

第3章 空域增强：模板操作

图像是由其基本单元——像素组成的,像素在图像空间是按某种规律排列的,互相之间有一定的联系。在图像处理中,可以且需要根据像素之间的联系来对图像进行加工。

在真实图像中,相邻或接近的像素之间有更密切的联系,常可结合在一起考虑。在图像处理中,常用模板来组合相邻或接近的像素,根据这些像素的统计特性或局部运算来进行操作,称为模板操作或模板运算。利用模板操作来进行图像增强常称为滤波,可以是线性的也可以是非线性的。由于模板操作涉及图像中的局部区域,所以也可方便地进行局部增强。

根据上述的讨论,本章各节将安排如下。

3.1 节先对像素的邻域以及像素间的各种联系,包括像素间的邻接、连接和连通,像素集合的邻接和连通,以及像素之间的距离进行介绍。

3.2 节讨论模板操作的基本原理和方法(主要是模板卷积操作和模板排序操作),还讨论模板运算功能的分类情况。

3.3 节介绍一些典型的利用模板操作实现的线性滤波方法,其功能包括平滑和锐化图像。

3.4 节介绍一些典型的非线性滤波方法,其功能也包括平滑和锐化图像。非线性滤波也可与线性滤波方法结合使用。

3.5 节分析利用模板操作进行图像局部增强的原理、思路和效果。

3.1 像素间联系

模板操作涉及对一组像素的同时操作,为此需要对像素间联系有一定了解。像素之间的联系有多种,既有空间上的联系也有幅度上的联系。下面介绍邻域、邻接、连接和连通等基本概念。

3.1.1 像素的邻域和邻接

对一个像素来说,与它关系最密切的常是它的邻近像素/近邻像素,它们组成该像素的**邻域**。根据对一个坐标为 (x,y) 的像素 p 的近邻像素的不同定义,可以得到由不同近邻像素所组成的不同的邻域。常见的像素邻域主要有如下3种形式(更多形式可见本套书中册)。

(1) 4-邻域 $N_4(p)$

它由像素 p 的水平(左,右)和垂直(上,下)共4个近邻像素组成,这些近邻像素的坐标分别是 $(x+1,y)$, $(x-1,y)$, $(x,y+1)$, $(x,y-1)$ 。图3.1.1(a)给出4-邻域的一个示例,组

成 p 的 4-邻域的 4 个像素均用 r 表示,它们与 p 有公共的边。

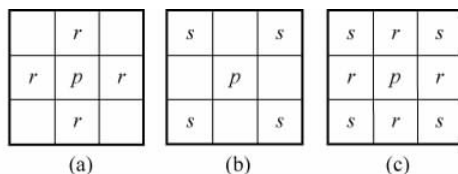


图 3.1.1 像素的邻域

(2) 对角邻域 $N_D(p)$

它由 p 的对角(左上,右上,左下,右下)共 4 个近邻像素组成,这些近邻像素的坐标分别是 $(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$ 。图 3.1.1(b)给出对角邻域的一个示例,组成 p 的对角邻域的 4 个像素均用 s 表示,它们与 p 有公共的顶角。对角邻域一般不单独使用。

(3) 8-邻域 $N_8(p)$

它由 p 的 4 个 4-邻域像素加上 4 个对角邻域像素合起来构成。图 3.1.1(c)给出 8-邻域的一个示例,其中组成 p 的 8-邻域的 4 个 4-邻域像素用 r 表示,4 个对角邻域像素用 s 表示。

需要指出,根据上述对邻域的定义,如果像素 p 本身处在图像的边缘,则它的 $N_4(p)$ 、 $N_D(p)$ 和 $N_8(p)$ 中的若干个像素会落在图像之外。在图 3.1.1 中,如果将 p 的 8-邻域看作一幅 3×3 的图像,考虑一下 $N_4(r)$ 、 $N_D(s)$ 、 $N_8(r)$ 和 $N_8(s)$,就很容易理解这种情况。处理这种情况的方法可见 3.2 节。

