

$$u_7 = G(z_7) = \sum_{j=0}^7 P_z(z_j) = 1$$

上述结果记录了 u_k 与 z_k 之间的正变换关系,同时也记录了两者的逆变换关系,即 $z_k = G^{-1}(u_k)$,为下一步用步骤(1)中得到的 s_k 代替 u_k 进行逆变换奠定了基础。图 3-11(c) 给出了由此确定的变换函数。

(3) 用直方图均衡化中的 s_k 进行 G 的反变换,求 $z_k = G^{-1}(s_k)$ 。

这一步实际上是近似过程,也就是说找出 s_k 与 $G(z_k)$ 的最接近的值。例如, $s_0 = 1/7 \approx 0.14$,与它最接近的是 $G(z_3) = 0.15$,所以可写成 $G^{-1}(0.15) = z_3$ 。用这样的方法可得到下列变换值:

$$\begin{aligned} s_0 = \frac{1}{7} &\rightarrow z_3 = \frac{3}{7}, & s_1 = \frac{3}{7} &\rightarrow z_4 = \frac{4}{7} \\ s_2 = \frac{5}{7} &\rightarrow z_5 = \frac{5}{7}, & s_3 = \frac{6}{7} &\rightarrow z_6 = \frac{6}{7} \\ s_4 = 1 &\rightarrow z_7 = 1 \end{aligned}$$

(4) 用 $z = G^{-1}[T(r)]$ 找出 r 与 z 之间的映射关系。根据这些映射重新分配像素的灰度级,并用 $n=4096$ 去除,可得到对原始图像直方图规定化增强的最终结果。

$$\begin{aligned} r_1 = \frac{1}{7} &\rightarrow z_4 = \frac{4}{7}, & r_2 = \frac{2}{7} &\rightarrow z_5 = \frac{5}{7} \\ r_3 = \frac{3}{7} &\rightarrow z_6 = \frac{6}{7}, & r_4 = \frac{4}{7} &\rightarrow z_6 = \frac{6}{7} \\ r_5 = \frac{5}{7} &\rightarrow z_7 = 1, & r_6 = \frac{6}{7} &\rightarrow z_7 = 1 \\ r_7 = 1 &\rightarrow z_7 = 1 \end{aligned}$$

计算结果见表 3-5。由图 3-11(d)可见,结果直方图与规定直方图之间仍存在一定的差距,结果直方图并不很接近希望的形状,与直方图均衡化的情况一样。这是由于从连续到离散的转换过程中对此近似引入了离散误差的原因造成的,只有在连续的情况下,求得准确的反变换函数才能得到准确的结果。而且在灰度级减少时,这种规定的和最后得到的直方图之间的误差有增大的趋势。但从实践应用中的情况看,尽管直方图规定化是一种近似的直方图,其增强效果还是很明显的。

表 3-5 结果直方图数据

z_k	n_k	$P_t(z_k)$
$z_0 = 0$	0	0.00
$z_1 = 1/7$	0	0.00
$z_2 = 2/7$	0	0.00
$z_3 = 3/7$	790	0.19
$z_4 = 4/7$	1023	0.25
$z_5 = 5/7$	850	0.21
$z_6 = 6/7$	985	0.24
$z_7 = 1$	448	0.11

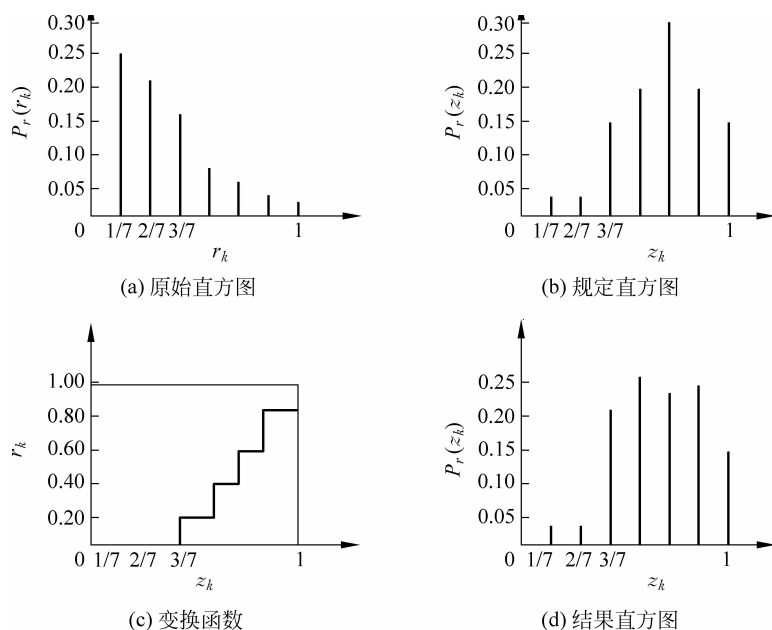


图 3-11 直方图规范化处理方法

3.2.3 空间平滑滤波增强

任何一幅原始图像,在获取和传输过程中,都会受到各种噪声的干扰,使图像退化,质量下降。图像平滑化的主要目标就是既能消除这些随机噪声,又不使图像的边缘轮廓和线条变模糊。图像平滑化处理方法有空域法和频域法两大类。本节主要讨论空域的平滑化处理方法。

空域平滑滤波器的设计比较简单,常用的有邻域平均法和中值滤波法,前者是线性的,而后者则是非线性的。

1. 邻域平均法

邻域平均法是一种直接在空间域上进行平滑的技术。该技术是基于这样一种假设:图像由许多灰度恒定的小块组成,相邻像素间存在很高的空间相关性,而噪声则相对独立。基于以上假设,可以将一个像素及其邻域内的所有像素的平均灰度值赋给平滑图像中对应的像素,从而达到平滑的目的,又称其为均值滤波或局部平滑法。

最简单的邻域平均法称为非加权邻域平均,它均等地对待邻域中的每个像素。

设有一幅图像大小为 $N \times N$ 的图像 $f(x, y)$,用邻域平均法得到的平滑图像为 $g(x, y)$,则有

$$g(x, y) = \frac{1}{M} \sum_{i, j \in s} f(i, j) \quad (3-32)$$

式中, $x, y=0, 1, \dots, N-1$; s 为 (x, y) 邻域中像素坐标的集合; M 为集合 s 内像素的总数。常用的邻域有 4-邻域和 8-邻域。

非加权邻域平均法可以用模板形式进行描述,并通过卷积求得,即在待处理图像中逐点地移动模板,求模板系数与图像中相应像素的乘积之和。

在具体实现时,模板与图像值卷积时,模板中系数 $w(0,0)$ 应位于图像对应于 (x,y) 的位置。对于一个尺寸为 $m \times n$ 的模板,假设 $m=2a+1$ 且 $n=2b+1$,这里 a,b 为非负整数,即模板长与宽通常都为奇数,如 3×3 、 5×5 、 3×5 等。图 3-12 所示为 3×3 的模板,在图像中的点 (x,y) 处,用该模板求得的响应为

$$R = w(-1, -1) \cdot f(x-1, y-1) + w(-1, 0) \cdot f(x-1, y) + \cdots \\ + w(0, 0) \cdot f(x, y) + \cdots + w(1, 0) \cdot f(x+1, y) + w(1, 1) \cdot f(x+1, y+1) \quad (3-33)$$

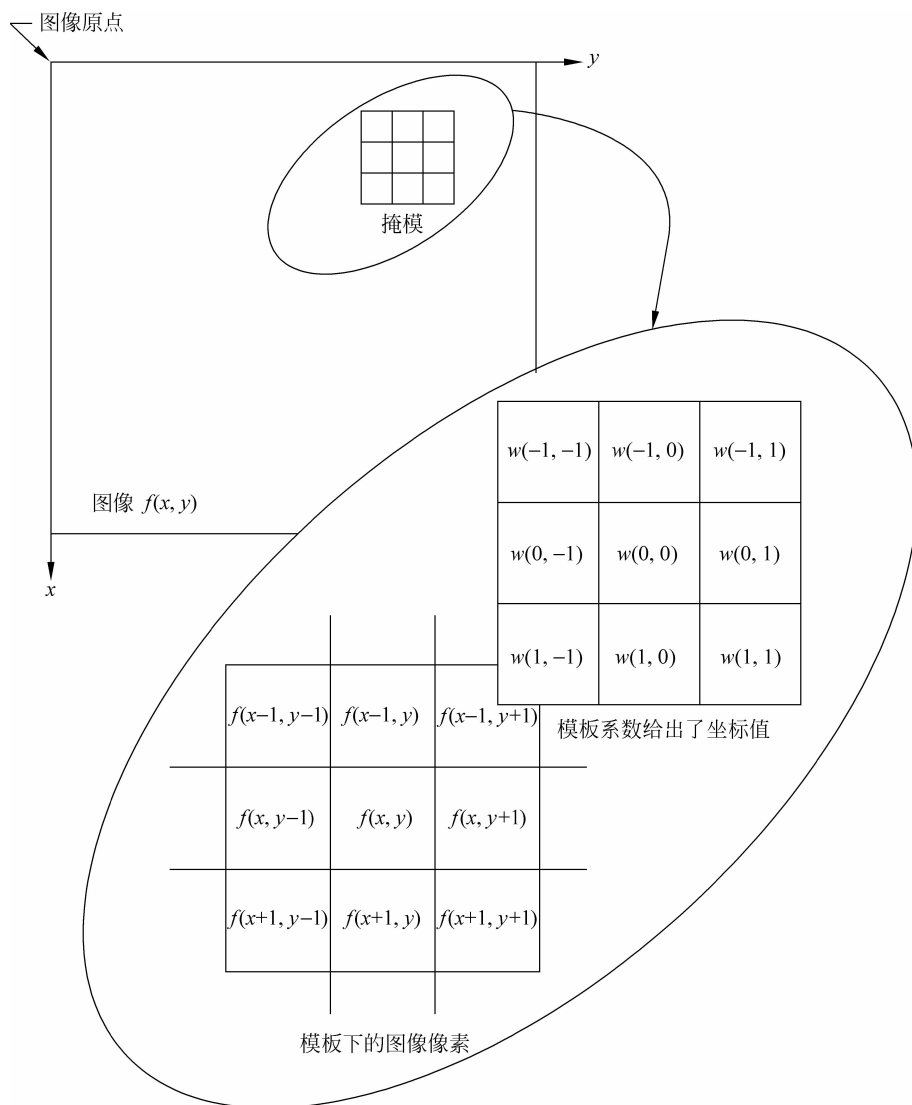


图 3-12 空间滤波过程

在非加权邻域平均法中,所有模板系数均为 1。图 3-13 给出了非加权邻域平均 3×3 模板。图 3-14 给出了邻域平均法的增强效果,图 3-14(a)所示为含有随机噪声的灰度图像,图 3-14(b)、图 3-14(c)、图 3-14(d)是分别用 3×3 、 5×5 、 7×7 模板得到的平滑图像。

$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

图 3-13 3×3 非加权均值滤波器模板

(a) 含有随机噪声的灰度图像

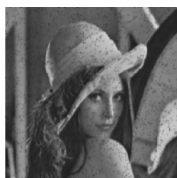
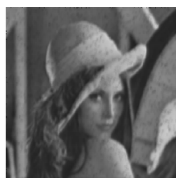
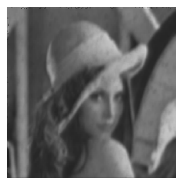
(b) 3×3 模板(c) 5×5 模板(d) 7×7 模板

图 3-14 非加权邻域平均法的增强效果

另一种邻域平均法称为加权邻域平均法。在加权邻域平均法中,所有模板系数可以有不同权值 1。图 3-15 给出了 3×3 加权均值滤波器模板,图 3-15(a)是一般形式,图 3-15(b)是一具体实例。