在上述定义的像素邻域中,一个像素与其邻域中的像素是有接触的,也称为邻接的。图像中两个像素是否邻接就看它们是否接触。邻接表示了一种像素间的空间接近关系。

根据像素邻域的不同,邻接也对应分成 3 种: **4-邻接**, **对角邻接**, **8-邻接**。

3.1.2 像素间的连接和连通

两个像素的邻接仅与它们的空间位置有关,而像素间的连接和连通还要考虑像素的属性值(以下讨论中以灰度值为例)之间的关系。

1. 像素的连接

对两个像素来说,要确定它们是否**连接**需要考虑两点: ①它们在空间上是否邻接; ②它们的灰度值是否满足某个特定的相似准则(例如它们灰度值相等,或同在一个灰度值集合中取值)。举例来说,在一幅只有 0 和 1 灰度的二值图中,只有当一个像素和在它邻域中的像素具有相同的灰度值时才可以说是连接的。

设用 V 表示定义连接的灰度值集合。例如在一幅二值图中,为考虑两个灰度值为 1 的像素之间的连接,可取 $V = \{1\}$ 。又如在一幅有 256 个灰度级的灰度图中,考虑灰度值为 128~150 的两个像素的连接时,取 $V = \{128, 129, \dots, 149, 150\}$ 。参见图 3.1.1,可讨论以下两种常用的连接。

(1) **4-连接**: 2 个像素 p 和 r 在 V 中取值且 r 在 $N_4(p)$ 中,则它们为 4-连接。

(2) **8-连接**: 2 个像素 p 和 r 在 V 中取值且 r 在 $N_8(p)$ 中,则它们为 8-连接。

可以看出,两个连接的像素一定是邻接的,但两个邻接的像素不一定是连接的。

2. 像素的连通

在像素连接的基础上,可进一步讨论和定义像素之间的**连通**。实际上,像素连通可以看作是像素连接的一种推广。为讨论连通先来定义两个像素间的通路。从具有坐标 (x,y) 的像素 p 到具有坐标 (s,t) 的像素 q 的一条通路由一系列具有坐标 $(x_0,y_0),(x_1,y_1),\dots,(x_n,y_n)$ 的独立像素组成。这里 $(x_0,y_0)=(x,y),(x_n,y_n)=(s,t)$,且 (x_i,y_i) 与 (x_{i-1},y_{i-1}) 邻接,其中 $1\leq i\leq n,n$ 为通路长度。根据所采用的邻接定义不同,可定义或得到不同的通路,如4-通路、8-通路。这里对通路的定义仅仅考虑了像素坐标空间上的联系(相当于对邻接关系的推广),没有考虑像素属性空间上的联系。

上述通路建立了两个像素 p 和 q 之间的空间邻接联系。进一步,如果这条通路上的所有像素的灰度值均满足某个特定的相似准则,即两两邻接的像素也是两两连接的,则可以说像素 p 和 q 是连通的。同样根据所采用的连接定义的不同,可定义或得到不同的连通,如**4-连通,8-连通**。当 $n=1$ 时,连通转化为其特例——连接。

邻接、连接、通路、连通之间的关系可借助图 3.1.2 来直观的理解。首先从邻接出发,如果将两个像素的邻接推广到一系列两两邻接的像素就得到通路;如果将两个像素的空间相近扩展到属性也相似则得到连接关系。进一步,如果将两个像素的连接推广到一系列像素的两两连接就实现了连通;而如果将仅考虑空间相连的通路扩展到属性相似则也可实现连通。反过来,连接是仅仅两个像素之间的连通;而通路是不考虑属性的简化。最后,连接不考虑属性的简化就是邻接;而邻接是仅两个像素的通路。