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

(a) 一般形式

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

(b) 具体实例

图 3-15 3×3 加权均值滤波器模板

对于一幅 $M \times N$ 的图像,经过一个 $m \times n$ (m, n 为奇数) 的加权均值滤波的过程可用式(3-34)给出,即

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)} \quad (3-34)$$

式中, $a=(m-1)/2$ 且 $b=(n-1)/2$,分母是模板系数总和,为一常数。为了得到完整的滤波图像,必须对 $x=0, 1, 2, \dots, M-1$ 和 $y=0, 1, 2, \dots, N-1$ 依次运用式(3-34)。

从图 3-15 可以看出,一些像素比另一些像素更重要。对于图 3-15(b)所示的模板,处于模板中心位置的像素比其他任何像素的权值都要大,因此,在均值计算中给定的这一像素显得更为重要。而距离模板中心较远的其他像素就显得不太重要。由于对角项离中心比正交方向相邻的像素更远,所以,它的重要性要比与中心直接相邻的 4 个像素低。把中心点加强为最高,而随着距中心点距离的增加减小系数值,是为了减小平滑处理中的模糊。也可以采取其他权重达到相同的目的。然而,图 3-15(b)所示模板中的所有系数的和应该为 16,这很便于计算机实现,因为它是 2 的整数次幂。在实践中,由于这些模板在一幅图像中所占的区域很小,通常很难看出使用图 3-15(b)所示的各种模板或用其他类似手段平滑处理后的

图像之间的区别。

例 3-4 给定图 3-16 所示的图像数据 $f(x, y)$, 图像坐标原点位于 $(0, 0)$ 处, 在仅考虑 3×3 模板覆盖的图像数据的情况下, 运用图 3-15(b) 所示的模板计算 $f(7, 5)$ (用圆圈标出) 的像素平滑后 $g(7, 5)$ 的像素值。

18	17	19	17	21	29	45	59	65	59	58	66	67	61	69	60
22	20	20	17	19	25	51	65	82	90	84	74	73	78	57	56
27	23	23	18	17	21	42	47	66	90	97	90	84	86	58	61
28	25	24	21	19	21	24	24	30	50	77	95	93	84	79	77
26	24	24	23	22	23	26	38	37	28	43	77	93	88	102	91
24	20	20	21	22	23	40	68	75	47	29	48	80	97	109	97
23	16	15	17	19	19	36	55	73	68	44	33	58	92	108	103
23	14	11	13	15	15	16	12	36	69	64	35	42	77	108	110
18	21	20	19	16	7	8	14	31	60	63	30	32	78	106	118
19	18	13	13	18	17	5	11	23	48	57	38	45	84	122	128
21	18	10	13	28	35	29	42	51	53	46	40	63	104	140	137
22	24	15	18	35	46	58	77	82	60	35	42	90	140	152	140
21	27	19	21	35	44	46	53	52	38	36	72	131	172	164	146
20	26	24	31	46	54	28	14	13	31	70	128	174	167	180	156
20	26	36	60	88	101	74	55	63	99	138	178	196	186	190	163
22	28	50	91	133	152	149	140	160	189	197	201	198	182	192	165

图 3-16 原始图像数据

解 将图 3-15(b) 所示的模板权系数和图 3-16 用矩形圈出的 3×3 邻域数据代入式(3-33), 得

$$\begin{aligned}
 g(7, 5) &= \frac{1}{16} [f(6, 4)w(-1, -1) + f(7, 4)w(-1, 0) + f(8, 4)w(-1, 1) \\
 &\quad + f(6, 5)w(0, -1) + f(7, 5)w(0, 0) + f(8, 5)w(0, 1) \\
 &\quad + f(6, 6)w(1, -1) + f(7, 6)w(1, 0) + f(8, 6)w(1, 1)] \\
 &= \frac{1}{16} [26 \times 1 + 39 \times 2 + 37 \times 1 + 40 \times 2 + 68 \times 4 + 75 \times 2 \\
 &\quad + 36 \times 1 + 55 \times 2 + 73 \times 1] \\
 &= 53.875 \approx 54
 \end{aligned}$$

在该例中, 仅仅通过对一个像素 $f(7, 5)$ 用加权平均平滑时进行了计算。在编程实现过程中, 需要扫描整幅图像, 对每一个像素进行处理。由于例子中采用 3×3 滤波模板, 因此, 在对图像滤波时, 图像最边缘一行或一列不进行处理。同理, 若采用 5×5 滤波模板, 则对图像滤波时, 图像最边缘两行或两列不进行处理。

该算法简单、运算速度快。但主要缺点是在降低噪声的同时使得图像模糊, 尤其是在边缘和细节处, 邻域越大, 模糊程度越厉害。

为了克服简单局部平均法的弊病, 可在如何选择邻域的大小、形状和方向, 如何选择参数平均的点数以及邻域各点的权重系数进行考虑。式(3-35)给出一种称为超限像素平均法的图像平滑方法, 即

$$\hat{f}(x, y) = \begin{cases} \frac{1}{M} \sum_{i, j \in S} f(i, j) & |f(x, y) - \frac{1}{M} \sum_{i, j \in S} f(i, j)| > T \\ f(x, y) & \text{否则} \end{cases} \quad (3-35)$$

超限像素平均法对抑制椒盐噪声有效, 对保护有微小灰度差的细节和纹理有效。

2. 中值滤波法

邻域平均法虽然可以平滑图像,但在消除噪声的同时,会使图像中的一些细节变得模糊。中值滤波则在消除噪声的同时还能保持图像中的细节部分,防止边缘模糊,与邻域平均法不同,中值滤波是一种非线性滤波。它首先确定一个奇数像素窗口 W ,窗口内各像素按灰度值从小到大排序后,用中间位置灰度值代替原灰度值。设增强图像在 (x,y) 的灰度值为 $f(x,y)$,增强图像在对应位置 (x,y) 的灰度值为 $g(x,y)$,则有

$$g(x,y) = \text{median}\{f(x-k, y-l), k, l \in W\} \quad (3-36)$$

式中, W 为选定窗口大小。

由于中值滤波不是通过对邻域内的所有像素求平均值来消除噪声,而是让与周围像素灰度值的差比较大的像素改取近似于周围像素灰度值的值,从而达到消除噪声的目的。

例 3-5 给定如图 3-16 所示的图像数据 $f(x,y)$,在仅考虑 3×3 窗口对应图像数据的情况下,运用 3×3 窗口计算 $f(7,5)$ 经中值滤波后在 $g(x,y)$ 的灰度值 $g(7,5)$ 。

解 将图 3-16 用矩形圈出的 3×3 邻域数据代入式(3-36),得

$$\begin{aligned} g(x,y) &= \text{median}\{26, 39, 37, 40, 68, 75, 36, 55, 73\} \\ &= \text{median}\{26, 36, 37, 39, 40, 55, 68, 73, 75\} \\ &= 40 \end{aligned}$$

图 3-17 给出了中值滤波的平滑结果,图 3-17(a)所示为含有随机噪声的灰度图像,图 3-17(b)、图 3-17(c)、图 3-17(d)是分别用 3×3 、 5×5 、 7×7 模板得到的平滑图像。从图 3-17 与图 3-14 比较可以看出,中值滤波的效果要优于均值滤波的效果,图像中的边缘轮廓比较清晰。



图 3-17 中值滤波效果

需要指出的是,应用空间平滑滤波方法编程进行图像平滑时,应注意下面两点:

(1) 模板的大小与平滑效果直接相关,模板越大,平滑的效果越明显,但会造成图像边缘信息的损失,模板的大小要在保证消除噪声的前提下尽可能保持图像的边缘信息。编程时,一般可考虑通过对话框的设置使用户可以灵活选择甚至自定义模板的大小及模板系数,模板的大小一般定为奇数,且不超过 11×11 ,如 3×3 、 5×5 、 7×7 等。

(2) 平滑处理属于邻域处理技术,因此图像边界处的行、列不能进行平滑处理,所以循环次数应小于图像的行列数,一般可取 $N-2$ (假定图像大小为 $N \times N$)。对于图像边界处的像素可以通过边界拓展的方法(如边界补 0)进行处理,也可以运用一维中值滤波单独对边缘行或列进行处理,从而达到平滑边界像素的目的。

3.3 频域增强

如前所述,变换域增强是首先经过某种变换(如傅里叶变换)将图像从空间域变换到变换域,然后在变换域对频谱进行操作和处理,再将其反变换到空间域,从而得到增强后的图像。在变换域处理中最为关键的预处理便是变换处理。这种变换一般是线性变换,其基本线性运算式是严格可逆的,并且满足一定的正交条件。在图像增强处理中,最常用的正交变换是傅里叶变换。当采用傅里叶变换进行增强时,把这种变换域增强称为频域增强。

3.3.1 傅里叶变换

傅里叶变换是大家所熟知的正交变换,在一维信号处理中得到了广泛应用。把这种处理方法推广到图像处理中是很自然的事。本节将对傅里叶变换的基本概念及算法作一些讨论。

1. 一维傅里叶变换

设 $f(x)$ 为实变量 x 的连续可积函数,则 $f(x)$ 的傅里叶变换定义为

$$F(u) = \int_{-\infty}^{+\infty} f(x) e^{-j2\pi ux} dx \quad (3-37)$$

式中, j 为虚数单位; x 为时域变量; u 为频域变量。

从 $F(u)$ 恢复 $f(x)$ 称为傅里叶反变换,定义为

$$f(x) = \int_{-\infty}^{+\infty} F(u) e^{j2\pi ux} du \quad (3-38)$$

如果令 $\omega = 2\pi u$, 则正变换和反变换分别为

$$F(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-j\omega x} dx \quad (3-39)$$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) e^{j\omega x} d\omega \quad (3-40)$$

注意,正反傅里叶变换的唯一区别是幂的符号。函数 $f(x)$ 和 $F(u)$ [或 $f(x)$ 和 $F(\omega)$] 通常称作一个傅里叶变换对,对于任一函数 $f(x)$,其傅里叶变换函数 $F(u)$ 是唯一的;反之亦然。

函数 $f(x)$ 的傅里叶变换一般是一个复数,它可以由式(3-41)表示,即

$$F(u) = R(u) + jI(u) \quad (3-41)$$

式中, $R(u)$ 、 $I(u)$ 分别为 $F(u)$ 的实部和虚部。 $F(u)$ 为复平面上的矢量,它的幅度和相角为

幅度

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} \quad (3-42)$$

相角

$$\Phi(u) = \arctan \frac{I(u)}{R(u)} \quad (3-43)$$

$|F(u)|$ 称为 $f(x)$ 的傅里叶谱,而 $\Phi(u)$ 称为相位谱。谱的平方称为 $f(x)$ 的能量谱,即

$$E(u) = |F(u)|^2 = R^2(u) + I^2(u) \quad (3-44)$$

例 3-6 $f(x)$ 为一简单函数,如图 3-18(a)所示,求其傅里叶变换 $F(u)$ 。

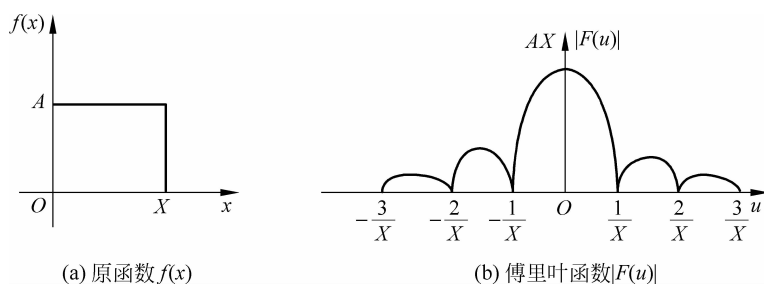


图 3-18 一维傅里叶变换举例

$$\begin{aligned}
 \text{解 } F(u) &= \int_{-\infty}^{\infty} f(x) \exp[-j2\pi ux] dx \\
 &= \int_0^X A \exp[-j2\pi ux] dx \\
 &= \frac{A}{\pi u} \sin(\pi u X) \exp[-j\pi u X]
 \end{aligned}$$