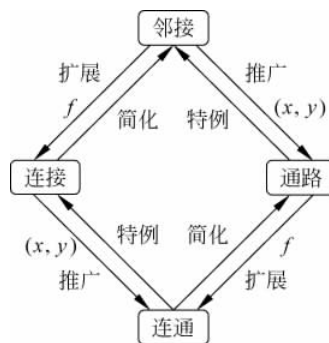


图 3.1.2 邻接、连接、通路、连通之间的关系

3. 像素集合的邻接、连接和连通

如果将一幅图像看作一个由像素构成的集合,则根据像素间的联系,常可将某些像素结合组成图像的子集合。换句话说,图像中的**子集**仍是像素的集合,是图像的一部分。对两个图像子集 S 和 T 来说,如果 S 中的一个或一些像素与 T 中的一个或一些像素邻接,则可以说两个图像子集 S 和 T 是邻接的。这里根据所采用的像素邻接定义,可以定义或得到不同的邻接图像子集。如可以说两个图像子集4-邻接,两个8-邻接的图像子集等。

类似于像素的连接,对两个图像子集 S 和 T 来说,要确定它们是否连接也需要考虑两点:①它们是否是邻接图像子集;②它们中邻接像素的灰度值是否满足某个特定的相似准则。换句话说,如果 S 中的一个或一些像素与 T 中的一个或一些像素连接,则可以说两个图像子集 S 和 T 是连接的。

设 p 和 q 是一个图像子集 S 中的两个像素,如果存在一条完全由在 S 中的像素组成的从 p 到 q 的通路,且其上像素灰度值满足相似准则,那么就称 p 在 S 中与 q 相连通。对 S 中任一个像素 p ,所有与 p 相连通且又在 S 中的像素组成的集合(包括 p)合起来称为 S 中的一个**连通组元**。如果 S 中只有一个连通组元,即 S 中所有像素都互相连通,则称 S 是一个连通集。如果一幅图像中所有的像素分属于几个连通集,则可以说这几个连通集分别是该幅图像的连通组元。在极端的情况下,一幅图像中所有的像素都互相连通,则该幅图像本身就是一个连通集。

一幅图像里每个连通集构成该图像的一个区域,所以图像可认为是由一系列区域组成的。如果一个区域中没有孔,称该区域是简单连通的,否则称有孔的区域是多重连通的。一个区域的边界也称区域的**轮廓**,一般认为轮廓是所在区域的一个子集,它将该区域与其他区域分离开。借助前面对像素邻域的介绍,可以认为组成一个区域的边界像素本身属于该区域而在其邻域中不属于该区域的像素(更多讨论见中册)。

3.1.3 像素间的距离

像素之间的联系常与像素在空间中的接近程度有关。像素在空间中的接近程度可以用像素之间的距离来测量。为测量**距离**,需要定义**距离量度函数**。给定 3 个像素 p, q, r , 坐标分别为 $(x, y), (s, t), (u, v)$, 如果满足下列条件, 称函数 D 为一个距离量度函数:

- (1) $D(p, q) \geq 0 (D(p, q) = 0, \text{当且仅当 } p = q)$;
- (2) $D(p, q) = D(q, p)$;
- (3) $D(p, q) \leq D(p, r) + D(r, q)$ 。

上述 3 个条件中,第 1 个条件表明两个像素之间的距离总是正的(两个像素空间位置相同时,其间的距离为零);第 2 个条件表明两个像素之间的距离与起终点的选择无关,或者说距离是相对的;第 3 个条件表明两个像素之间的最短距离是沿直线的。

例 3.1.1 测度空间

定义在抽象集合 A (其元素 $a_1, a_2, a_3, \dots$ 可称为点) 上的测度是一个从集合向实数集 $\mathbf{R}$ 映射的函数(可记为 $d: A \times A \rightarrow \mathbf{R}$), 对任意 3 个 $a_1, a_2, a_3 \in A$, 都有