其傅里叶谱为

$$|F(u)| = AX \left| \frac{\sin(\pi u X)}{\pi u X} \right|$$

$f(x)$ 的幅度谱 $|F(u)|$ 如图 3-18(b)所示。

2. 二维傅里叶变换

傅里叶变换可推广到二维函数。如果二维函数 $f(x, y)$ 是连续可积函数,则有下面二维傅里叶变换对存在:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-j2\pi(ux + vy)] dx dy \quad (3-45)$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp[j2\pi(ux + vy)] du dv \quad (3-46)$$

与一维傅里叶变换类似,二维傅里叶变换的幅度谱和相位谱为

$$|F(u, v)| = \sqrt{R^2(u, v) + I^2(u, v)} \quad (3-47)$$

$$\Phi(u, v) = \arctan \frac{I(u, v)}{R(u, v)} \quad (3-48)$$

$$E(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v) \quad (3-49)$$

式中, $F(u, v)$ 为幅度谱; $\Phi(u, v)$ 为相位谱; $E(u, v)$ 为能量谱。

例 3-7 给定二维函数 $f(x, y)$ 如图 3-19 所示,求其傅里叶变换 $F(u, v)$ 。

$$\text{解 } f(x, y) = \begin{cases} A & 0 \leq x \leq X, 0 \leq y \leq Y \\ 0 & x > X, x < 0; y > Y, y < 0 \end{cases}$$

$$\begin{aligned}
 F(u, v) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-j2\pi(ux + vy)} dx dy \\
 &= \int_0^X \int_0^Y A e^{-j2\pi(ux + vy)} dx dy \\
 &= A \int_0^X e^{-j2\pi ux} dx \int_0^Y e^{-j2\pi vy} dy
 \end{aligned}$$

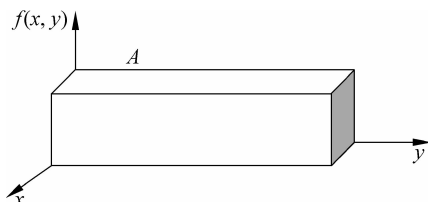


图 3-19 函数 $f(x, y)$

$$\begin{aligned}
&= A \left[-\frac{e^{-j2\pi ux}}{j2\pi u} \right]_0^X \left[-\frac{e^{-j2\pi vx}}{j2\pi v} \right]_0^Y \\
&= \left(-\frac{A}{j2\pi u} \right) [e^{-j2\pi ux} - 1] \left(-\frac{A}{j2\pi v} \right) [e^{-j2\pi vx} - 1] \\
&= AXY \left[\frac{\sin(\pi uX)e^{-j\pi ux}}{\pi uX} \right] \left[\frac{\sin(\pi vY)e^{-j\pi vy}}{\pi vY} \right]
\end{aligned}$$

其傅里叶谱表示为

$$|F(u, v)| = AXY \left| \frac{\sin(\pi uX)}{\pi uX} \right| \cdot \left| \frac{\sin(\pi vY)}{\pi vY} \right|$$

3. 离散傅里叶变换

连续函数的傅里叶变换是波形分析的有力工具,这在理论分析中无疑具有很大的价值。而离散傅里叶变换使得数学方法与计算机技术建立了联系,为傅里叶变换在实用中开辟了一条宽阔的道路。因此,它不仅具有理论价值,而且在某种意义上说也有了更重要的实用价值。

(1) 一维离散傅里叶变换

设 $f(x)$ 用 N 个互相间隔 Δx 单位的采样方法来离散化,成为一个序列,即

$$\{f(x_0), f(x_0 + \Delta x), \dots, f(x_0 + [N-1]\Delta x)\} \quad (3-50)$$

若我们规定

$$f(x) = f(x_0 + \Delta x) \quad (3-51)$$

这样可把 x 作为离散值 $x=0, 1, 2, \dots, N-1$, 就可以用 $\{f(0), f(1), \dots, f(N-1)\}$ 表示 $\{f(x_0), f(x_0 + \Delta x), \dots, f(x_0 + [N-1]\Delta x)\}$ 的等间隔的采样值序列,用这种表示法,则采样函数的离散傅里叶变换对为

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-j2\pi ux/N] \quad (3-52)$$

$$f(x) = \sum_{\mu=0}^{N-1} F(u) \exp[j2\pi ux/N] \quad (3-53)$$

式中, $u=0, 1, 2, \dots, N-1$, 也类似于 x 相应为 $0, \Delta u, 2\Delta u, \dots, (N-1)\Delta u$, 即 $F(u) = F(u\Delta u)$ 且 $\Delta u = 1/N\Delta u$ 。

式(3-52)称为离散傅里叶变换(DFT),式(3-53)则称为离散傅里叶反变换(IDFT),两者构成一个离散傅里叶变换对。由式(3-52)和式(3-53)可见,离散傅里叶变换是直接处理离散时间信号的傅里叶变换。

(2) 二维离散傅里叶变换

类似于一维傅里叶变换,对 M 行 N 列二维离散图像 $f(x, y)$ 的傅里叶变换式为

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-j2\pi ux/M + vy/N], \quad u, v = 0, 1, \dots, N-1 \quad (3-54)$$

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp[-j2\pi ux/M + vy/N], \quad x, y = 0, 1, \dots, N-1 \quad (3-55)$$

4. 离散傅里叶变换应用中的问题

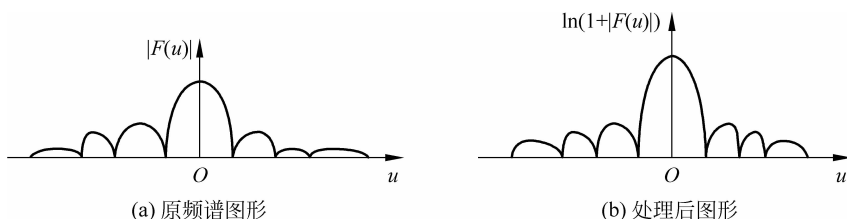
离散傅里叶变换在计算机图像处理中应用的第一个问题是计算的中间过程和结果要图像化。对 DFT 来讲不但 $f(x, y)$ 是图像, $F(u, v)$ 也要用图像来显示其结果。第二个问题是要尽量加快其计算速度。从软件角度,要不断改进算法。另一途径为专用 FFT 硬件,它不但体积小而且速度快。

(1) 频谱的图像显示

谱图像就是把 $|F(u, v)|$ 作为亮度显示在屏幕上。但在傅里叶变换中, $F(u, v)$ 随 u, v 衰减太快, 直接显示高频项只看到一两个峰, 其余都不清楚。为了符合图像处理中常用图像来显示结果的惯例, 通常用 $D(u, v)$ 来代替, 以弥补只显示 $|F(u, v)|$ 不够清楚这一缺陷。 $D(u, v)$ 定义为

$$D(u, v) = \log(1 + |F(u, v)|) \quad (3-56)$$

图 3-20 给出一维傅里叶变换原频谱 $|F(u)|$ 图形和 $D(u)$ 图形的差别。原 $|F(u)|$ 图形只有中间几个峰可见[图 3-20(a)], 图 3-20(b)所示为处理后 $D(u)$ 的图形。

图 3-20 $|F(u)|$ 的对数图像

实用的公式还要用 K 系数调整显示的图像, 即

$$D(u, v) = \log(1 + K |F(u, v)|) \quad (3-57)$$

式中, K 为常数, 常选 K 的范围在 $1 \sim 10$ 之间, 可调节显示最大值和最小值的比例。

谱的显示加深了对图像的视觉理解。如一幅遥感图像受正弦网纹的干扰, 从频谱图上立即可指出干扰的空间频率, 并可方便地从频域去除。

(2) 频谱的频域移中

常用的傅里叶正反变换公式都是以零点为中心的公式, 其结果中心最亮点却在图像的左上角, 作为周期性函数, 其中心最亮点将分布在四角, 这和正常的习惯不同, 因此, 需要把这个图像的零点移到显示的中心, 如把 $F(u, v)$ 的原零点从左上角移到显示屏的中心。

当周期为 N 时, 应在频域移动 $N/2$ 。利用傅里叶的频域移动的性质, 即

$$F(u - u_0, v - v_0) \Leftrightarrow f(x, y) \exp[j2\pi(u_0 x + v_0 y)/N] \quad (3-58)$$

当 $u_0 = v_0 = N/2$ 时,

$$\begin{aligned} \exp[j2\pi(u_0 x + v_0 y)/N] &= \exp[j\pi(x + y)] = (-1)^{x+y} \\ F(u - N/2, v - N/2) &\Leftrightarrow f(x, y) (-1)^{x+y} \end{aligned} \quad (3-59)$$

从公式可看出, 在做傅里叶变换时, 先把原图像 $f(x, y)$ 乘以 $(-1)^{x+y}$, 然后再进行傅里叶变换, 其结果谱就是移 $N/2$ 的 $F(u, v)$ 。其频谱图为 $|F(u, v)|$, 见图 3-21。应当注意, 显示是为了观看, 而实际 $F(u, v)$ 数据仍在内存中保留。

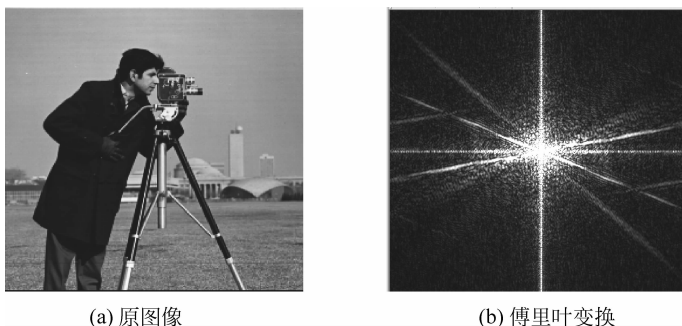


图 3-21 图像的 FFT 变换

离散傅里叶变换已成为数字信号处理的重要工具。然而,它的计算量较大,运算时间长,在某种程度上限制了它的使用范围。因此,需要有快速的运算方法。目前,从软件角度已经有快速傅里叶变换算法(FFT)。快速傅里叶算法大大提高了运算速度,在某些应用场合已可能做到实时处理,并且开始应用于控制系统。快速傅里叶变换(FFT)并不是一种新的变换,它是离散傅里叶变换的一种算法。这种方法是在分析离散傅里叶变换中的多余运算的基础上,进而消除这些重复工作的思想指导下得到的,所以在运算中大大节省了工作量,达到了快速运算的目的。从硬件角度讲也有专用FFT硬件。关于快速算法及其硬件实现这里就不作详细介绍了。

3.3.2 频域滤波增强

频域滤波增强技术是在频率域空间对图像进行滤波,因此需要将图像从空间域变换到频率域,一般通过傅里叶变换即可实现。

假定原图像 $f(x, y)$, 经傅里叶变换为 $F(u, v)$, 频率域增强就是选择合适的滤波器函数 $H(u, v)$, 对 $F(u, v)$ 的频谱成分进行调整, 然后经逆傅里叶变换得到增强的图像 $g(x, y)$ 。该过程可以通过下面流程描述, 即

$$f(x, y) \xrightarrow{\text{DFT}} F(u, v) \xrightarrow[\text{滤波}]{H(u, v)} G(u, v) \xrightarrow{\text{IDFT}} g(x, y)$$