- (1) $d(a_1, a_2) = 0$, 当且仅当 $a_1 = a_2$;
- (2) $d(a_1, a_2) \leq d(a_3, a_1) + d(a_3, a_2)$;
- (3) $d(a_1, a_2) = d(a_2, a_1)$;
- (4) $d(a_1, a_2) > 0$, 如果 $a_1 \neq a_2$ 。

二元组 (A, d) 称为测度空间。 □

在图像中,对距离有不同的量度方法。点 p 和 q 之间的**欧氏距离**(也是范数为 2 的距离)定义为

$$D_E(p, q) = [(x-s)^2 + (y-t)^2]^{1/2} \quad (3.1.1)$$

根据这个距离量度,与坐标为 (x, y) 的像素的 D_E 距离小于或等于某个值 d 的像素都包括在以 (x, y) 为中心、以 d 为半径的圆中。在数字图像中,只能近似地表示一个圆,例如与 (x, y) 的 D_E 距离小于或等于 3 的像素组成如图 3.1.3(a) 所示的多层嵌套的等距离轮廓(图中距离值已四舍五入到保留一位小数)。

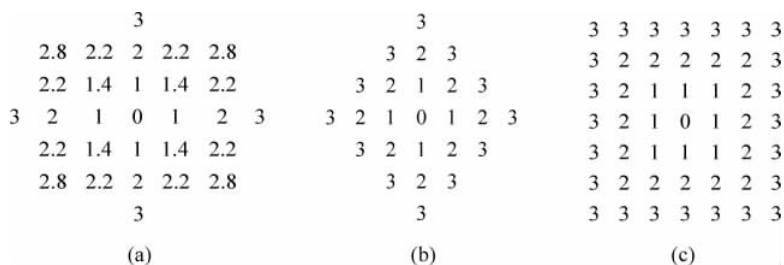


图 3.1.3 等距离轮廓示例

点 p 和 q 之间的 D_4 距离(也是范数为 1 的距离)也称为**城区距离**, 定义为

$$D_4(p, q) = |x - s| + |y - t| \quad (3.1.2)$$

根据这个距离量度, 与坐标为 (x, y) 的像素的 D_4 距离小于或等于某个值 d 的像素组成以 (x, y) 为中心的菱形。例如与 (x, y) 的 D_4 距离小于或等于 3 的像素组成如图 3.1.3(b) 所示的菱形区域。 $D_4=1$ 的像素构成像素 p 的 4-邻域。换句话说, 像素 p 的 4-邻域也可用 D_4 距离定义为

$$N_4(p) = \{r | D_4(p, r) = 1\} \quad (3.1.3)$$

点 p 和 q 之间的 D_8 距离(也是范数为 ∞ 的距离)也称为**棋盘距离**, 定义为

$$D_8(p, q) = \max(|x - s|, |y - t|) \quad (3.1.4)$$

根据这个距离量度, 与坐标为 (x, y) 的像素的 D_8 距离小于或等于某个值 d 的像素组成以 (x, y) 为中心的正方形。例如, 与 (x, y) 的 D_8 距离小于或等于 3 的像素组成如图 3.1.3(c) 所示的方形区域。 $D_8=1$ 的像素构成像素 p 的 8-邻域。这样, 像素 p 的 8-邻域也可用 D_8 距离定义为

$$N_8(p) = \{r | D_8(p, r) = 1\} \quad (3.1.5)$$

例 3.1.2 距离定义和计算示例

根据上述 3 种距离定义, 在计算图像中相同两个像素间的距离时会得到不同的数值。如在图 3.1.4 中, 两个像素 p 和 q 之间的 D_E 距离为 5(见图(a)), D_4 距离为 7(见图(b)), D_8 距离为 4(见图(c))。图 3.1.3(d) 将 3 种距离画在同一图上, 更容易看出它们的区别。