其中,

$$G(u, v) = H(u, v) \cdot F(u, v) \quad (3-60)$$

这里, $H(u, v)$ 称为传递函数或滤波器函数。在图像增强中, 待增强的图像 $f(x, y)$ 是已知的, 因此 $F(u, v)$ 可由图像的傅里叶变换得到。实际应用中, 首先需要确定的是 $H(u, v)$, 然后就可以求得 $G(u, v)$, 对 $G(u, v)$ 求傅里叶反变换后即可得到增强的图像 $g(x, y)$ 。 $g(x, y)$ 可以突出 $f(x, y)$ 的某一方面的特征, 如利用传递函数 $H(u, v)$ 突出 $F(u, v)$ 的高频分量, 以增强图像的边缘信息, 即高通滤波; 反之, 如果突出 $F(u, v)$ 的低频分量, 就可以使图像显得比较平滑, 即低通滤波。在介绍具体的滤波器之前, 先根据以上的描述给出频域滤波的主要步骤:

- (1) 对原始图像 $f(x, y)$ 进行傅里叶变换得到 $F(u, v)$ 。
- (2) 将 $F(u, v)$ 与传递函数 $H(u, v)$ 进行卷积运算得到 $G(u, v)$ 。
- (3) 将 $G(u, v)$ 进行傅里叶逆变换得到增强图像 $g(x, y)$ 。

可以看出, 频域滤波的核心在于如何确定传递函数, 即 $H(u, v)$ 。

1. 低通滤波

图像从空间域变换到频率域后, 其低频分量对应图像中灰度值变化比较缓慢的区域, 而高频分量则表征了图像中物体的边缘和随机噪声等信息。低通滤波是指保留低频分量, 而通过滤波器函数 $H(u, v)$ 减弱或抑制高频分量的过程。因此, 低通滤波与空域中的平滑滤波器一样可以消除图像中的随机噪声, 减弱边缘效应, 起到平滑图像的作用。常用的频率域低通滤波器 $H(u, v)$ 有 4 种。

- (1) 理想低通滤波器

一个二维的理想低通滤波器的传递函数为

$$H(u, v) = \begin{cases} 1 & \text{如果 } D(u, v) \leq D_0 \\ 0 & \text{如果 } D(u, v) > D_0 \end{cases} \quad (3-61)$$

式中, D_0 为一个非负整数; D 为从点 (u, v) 到频率平面原点的距离, 即

$$D(u, v) = \sqrt{u^2 + v^2} \quad (3-62)$$

图 3-22 给出了理想低通滤波器的剖面图和三维透视图。理想低通滤波器的含义是指小于 D_0 的频率, 即以 D_0 为半径的圆内的所有频率分量可以完全无损地通过, 而圆外的频率, 即大于 D_0 的频率分量则完全被除掉。

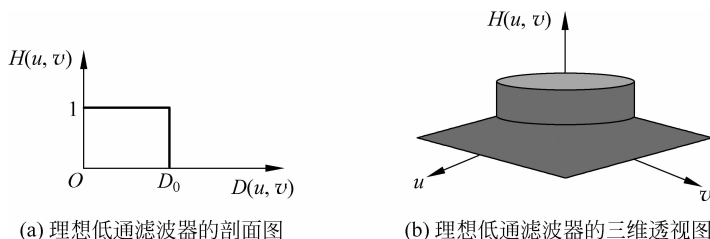


图 3-22 理想低通滤波器的剖面图和三维透视图

应当指出的是, 本节讨论的所有滤波器函数都是以坐标原点径向对称的, 对于一个图像所对应的 $N \times N$ 频率矩阵, 坐标原点已经转移到这个矩形的中心, 因而滤波器是在图 3-22(a) 所示的二维平面上, 由该剖面绕 $H(u, v)$ 轴旋转 360° 所得到的。尽管理想低通滤波器在数学上定义得非常清楚, 但在截断频率处 (D_0) 直上直下的理想低通滤波器是不能用实际的电子器件实现的。理想低通滤波器的平滑作用非常明显, 但由于变换有一个陡峭的波形, 它的反变换 $H(x, y)$ 有强烈的振铃特性。由于在变换域中 $F(u, v)$ 和 $H(u, v)$ 相乘相当于在空域中振铃波形与原图像 $f(x, y)$ 相卷积, 这种振铃使滤波后图像产生模糊效果。因此, 这种理想低通滤波实用中不被采用。图 3-23 给出了原图像及变换后有振铃现象的图像。

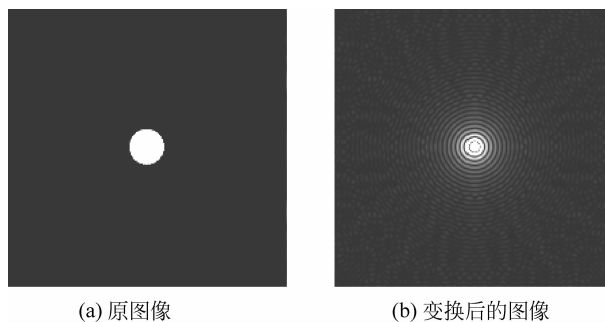


图 3-23 原图像及有振铃现象的图像

由于图像变换域中 $F(u, v)$ 的幅度随 u, v 的增加衰减很快, 也就是说其能量在变换域中集中在低频区域。以理想低通滤波作用于 $N \times N$ 的数字图像为例, 其总能量为

$$E_A = \sum_{\mu=0}^{N-1} \sum_{v=0}^{N-1} |F(u, v)|^2 = \sum_{\mu=0}^{N-1} \sum_{v=0}^{N-1} | [R^2(u, v) + I^2(u, v)]^{\frac{1}{2}} | \quad (3-63)$$

若变换域已经以原点为中心, 则当理想低通滤波的 D_0 变化时, 通过的能量和总能量 E

比值必然与 D_0 有关,而 $D_0(u,v)=[u_0^2+v_0^2]^{\frac{1}{2}}$ 可表示 u,v 的通过能量百分数。

$$\alpha = 100 \left[\sum_{\mu} \sum_v \frac{E(u,v)}{E_A} \right] \quad (3-64)$$

式中, u,v 是以 D_0 为半径的圆所包括的全部 u 和 v 。用图 3-24 表示 α 和半径的关系。这个能量比例的概念十分有用。从图中可以看出反映边缘信息的高频部分的能量是很少的,即半径为 5 的较小的范围却包含了能量的 90%,半径为 22 时已包含能量的 98%。式中 E_A 为总能量。其半径为 100 时(即 D_0)包含全部能量,这时低通滤波器半径 r 的变化和所包含总能量的关系如表 3-6 所示。

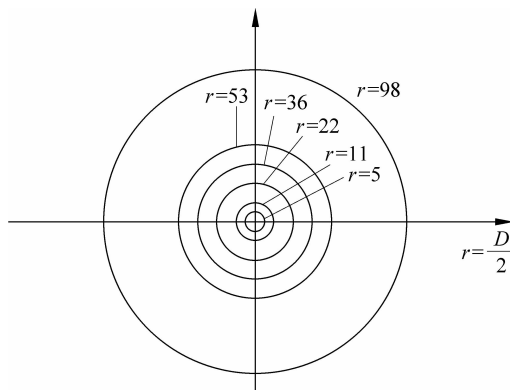


图 3-24 低通滤波的能量和 D_0 的关系

表 3-6 半径 r 与包含总能量的关系

半径 r	包含总能量/%
5	90.0
11	96.0
22	98.0
36	99.0
53	99.5
98	99.9

(2) 巴特沃斯低通滤波器

巴特沃斯(Butterworth)低通滤波器的传递函数为

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)}{D_0} \right]^{2n}} \quad (3-65)$$

式中, D_0 为截止频率; n 为函数的阶。一般情况下,取使 $H(u,v)$ 最大值下降至原来的 $1/2$ 时的 $D(u,v)$ 为截止频率 D_0 。图 3-25 给出了 $H=0.5$, 阶 $n=1$ 时的巴特沃斯低通滤波器的剖面示意图。实际应用中,有时也取使 $H(u,v)$ 最大值下降至原来的 $1/\sqrt{2}$ 的 $D(u,v)$ 为截止频率 D_0 ,其传递函数为

$$H(u,v) = \frac{1}{(\sqrt{2}-1)} \left[\frac{D(u,v)}{D_0} \right]^{2n} \quad (3-66)$$

与理想低通滤波器不同,巴特沃斯低通滤波器的特点是在通过频率与截止频率之间没有明显的不连续性,不会出现“振铃”现象,其效果好于理想低通滤波器。

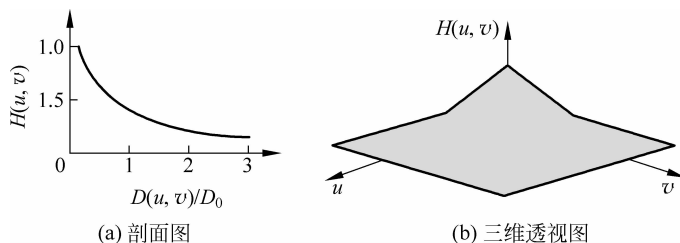


图 3-25 巴特沃斯低通滤波器的剖面图和三维透视图

(3) 指数低通滤波器

指数低通滤波器的传递函数为

$$H(u, v) = e^{-\left[\frac{D(u, v)}{D_0}\right]^n} \quad (3-67)$$

一般情况下,取使 $H(u, v)$ 最大值下降至原来的 $1/2$ 时的 $D(u, v)$ 为截止频率 D_0 , 其剖面图如图 3-26 所示。与巴特沃斯低通滤波器一样,指数低通滤波器从通过频率到截止频率之间没有明显的不连续性,而是存在一个平滑的过渡带。指数低通滤波器实用效果比巴特沃斯低通滤波器稍差,但仍无明显的振铃现象。

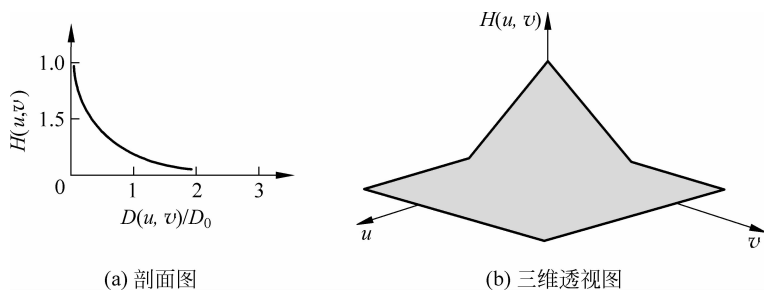


图 3-26 指数低通滤波器的剖面图和三维透视图

(4) 梯形低通滤波器

梯形低通滤波器的传递函数为

$$H(u, v) = \begin{cases} 1 & D(u, v) < D_0 \\ [D(u, v) - D_1] / (D_0 - D_1) & D_0 \leq D(u, v) \leq D_1 \\ 0 & D(u, v) > D_1 \end{cases} \quad (3-68)$$

梯形低通滤波器的剖面图如图 3-27(a) 所示。可以看出,在 D_0 的尾部包含有一部分高频分量 ($D_1 > D_0$), 因而,结果图像的清晰度较理想低通滤波器有所改善,振铃效应也有所减弱。应用时可调整 D_1 值,既能达到平滑图像的目的,又可以使图像保持足够的清晰度。