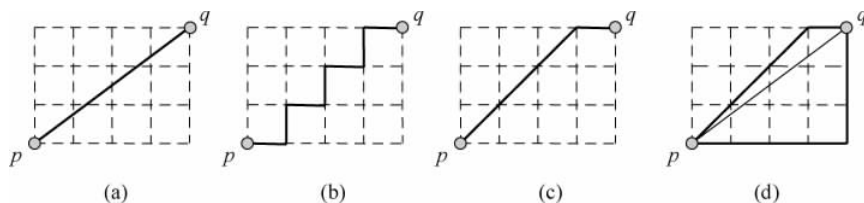


图 3.1.4 像素间距离定义示例

□

欧氏距离给出的结果应该最准确, 但由于计算时需要进行平方和开方运算, 计算量大。城区距离和棋盘距离均为非欧氏距离, 计算量小, 但有一定的误差。这种误差在两个像素处于对角方向时达到最大。如果用 N 表示两个像素间的水平距离(也等于垂直距离), 则城区距离的误差为 $|(\sqrt{2} - 2)N| = 0.59N$; 棋盘距离的误差为 $|(\sqrt{2} - 1)N| = 0.41N$ 。

例 3.1.3 范数和距离

范数是测度空间的一个基本概念。一个函数 $f(x)$ 的范数可表示为(其中 m 称为指数或指标)

$$\|f\|_m = \left[\int |f(x)|^m dx \right]^{1/m} \quad (3.1.6)$$

在距离计算中, 可定义两点之间的 Minkowski 距离度量为

$$D_m(p, q) = [|x - s|^m + |y - t|^m]^{1/m} \quad (3.1.7)$$

式中, m 取 1、2 和 ∞ 是几种常用的特殊情况。参见图 3.1.5, 考虑与原点为单位距离的点的集合, 当 m 取 1 时, 得到一个菱形; 当 m 取 2 时, 得到一个圆形; 当 m 取 ∞ 时, 得到一个正方形。可将图 3.1.5 与前面的图 3.1.3 进行对照。

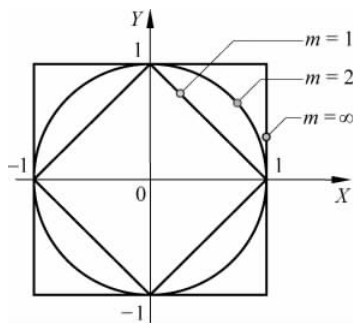


图 3.1.5 3 种范数和 3 种距离

□

3.2 模板运算

模板也称样板或窗,一般可看作一幅尺寸为 $n \times n$ (n 一般为奇数,远小于常见图像尺寸)的小图像 $W(x, y)$,其各个位置上的值常称为系数值,系数值由功能确定。根据像素间的联系可以定义各种模板操作并实现各种功能。模板运算的基本思路是将赋予某个像素的值作为它本身灰度值和其相邻像素灰度值的函数。函数的形式可线性也可非线性,运算可以是卷积也可以是排序等。利用像素本身以及其邻域像素的灰度关系进行增强的方法常称为**滤波**,而实现其功能的模板就相当于滤波器。

下面先介绍两种**模板运算**(模板卷积和模板排序),再讨论在图像边界处进行模板操作要注意的问题,最后对利用模板操作可实现的图像空域增强技术进行分类。

1. 模板卷积

模板卷积指用模板与需处理图像在图像空间进行卷积的运算过程。该过程不能原地完成(这与点操作不同),所以输出结果要使用另一幅图像。模板卷积的主要步骤为

- (1) 将模板在输入图像中漫游,并将模板中心与图像中某个像素位置重合;
- (2) 将模板上的各个系数与模板下各对应像素的灰度值相乘;
- (3) 将所有乘积相加(为保持灰度范围,常将结果再除以模板系数之和);
- (4) 将上述运算结果(模板的输出响应)赋给输出图像中对应模板中心位置的像素。