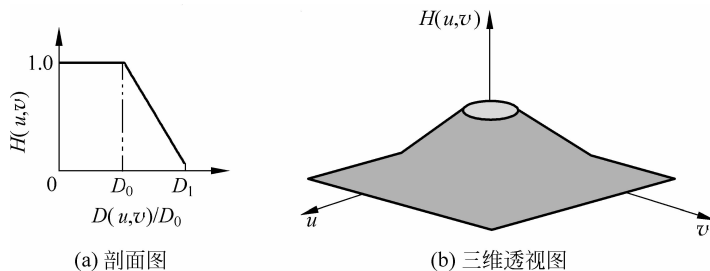


图 3-27 梯形滤波器的剖面图和三维透视图

在应用低通滤波器对图像进行处理时,一定要注意二维图像傅里叶变换的主要能量集中在低频分量,在频谱图上则体现为其能量集中在频谱的中心,图 3-28(a)所示为一幅 256×256 的图像,图 3-28(b)表示它的傅里叶频谱。当截止频率到原点的距离 $D_0=5$ 时,理想低通滤波器将保存能量的 90%,当 $D_0=11$ 时,通过的能量迅速增加,低通滤波器将保存能量的 95%,当 $D_0=22$ 时,则可以保存总能量的 98%,当 $D_0=45$ 时,则可以保存总能量的 99%,图 3-28(c)、图 3-28(d)、图 3-28(e)、图 3-28(f)分别给出了相应 D_0 的滤波效果。因此,合理地选取 D_0 是应用低通滤波器平滑图像的关键。

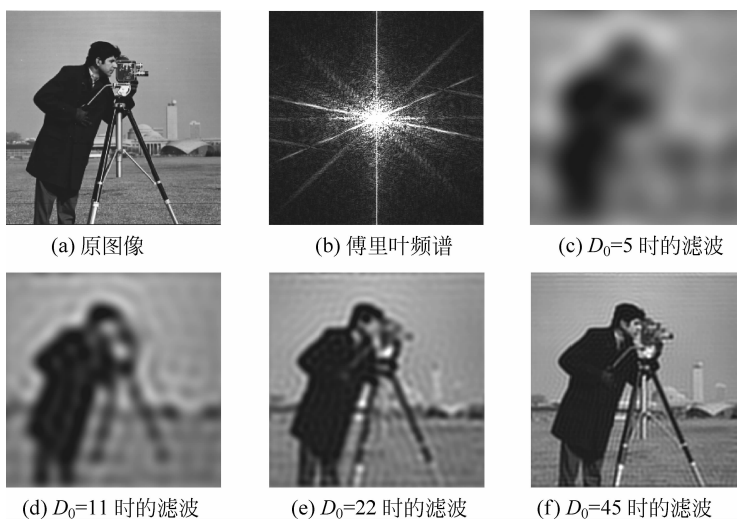


图 3-28 低通滤波结果

2. 高通滤波

图像的边缘、细节主要在高频部分得到反映,而图像的模糊是由于高频成分比较弱产生的。为了消除模糊,突出边缘,可以采用高通滤波的方法,使低频分量得到抑制,从而达到增强高频分量,使图像的边缘或线条变得清晰,实现图像的锐化。常用的高通滤波器有以下几种:

(1) 理想高频滤波器

理想高频滤波器的转移函数为

$$H(u, v) = \begin{cases} 0 & D(u, v) \leq D_0 \\ 1 & D(u, v) > D_0 \end{cases} \quad (3-69)$$

式中, D_0 为截止频率, $D(u, v) = (u^2 + v^2)^{1/2}$ 为点 (u, v) 到频率平面原点的距离。它的透视图和剖面图分别如图 3-29 所示。

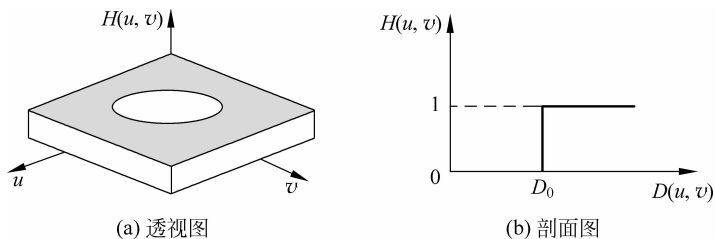


图 3-29 理想高通滤波器

(2) 巴特沃斯滤波器

n 阶高通具有 D_0 截止频率的巴特沃斯高通滤波器滤波函数定义为

$$H(u, v) = \frac{1}{\left[1 + \left(\frac{D_0}{D(u, v)}\right)^{2n}\right]} \quad (3-70)$$

和低通滤波一样, $D(u, v)$ 定义为从点 (u, v) 到频率平面原点的距离, 即

$$D(u, v) = \sqrt{u^2 + v^2} \quad (3-71)$$

同样, 对截止频率的定义有两种: 一种是在 D_0 点 $H(u, v)$ 降至 $1/2$, 另一种是降至 $1/\sqrt{2}$ 。常用 $1/\sqrt{2}$ 半功率点的定义, 则有

$$H(u, v) = \frac{1}{\left[1 + (\sqrt{2} - 1) \left(\frac{D_0}{D(u, v)}\right)^{2n}\right]} \quad (3-72)$$

(3) 指数滤波器

具有截止频率为 D_0 的指数高通滤波函数的转移函数定义为

$$H(u, v) = \exp\left[-\left(\frac{D_0}{D(u, v)}\right)^n\right] \quad (3-73)$$

同样, n 可控制指数函数 $H(u, v)$ 的增长率。当 $H(u, v)$ 降至 $1/\sqrt{2}$ 时为 D_0 , 则

$$H(u, v) = \exp\left[\left(\ln \frac{1}{\sqrt{2}}\right) \left(\frac{D_0}{D(u, v)}\right)^n\right] \quad (3-74)$$

(4) 梯形高通滤波器

梯形高通滤波器的滤波函数由式(3-75)给出, 即

$$H(u, v) = \begin{cases} 0 & D(u, v) < D_1 \\ (D_0 - D_1)/(D(u, v) - D_1) & D_1 \leq D(u, v) \leq D_0 \\ 1 & D(u, v) > D_0 \end{cases} \quad (3-75)$$

式中, 设 $D_0 > D_1$, 把第一个转折点定义为 D_0 , 而 D_1 只要小于 D_0 即可。

以上 4 种滤波函数的选用类似于低通。理想高通有明显振铃, 从而使图像的边缘模糊不清。而巴特沃斯高通效果较好, 但计算复杂, 其优点是有少量低频通过, 故 $H(u, v)$ 是渐变的, 振铃不明显; 指数高通效果比巴特沃斯差些, 但振铃也不明显; 梯形高通的效果是微有振铃, 但计算简单, 故较常用。

从以上 4 种高通滤波的讨论可知, 加强了高频分量, 但是由于低频通过太少, 故处理后图像仍不清晰。理想的方案是把多种处理方法综合使用。例如, 每次频域处理之后, 再用空域处理, 如直方图均衡化修正, 这样处理后, 图像效果更好, 轮廓更加突出, 图像更加清晰。

3. 带阻滤波与带通滤波

在某些情况下, 信号或图像中的有用成分和希望除掉的成分主要分别出现在频谱的不同频段, 这时允许或阻止特定频段通过的传递函数就非常有用。例如, 遥感图像传到地面时常有网络状干扰信号, 这种干扰信号用频谱分析的观点看是原图像和另一干扰图像的叠加。若在频域中设法把这些频带去掉或阻挡掉, 再反变换到空域, 就可把空域网络图形干扰去掉。这就叫做带阻滤波。干扰的邻域图形多为 (u_0, v_0) 和 $(-u_0, -v_0)$ 两点成对出现, 如图 3-30(a) 所示。这样可在点 (u_0, v_0) 和 $(-u_0, -v_0)$ 某个圆形邻域 D 处设计带阻滤波器, 即抑制以 (u_0, v_0) 为中心, D_0 为半径的邻域中所有频率都阻止通过的滤波器。它的滤波函数为

$$H_R(u, v) = \begin{cases} 0 & D(u, v) \leq D_0 \\ 1 & D(u, v) > D_0 \end{cases} \quad (3-76)$$

式中, $D_0 = [(u-u_0)^2 + (v-v_0)^2]^{\frac{1}{2}}$ 。

若消去围绕原点的一个径向频带, 理想的带阻滤波器函数为

$$H_R(u, v) = \begin{cases} 1 & \text{若 } D(u, v) < D_0 - W/2 \\ 0 & \text{若 } D_0 - W/2 \leq D(u, v) \leq D_0 + W/2 \\ 1 & \text{若 } D(u, v) > D_0 + W/2 \end{cases} \quad (3-77)$$

式中, W 为阻带的宽度; D_0 为阻带的中心半径, 如图 3-30(b) 所示。

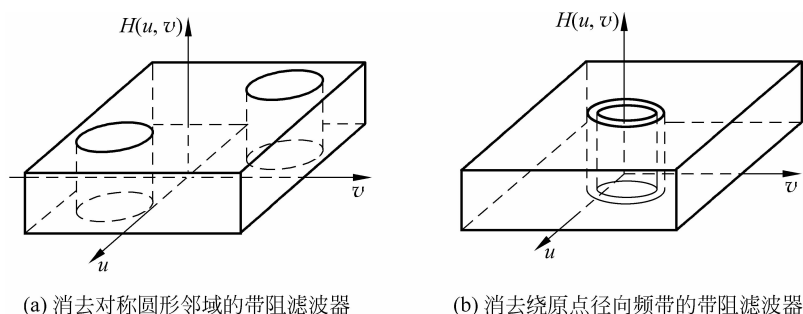


图 3-30 带阻滤波

和带阻滤波器相反的是带通滤波器, 它的公式可由带阻滤波器的公式得到。带通滤波器 $H_P(u, v)$ 可用带阻滤波器公式表示为

$$H_P(u, v) = -[H_R(u, v) - 1] \quad (3-78)$$

4. 同态滤波

图像 $f(x, y)$ 是由光源产生的照度场 $i(x, y)$ 和目标(景物或照片)的反射系数场 $r(x, y)$ 的共同作用下产生的, 且前者可以表达成后两者的乘积, 即

$$f(x, y) = i(x, y)r(x, y) \quad (3-79)$$

上述“照明-反射”模型可以用作频率域中同时压缩图像的亮度范围和增强图像的对比度的基础。但是, 在频率域中却不能直接对照度场和反射系数场频率分量分别进行独立的操作。因为根据傅里叶变换理论及式(3-79), 在频率域中, 两者是不可分的。换句话说, $F\{f(x, y)\} \neq F\{i(x, y)\}F\{r(x, y)\}$, 这里符号 $F\{\}$ 表示对 $\{\}$ 中的函数进行傅里叶变换。然而, 如果定义

$$z(x, y) = \ln f(x, y) = \ln i(x, y) + \ln r(x, y) \quad (3-80)$$

则有

$$F\{z(x, y)\} = F\{\ln f(x, y)\} = F\{\ln i(x, y)\} + F\{\ln r(x, y)\} \quad (3-81)$$

或者

$$Z(u, v) = I(u, v) + R(u, v) \quad (3-82)$$

式中, $I(u, v)$ 、 $R(u, v)$ 分别是 $\ln i(x, y)$ 和 $\ln r(x, y)$ 的傅里叶变换。

如果通过一个传递函数 $H(u, v)$ 对 $Z(u, v)$ 进行处理, 根据式(3-82), 有

$$S(u, v) = H(u, v)Z(u, v) = H(u, v)I(u, v) + H(u, v)R(u, v) \quad (3-83)$$

式中, $S(u, v)$ 为所要求的结果的傅里叶变换。在空间域中, 有

$$\begin{aligned} s(x, y) &= F^{-1}\{H(u, v)Z(u, v)\} \\ &= F^{-1}\{H(u, v)I(u, v)\} + F^{-1}\{H(u, v)R(u, v)\} \end{aligned} \quad (3-84)$$

若令

$$i'(x, y) = F^{-1}\{H(u, v)I(u, v)\} \quad (3-85)$$

以及

$$r'(x, y) = F^{-1}\{H(u, v)R(u, v)\} \quad (3-86)$$

则式(3-84)可以表示成

$$s(x, y) = i'(x, y) + r'(x, y) \quad (3-87)$$

最后, 由于 $z(x, y)$ 可以通过对原始图像取自然对数而得到, 因此可以通过对式(3-87)求自然数的逆过程而获得所要的增强后的图像 $g(x, y)$ 。也就是说

$$\begin{aligned} g(x, y) &= \exp[s(x, y)] = \exp[i'(x, y)]\exp[r'(x, y)] \\ &= i_0(x, y)r_0(x, y) \end{aligned} \quad (3-88)$$

这里

$$i_0(x, y) = \exp[i'(x, y)] \quad (3-89)$$

$$r_0(x, y) = \exp[r'(x, y)] \quad (3-90)$$

分别是输出图像的照明成分和反射成分。

同态滤波过程可以用图 3-31 描述。从图 3-31 可以看出, 同态滤波方法的关键在于利用式(3-82)的形式将图像中的照明分量和反射分量分开。这样同态滤波函数 $H(u, v)$ 就可以分别作用在这两个分量上。由于图像中的照明分量往往具有变化缓慢的特征, 而反射分量则倾向于剧烈变化, 特别是在不同物体的交界处, 所以图像对数的傅里叶变换后的低频部分主要对应照度分量, 而高频部分对应反射分量。这样可以设计一个对傅里叶变换结果的高频和低频分量影响不同的滤波函数 $H(u, v)$ 。选择 $H_L < 1, H_H > 1, H(u, v)$ 就会一方面减弱图像中的低频分量, 而另一方面加强图像中的高频分量, 最终结果是既压缩了图像的动态范围又增加了图像相邻各部分之间的对比度, 因此可以用于消除图像中的乘性噪声。图 3-32 给出了同态滤波函数的剖面图, 图 3-33 给出了一个利用同态滤波进行增强的实例。

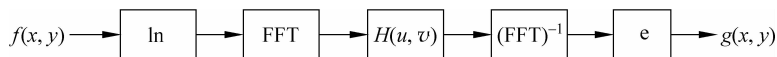


图 3-31 图像增强的同态滤波法

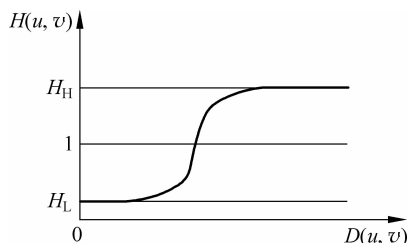


图 3-32 同态滤波函数的剖面图

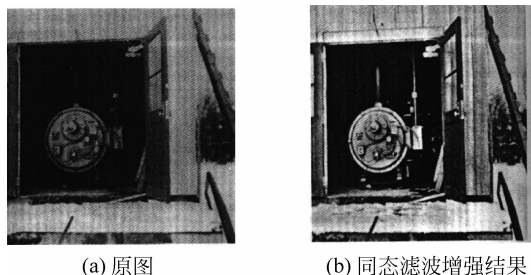


图 3-33 同态滤波增强实例

3.4 图像的锐化

图像锐化的目的是使灰度反差增强,从而增强图像中边缘信息,有利于轮廓抽取。因为轮廓或边缘就是图像中灰度变化率最大的地方。因此,为了把轮廓抽取出来,就要找一种方法把图像的最大灰度变化处找出来。

3.4.1 基于一阶微分的图像增强——梯度算子

梯度对应的是一阶导数,梯度算子是一阶导数算子。对一个图像 $f(x, y)$ 函数,它在位置 (x, y) 处的梯度可定义为梯度算子,即

$$G[f(x, y)] = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \quad (3-91)$$

梯度算子是图像处理中最常用的一阶微分算法,式(3-91)中 $f(x, y)$ 表示图像的灰度值,图像梯度的最重要性质是梯度方向,是在图像灰度最大变化率上,它恰好可以反映出图像边缘上的灰度变化。其定义为

$$\phi(x, y) = \arctan \frac{\left(\frac{\partial f}{\partial y} \right)}{\left(\frac{\partial f}{\partial x} \right)} \quad (3-92)$$

以上两式中的偏导数需要对每个像素位置计算,而实际上在数字图像中求导数是利用一阶差分近似一阶微分来进行的,即可以用一阶差分代替一阶微分,有

$$\begin{cases} \frac{\partial f}{\partial x} = f(i, j+1) - f(i, j) \\ \frac{\partial f}{\partial y} = f(i, j) - f(i+1, j) \end{cases} \quad (3-93)$$

求梯度时对于平方和运算及开方运算,可以用两个分量的绝对值之和表示,即

$$G(i, j) \approx \left\{ \left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right\}^{\frac{1}{2}} \approx \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right| \quad (3-94)$$

其中, j 对应于 X 轴方向, i 对应于 Y 负轴方向。对于边缘检测算子的数学分析,读者可能看起来有些吃力,不过不要紧,因为在实际中常用小区域模板卷积法来近似计算。其基本思想就是对式(3-91)中, X 、 Y 方向的偏导数各用一个模板,即需要两个模板组合起来以构成一个梯度算子。也就是说,对于边缘检测,有若干个检测模板(或边缘检测矩阵)可以直接实现检测功能。下面介绍几种常用的边缘检测实现算子和检测模板。其中 $G(i, j)$ 表示处理后的 (i, j) 点的灰度值, $f(i, j)$ 表示处理前该点的灰度值。求梯度幅值时对于平方和及开方运算,可以用两个分量的绝对值之和来表示。

除梯度算子外,还可以采用 Roberts 算子、Prewitt 和 Sobel 算子来计算梯度。

1. Roberts 算子

Roberts(罗伯特)边缘检测算子又称为梯度交叉算子,是一种利用局部差分算子寻找边缘的算子。梯度幅值计算近似方法如图 3-34 所示。

(i, j)	$(i, j+1)$
$(i+1, j)$	$(i+1, j+1)$

图 3-34 Roberts 算子梯度幅值计算示意图

(i, j) 为当前像素的位置, 其计算式为

$$G(i, j) = |f(i, j) - f(i+1, j+1)| + |f(i+1, j) - f(i, j+1)| \quad (3-95)$$

它是由两个 2×2 模板组成。用卷积模板表示为

$$G(i, j) = |G_x| + |G_y| \quad (3-96)$$

式中, $G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, 标注 $\cdot$ 的是当前像素的位置。

2. Sobel 算子

Roberts 算子简单直观, 但边缘检测的效果显然不好。1970 年前后 Sobel 提出了一个算子, 这就是 Sobel 算子。Sobel 算子梯度幅值计算如图 3-35 所示。 (i, j) 为当前像素点, 梯度幅值计算式为

$$\begin{aligned} G(i, j) = & |f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1) - f(i-1, j-1) \\ & - 2f(i, j-1) - f(i+1, j-1)| \\ & + |f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1) - f(i+1, j-1) \\ & - 2f(i+1, j) - f(i+1, j+1)| \end{aligned} \quad (3-97)$$

为了简化计算公式的表述, 将图 3-35 中的坐标对应记为图 3-36 所示的符号。

$(i-1, j-1)$	$(i-1, j)$	$(i-1, j+1)$
$(i, j-1)$	(i, j)	$(i, j+1)$
$(i+1, j-1)$	$(i+1, j)$	$(i+1, j+1)$

图 3-35 Sobel 算子中各个像素点的关系

a_0	a_1	a_2
a_7	(i, j)	a_3
a_6	a_5	a_4

图 3-36 梯度幅值计算示意图

因此式(3-97)则可以简记为

$$G(i, j) = |a_2 + ca_3 + a_4 - a_0 - ca_7 - a_8| + |a_0 + ca_1 + a_2 - a_6 - ca_5 - a_4| \quad (3-98)$$

式中, $c=2$ 。用卷积模板来实现, 即

$$G(i, j) = |S_x| + |S_y| \quad (3-99)$$

式中, $S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$, $S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ 。 S_x 为水平模板, 对水平边缘响应最大; S_y 是垂直模板, 对垂直边缘响应最大。图像中的每个点都用这两个模板做卷积, 两个模板卷积的最大值作为该点的输出值。其运算结果是一幅边缘幅度图像。这一算子把重点放在接近于模板中心的像素点。因此 Sobel 算子是边缘检测中最常用的算子之一。

Sobel 梯度算子是先做成加权平均, 再微分, 然后求梯度, 即用式(3-97)、式(3-98)和式(3-99)来实现的。

3. 编程举例

在视图类中定义响应菜单命令的边缘检测 Sobel 算子实现灰度图像边缘检测的函数, 代码如下:

```

void CDibView::OnSobel()
{
    CClientDC pDC(this);
    HDC hDC = pDC.GetSafeHdc();    //获取当前设备上下文的句柄
    SetStretchBltMode(hDC, COLORONCOLOR);
    HANDLE data1handle;
    LPDIBHDRMAPINFOHEADER lpDIBHdr;
    CDibDoc * pDoc = GetDocument();
    HDIB hdib;
    unsigned char * lpDIBBits;
    unsigned char * data;
    hdib = pDoc->m_hdib;    //得到图像数据
    lpDIBHdr = (LPDIBHDRMAPINFOHEADER)GlobalLock((HGLOBAL)hdib);
    lpDIBBits = lpDIBHdr + * (LPDWORD)lpDIBHdr + 256 * sizeof(RGBQUAD);
    //得到指向位图像素值的指针
    data1handle = GlobalAlloc(GMEM_SHARE, WIDTHBYTES(lpDIBHdr->biWidth * 8) * lpDIBHdr->
biHeight);    //申请存放处理后的像素值的缓冲区
    data = (unsigned char *)GlobalLock((HGLOBAL)data1handle);
    AfxGetApp()->BeginWaitCursor();
    int i, j, buf, buf1, buf2;
    for(j = 0; j<biHeight; j++)//以下循环求(x,y)位置的灰度值
        for(i = 0; i<biWidth; i++)
        {
            if(((i-1)>=0)&&((i+1)<biWidth)&&((j-1)>=0)&&((j+1)<biHeight))
            {
                //对于图像四周边界处的像素点不处理
                buf1 = (int) * (lpDIBBits + (i+1) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j-1))
                    + 2 * (int) * (lpDIBBits + (i+1) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j))
                    + (int) * (int) * (lpDIBBits + (i+1) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j+1));
                buf1 = buf1 - (int) * (int) * (lpDIBBits + (i-1) * WIDTHBYTES(lpDIBHdr->biWidth * 8)
                    + (j-1))
                    - 2 * (int) * (int) * (lpDIBBits + (i-1) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j))
                    - (int) * (int) * (lpDIBBits + (i-1) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j+1));
                //以上是对图像进行水平(x)方向的加权微分
                buf2 = (int) * (int) * (lpDIBBits + (i-1) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j+1))
                    + 2 * (int) * (int) * (lpDIBBits + (i) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j+1))
                    + (int) * (int) * (lpDIBBits + (i+1) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j+1));
                buf2 = buf2 - (int) * (int) * (lpDIBBits + (i-1) * WIDTHBYTES(lpDIBHdr->biWidth * 8)
                    + (j-1))
                    - 2 * (int) * (int) * (lpDIBBits + (i) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j-1))
                    - (int) * (int) * (lpDIBBits + (i+1) * WIDTHBYTES(lpDIBHdr->biWidth * 8) + (j-1));
                //以上是对图像进行垂直(y)方向的加权微分
                buf = abs(buf1) + abs(buf2); //求梯度
                if(buf>255) buf = 255;
                if(buf<0) buf = 0;
                * (data + i * WIDTHBYTES(lpDIBHdr->biWidth * 8) + j) = (BYTE)buf;
            }
            else * (data + i * lpDIBHdr->biWidth + j) = (BYTE)0;
        }
    for(j = 0; j<biHeight; j++)
    for(i = 0; i<biWidth; i++)
        * (lpDIBBits + i * WIDTHBYTES(lpDIBHdr->biWidth * 8) + j) = * (data + i * WIDTHBYTES(lpDIBHdr->
biWidth * 8) + j);    //处理后的数据写回原缓冲区
    StretchDIBits(hDC, 0, 0, lpDIBHdr->biWidth, lpDIBHdr->biHeight, 0, 0,
lpDIBHdr->biWidth, lpDIBHdr->biHeight,

```

```
lpDIBBits, (LPDIBHRTMAPINFO)lpDIBHdr,
DIB_RGB_COLORS,
SRCCOPY);
}
```

边缘检测实验用原始图像如图 3-37(a)所示。Sobel 算子对图 3-37(a)的检测效果如图 3-37(b)所示。

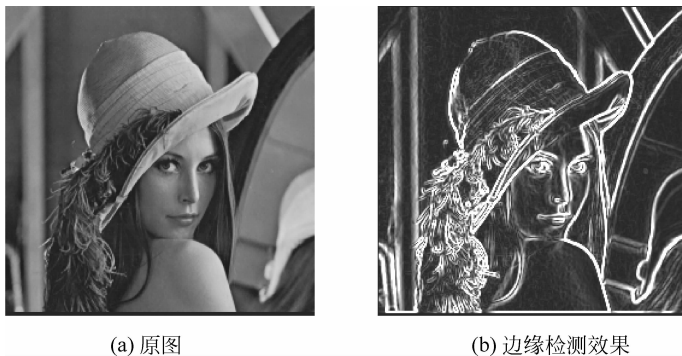


图 3-37 边缘检测结果

3.4.2 基于二阶微分的图像增强——拉普拉斯算子

拉普拉斯(Laplace)算子是不依赖于边缘方向的二阶微分算子,是常用的二阶导数算子。对一个连续函数 $f(x, y)$,它在位置 (x, y) 的拉普拉斯表示式,即

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (3-100)$$

对于数字图像来说,计算函数的拉普拉斯值也可以借助于各种模板来实现。拉普拉斯对模板的基本要求是对应中心像素的系数应该是正的,而对应于中心像素邻近像素的系数应该是负的,它们的和应该为零。拉普拉斯算子通常可以简单表示为

$$G(i, j) = |4f(i, j) - f(i+1, j) - f(i-1, j) - f(i, j+1) - f(i, j-1)| \quad (3-101)$$

或者

$$G(i, j) = \begin{vmatrix} 8f(i, j) - f(i-1, j-1) - f(i-1, j+1) \\ -f(i-1, j) - f(i+1, j) - f(i+1, j-1) \\ -f(i+1, j+1) - f(i, j+1) - f(i, j-1) \end{vmatrix} \quad (3-102)$$

也就是说,拉普拉斯算子常用两种模板来进行检测,其模板分别见图 3-38(a)和图 3-38(b)。

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

(a) 模板一

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

(b) 模板二

图 3-38 拉普拉斯算子模板

拉普拉斯算子是一个标量而不是矢量,具有线性特性和旋转不变性,即各向同性的性质,它常常用在图像处理的过程中。但由于拉普拉斯算子是一种二阶导数算子,它将在边缘

处产生一个陡峭的零交叉。因此,经过拉普拉斯算子滤波过的图像具有零平均灰度。也正是由于拉普拉斯算子是二阶微分算子,对图像中的噪声相当敏感。另外,它也常产生双像素宽的边缘,且不能提供边缘方向信息,拉普拉斯算子很少直接用于检测边缘,而主要用于已知边缘像素后确定该像素是在图像的暗区还是明区一边。

3.5 彩色图像增强

前面介绍的图像增强技术都是对灰度图像进行处理的,而且生成的结果也是灰度图像。本节的彩色增强技术处理的对象虽然也是灰度图像,但生成的结果却是彩色图像。众所周知,人的视觉系统对色彩非常敏感,人眼一般能区分的灰度等级只有 20 多个,但是能区分有不同亮度、色度和饱和度的几千种颜色。根据人的这个特点,可将彩色用于增强中,以提高图像的可鉴别性。因此,如果能将一幅灰度图像变成彩色图像,就可以达到增强图像的视觉效果。常用的彩色增强方法有真彩色增强技术、假彩色增强技术和伪彩色增强技术 3 种。前两种方法着眼于对多幅灰度图像的合成处理,一般是将 3 幅图像分别作为红、绿、蓝 3 个通道进行合成。伪彩色增强技术与前两者不同,它是对一幅灰度图像的处理,通过一定的方法,将一幅灰度图像变换生成一幅彩色图像。下面分别对伪彩色、真彩色和假彩色增强技术做详细论述。

3.5.1 伪彩色增强

伪彩色(Pseudo Coloring)增强是把一幅黑白图像的每个不同灰度级,按照线性或非线性的映射函数变换成不同的彩色,与彩色空间中的一点相匹配,得到一幅彩色图像的技术。它使原图像细节更易辨认、目标更容易识别。伪彩色增强的方法主要有以下 3 种。

1. 密度分割法

密度分割法也称强度分割法,是伪彩色增强中最简单而又最常用的一种方法,它是对图像的灰度值动态范围进行分割,使分割后的每一灰度值区间甚至每一灰度值本身对应某一种颜色。如图 3-39(a)、图 3-39(b)所示。具体而言,假定把一幅图像看作一个二维的强度函数,可以用一个平行于图像坐标平面的平面(称为密度切割平面)去切割图像的强度函数,这样强度函数在分割处被分为上、下两部分,即两个灰度值区间。如果再对每一个区间赋予某种颜色,就可以将原来的灰度图像变换成只有两种颜色的图像。更进一步,如果用多个密度切割平面对图像函数进行分割,那么就可以将图像的灰度值动态范围切割成多个区间,每一个区间赋予某一种颜色,则原来的一幅灰度图像就可以变成一幅彩色图像。特别地,如果将每一个灰度值都划分成一个区间,如将 8bit 灰度图像划分成 256 个区间,就是索引图像,从这个意义上讲,可以认为索引图像是由灰度图像经密度分割生成的。

如果用 N 个平面去切割图像,则可以得到 $N+1$ 个灰度值区间,每一个区间对应一种颜色 C_i 。对于每一像元 (x, y) ,如果 $I_{i-1} \leq f(x, y) \leq I_i$,则有

$$g(x, y) = C_i \quad i = 1, 2, \dots, N \quad (3-103)$$

$g(x, y)$ 和 $f(x, y)$ 分别表示变换后的彩色图像和原始灰度图像。这样便可以把一幅灰度图像变成一幅伪彩色图像。此法比较直观、简单,缺点是变换出的彩色数目有限。

应当指出,每一灰度值区间赋予何种颜色,是由具体应用所决定的,并无规律可言。但

总的来讲,相邻灰度值区间的颜色差别不宜太小也不宜太大,太小将无法反映细节上的差异,太大则会导致图像出现不连续性。实际应用中,密度切割平面之间可以是等间隔的,也可以是不等间隔的,而且切割平面的划分也应依据具体的应用范围和研究对象而定。

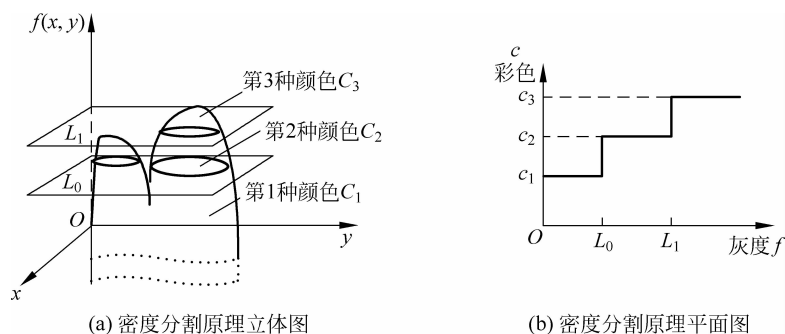


图 3-39 密度分割原理

2. 空间域灰度级——彩色变换

密度分割法实质是通过一个分段线性函数实现从灰度到彩色的变换,每个像元只经过一个变换对应到某一种颜色。与密度分割不同,空间域灰度级——彩色变换是一种更为常用的、比密度分割更有效的伪彩色增强法。变换过程如图 3-40 所示,它是根据色度学的原理,将原图像 $f(x, y)$ 中每一个像元的灰度值分别经过红、绿、蓝 3 种独立变换,即 $T_R(\cdot)$ 、 $T_G(\cdot)$ 和 $T_B(\cdot)$,变成红、绿、蓝三基色分量,即 $R(x, y)$ 、 $G(x, y)$ 、 $B(x, y)$ 分量图像,然后用它们分别去控制彩色显示器的红、绿、蓝电子枪,便可以在彩色显示器的屏幕上合成一幅彩色图像。3 个变换是独立的,彩色的含量由变换函数 $T_R(\cdot)$ 、 $T_G(\cdot)$ 和 $T_B(\cdot)$ 的形状而定。但在实际应用中这 3 个变换函数一般取同一类的函数,如可以取带绝对值的正弦函数,也可以取线性变换函数。典型的变换函数如图 3-41 所示,灰度值范围为 $[0, L]$,每个变换取不同的分段线性函数。可以看出,最小的灰度值(0)对应蓝色,中间的灰度值($L/2$)对应绿色,最高的灰度值(L)对应红色,其余的灰度值则分别对应不同的颜色。其中图 3-41(a)、图 3-41(b)、图 3-41(c)分别为红、绿、蓝 3 种变换函数,而图 3-41(d)是把 3 种变换画在同一张图上以便看清楚互相间的关系。由图 3-41(d)可见,只有在灰度为零时呈蓝色,灰度为 $L/2$ 时呈绿色,灰度为 L 时呈红色,灰度为其他值时将由三基色混合成不同的色调。

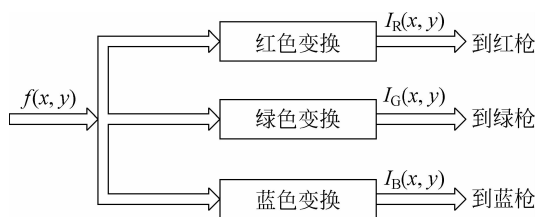


图 3-40 灰度级——彩色变换过程

3. 频率域伪彩色增强

频率域伪彩色增强首先把黑白图像从空间域经傅里叶变换变到频率域,然后在频率域内用 3 个不同传递特性的滤波器(如高通、带通/带阻、低通)将图像分离成 3 个独立的分量,

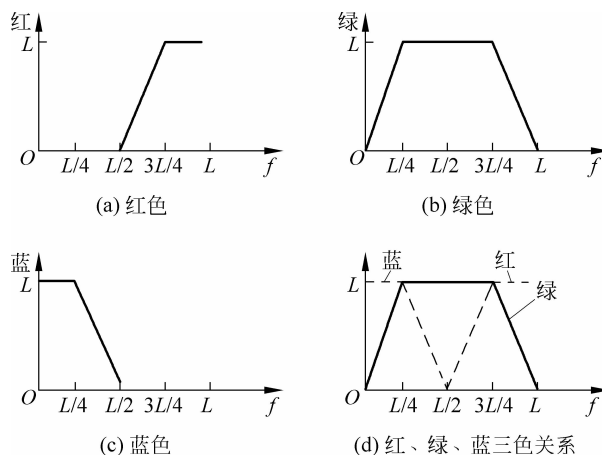


图 3-41 典型的变换函数

对每个范围内的频率分量分别进行傅里叶反变换,得到 3 幅代表不同频率分量的单色图像,接着对这 3 幅图像做进一步的处理(如直方图均衡化),最后将它们作为三基色分量分别加到彩色显示器的红、绿、蓝显示通道,从而实现频率域的伪彩色增强,如图 3-42 所示。

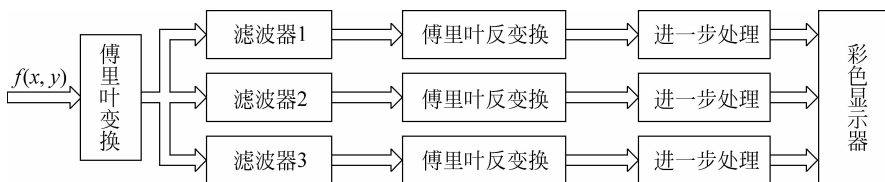


图 3-42 频率域伪彩色增强原理

3.5.2 假彩色增强

假彩色处理(False Coloring)是将一幅图像或多光谱图像映射到 RGB 空间中心位置上的过程。假彩色处理是经常出现的一个操作过程,如调节彩色电视机的色调、饱和度的过程实际就是假彩色处理。又如红光成像设备拍摄了 N 幅不同波段上的 N 幅图像: $f_1(x, y)$, $f_2(x, y), \dots, f_N(x, y)$, 可以将它们经过假彩色处理再现出可见光谱图像,其处理函数为

$$R(x, y) = F_R[f_1(x, y), f_2(x, y), \dots, f_N(x, y)] \quad (3-104)$$

$$G(x, y) = F_G[f_1(x, y), f_2(x, y), \dots, f_N(x, y)] \quad (3-105)$$

$$B(x, y) = F_B[f_1(x, y), f_2(x, y), \dots, f_N(x, y)] \quad (3-106)$$

式中, F_R 、 F_G 、 F_B 为映射函数; $R(x, y)$ 、 $G(x, y)$ 、 $B(x, y)$ 为显示空间三基色分量。

伪彩色或假彩色处理都不改变图像像素的几何位置,而仅仅改变其颜色。因此可以与肉眼色觉特性相结合设计它们的映射函数 F_R 、 F_G 、 F_B , 提高人眼对图像的分辨能力。该技术已被广泛应用于遥感、医学图像处理中。

3.5.3 真彩色增强

自然物体的彩色叫真彩色,把能真实反映自然物体本来颜色的图像叫真彩色图像。真

彩色图像可由彩色摄像机摄制,并由彩色监视器近似复原。然而,在没有彩色摄像机的情况下,也可以通过真彩色增强技术实现真彩色处理。

正如第2章所介绍,任何一幅真彩色图像可由红、绿、蓝三基色混合而成。因此,在处理过程中,首先用加有红色滤色片的摄像机(黑白摄像机)摄取彩色图像,图像信号经数字化送入一块图像存储板存起来,第二步用带有绿色滤色片的摄像机摄取图像,图像信号经数字化送入第二块图像存储板,最后用带有蓝色滤色片的摄像机摄取图像,图像数据存储在第三块图像存储板内。3幅图像数据准备好后就可以在系统的输出设备——彩色监视器上合成一幅真彩色图像。其原理如图3-43所示。

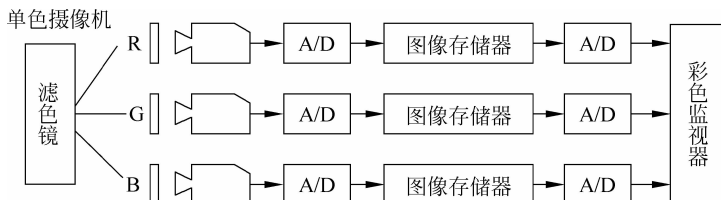


图 3-43 彩色图像合成原理

3.6 小结

本章在介绍图像增强的基本概念的基础上,对增强方法进行了分类,并重点介绍了空域增强、频域增强、图像锐化和彩色增强的一些主要方法。

基于空间域的增强方法直接在图像所在的二维空间进行处理,即增强构成图像的像素。空间域增强方法主要有灰度变换增强、直方图增强、图像平滑和图像锐化等。基于空间域的增强方法按照所采用的技术不同,可分为灰度变换和空域滤波两种方法。灰度变换是基于点操作的增强方法,它将每一个像素的灰度值按照一定的数学变换公式转换为一个新的灰度值,如增强处理中常用的对比度增强、直方图均衡化等方法。

对比度增强可以采用线性拉伸和非线性拉伸。线性拉伸可以将原始输入图像中的灰度值不加区别地扩展。在实际应用中,为了突出图像中感兴趣的研究对象,常常要求局部扩展拉伸某一范围的灰度值,或对不同范围的灰度值进行不同的拉伸处理,即分段线性拉伸。非线性拉伸在整个灰度值范围内采用统一的非线性变换函数,利用变换函数的数学性质实现对不同灰度值区间的扩展与压缩。

为了改变图像整体偏暗或整体偏亮、灰度层次不丰富的状况,可以将原图像的直方图通过变换函数修正为均匀的直方图,使直方图不再偏于低端,也不再偏于高端,而是变成比较均匀的分布,这种技术叫直方图均衡化。直方图均衡化一般会使原始图像的灰度等级减少,这是由于均衡化过程中要进行近似舍入造成的。在实际应用中,有时需要具有特定直方图的图像,以便能够有目的地对图像中的某些灰度级分布范围内的图像加以增强,此时可采用直方图规定化方法按照预先设定的某个形状来调整图像的直方图,从而达到增强图像效果的目的。

空域滤波是基于邻域处理的增强方法,它应用某一模板对每个像素及其周围邻域的所有像素进行某种数学运算,得到该像素的新的灰度值,输出值的大小不仅与该像素的灰度值

有关,而且还与其邻域内的像素的灰度值有关,常用的图像平滑与锐化技术就属于空域滤波的范畴。图像平滑化的主要目标是消除随机噪声的同时,又不使图像的边缘轮廓和线条变模糊。图像平滑化处理方法有空域法和频域法两大类。空域平滑滤波器的设计比较简单,常用的有邻域均值法和中值滤波法。邻域平均法是一种直接在空间域上进行平滑的技术。该技术是基于这样一种假设:图像由许多灰度恒定的小块组成,相邻像素间存在很高的空间相关性,而噪声则相对独立。因此,可以将一个像素及其邻域内的所有像素的平均灰度值赋给平滑图像中对应的像素,从而达到平滑的目的。最简单的邻域平均法称为非加权邻域平均法,它均等地对待邻域中的每个像素。非加权邻域平均法可以用模板形式进行描述,并通过卷积求得,即在待处理图像中逐点地移动模板,求模板系数与图像中相应像素的乘积之和。另一种邻域平均法称为加权邻域平均法。在非加权邻域平均法中,所有模板系数可以有不同的权值。邻域平均法虽然可以平滑图像,但在消除噪声的同时,会使图像中的一些细节变得模糊。中值滤波则在消除噪声的同时还能保持图像中的细节部分,防止边缘模糊,与邻域平均法不同,中值滤波是一种非线性滤波。它首先确定一个奇数像素窗口,窗口内各像素按灰度值从小到大排序后,用中间位置灰度值代替原灰度值。

基于频率域的增强方法则是首先经过傅里叶变换将图像从空间域变换到频率域,然后在频率域对频谱进行操作和处理,再将其反变换到空间域,从而得到增强后的图像。基于频率域的增强方法主要有低通滤波和高通滤波。低通滤波是指保留低频分量,而通过滤波器函数减弱或抑制高频分量的过程。目的是消除图像中的随机噪声,减弱边缘效应,起到平滑图像的作用。常用的低通滤波器有理想低通滤波器、巴特沃斯低通滤波器、指数低通滤波器、梯形低通滤波器等。高通滤波是指抑制低频分量,增强高频分量。目的是为了使图像的边沿或线条变得清晰,实现图像的锐化。常用的高通滤波器有理想高频滤波器、巴特沃斯高通滤波器、指数高通滤波器、梯形高通滤波器。在某些情况下,图像中的有用成分主要分别出现在频谱的不同频段,这时允许或阻止特定频段通过的传递函数就非常有用。若在频域中设法把这些频带去掉或阻挡掉,再反变换到空域,就可把空域网络图形干扰去掉,这就叫做带阻滤波。

图像锐化的目的是使灰度反差增强,从而增强图像的边缘信息,有利于轮廓抽取。因为轮廓或边缘就是图像中灰度变化率最大的地方。因此,为了把轮廓抽取出来,就是要找一种方法把图像的最大灰度变化处找出来。常用的图像锐化方法有基于一阶微分的梯度算子、Roberts 算子、Prewitt 和 Sobel 算子以及基于二阶微分的拉普拉斯算子等。

彩色增强生成的结果是彩色图像。常用的彩色增强方法有真彩色增强技术、假彩色增强技术和伪彩色增强技术 3 种。前两种方法着眼于对多幅灰度图像的合成处理,一般是将 3 幅图像分别作为红、绿、蓝 3 个通道进行合成。伪彩色增强技术与前两者不同,它是对一幅灰度图像的处理,通过一定的方法,将一幅灰度图像变换生成一幅彩色图像。

习题

1. 请写出图 3-4(b)和图 3-4(c)所对应的变换公式。
2. 假定有 64×64 大小的图像,灰度为 16 级,概率分布如下表,试进行直方图均衡化,并画出处理前后的直方图。

r	n_k	$P_r(r_k)$
$r_0=0$	800	0.195
$r_1=1/15$	650	0.160
$r_2=2/15$	600	0.147
$r_3=3/15$	430	0.106
$r_4=4/15$	300	0.073
$r_5=5/15$	230	0.056
$r_6=6/15$	200	0.049
$r_7=7/15$	170	0.041
$r_8=8/15$	150	0.037
$r_9=9/15$	130	0.031
$r_{10}=10/15$	110	0.027
$r_{11}=11/15$	96	0.013
$r_{12}=12/15$	80	0.019
$r_{13}=13/15$	70	0.017
$r_{14}=14/15$	50	0.012
$r_{15}=1$	30	0.007

3. 对下图作 3×3 的均值滤波和中值滤波处理, 写出处理结果并比较。

1	7	1	8	1	7	1	1
1	1	1	5	1	1	1	1
1	1	5	5	5	1	1	7
1	1	5	5	5	1	8	1
8	1	1	5	1	1	1	1
8	1	1	5	1	1	8	1
1	1	1	5	1	1	1	1
1	7	1	8	1	7	1	1

- 求信号 $x(t) = \sin(t)$ 的傅里叶变换。
- 请写出频域滤波增强的基本流程。
- 比较理想高通滤波器和低通滤波器的异同点。
- 比较理想低通滤波器、巴特沃斯低通滤波器、指数低通滤波器和梯形低通滤波器。
- 为什么同态滤波可以在压缩图像的动态范围的同时增加图像的对比度?
- 使用 Sobel 算子计算第 3 题给出的图像的梯度, 并画出梯度幅度图。
- 什么是伪彩色增强处理? 其主要目的是什么?