上述过程完成了利用输入图像中与模板同尺寸的图像子集给出输出图像中一个像素幅度值的工作。要对一幅图像卷积,需要对输出图像中的每个像素进行上述计算。一般的模板是方形的,最常用的尺寸为 3×3 ,有些时候也使用 5×5 、 7×7 或更大的模板。实用中 n 多为奇数以使模板对称并有一个中心像素,可以定义模板的半径 r 为 $(n-1)/2$ 。

图 3.2.1(a)给出一幅图像的一部分,其中所标 s_i 为代表像素的灰度值。现设有一个 3×3 的模板如图 3.2.1(b)所示,模板内所标为模板系数。如将 k_0 所在位置与图中灰度值为 s_0 的像素重合(即将模板中心放在图中 (x, y) 位置),模板的输出响应 R 为

$$R = k_0 s_0 + k_1 s_1 + \cdots + k_8 s_8 \quad (3.2.1)$$

将 R (实际中常常除以模板系数之和以保证原来的灰度动态范围)赋给输出图像在 (x, y) 位置的像素作为其新的灰度值(见图 3.2.1(c)),就完成了在该像素的卷积操作。

2. 模板排序

模板排序也是一种模板运算。模板排序是指用模板来提取需处理图像中与模板同尺寸

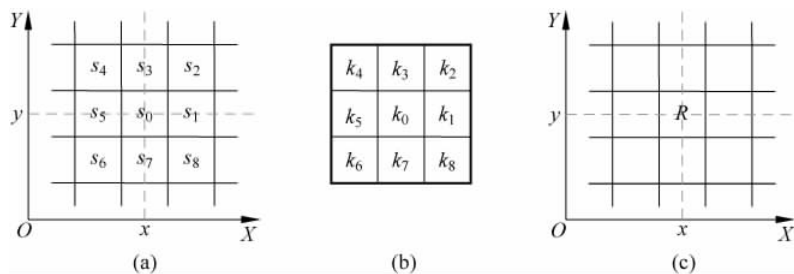


图 3.2.1 用 3×3 的模板进行模板操作的示意图

的图像子集并将其中像素根据其幅度值排序的运算过程。与模板卷积类似,模板排序过程也不能原地完成。模板排序的主要步骤为

- (1) 将模板在输入图像中漫游,并将模板中心与图像中某个像素位置重合;
- (2) 读取模板下输入图像中各对应像素的灰度值;
- (3) 将这些灰度值进行排序,一般将它们从小到大排成一列(单增);
- (4) 根据运算目的从排序结果中选一个序,取出该序像素的灰度值;
- (5) 将取出的灰度值赋给输出图像中对应模板中心位置的像素。

与模板卷积不同,模板排序中的模板只起到划定参与图像处理的像素范围的作用,其系数在读取像素灰度值时可看作均为 1,且不影响赋值。模板排序后如何取其中一个灰度值是区分其功能的重要因素。另外,模板排序后所赋给输出图像中对应模板中心位置像素的值必是输入图像中与模板对应的像素值中的一个。最后,模板排序中利用模板只是为了选取一些像素,所用的模板并不一定是方形的,或者虽然用方形的模板,但其中有些系数取 0 (见 3.4.1 小节)。

3. 图像边界处的模板运算

由于在模板运算中要用到输入图像中与模板中心的邻域所对应的像素,当模板中心对应输入图像的边界像素时,其邻域范围可能扩展到输入图像的边界之外,而那里并没有定义。解决这个问题的思路有两种。一种是忽略这些边界处的像素,仅考虑图像内部与边界距离小于等于模板半径的像素。当图像尺寸比较大且感兴趣目标在图像内部时这种方法的效果常可以接受。另一种是将输入图像进行扩展,即如果用半径为 r 的模板进行模板运算,则在图像的四条边界外各增加/扩展一个 r 行或 r 列的带(先在图像的第一行之前(上)和最后一行之后(下)各增加 r 行,再在图像的第一列左边和最后一列右边都各增加 r 列,这里操作可按行或列来迭代进行),从而可以正常地实现对边界上像素的运算。这些新增行或列中像素的幅度值可用不同的方法来确定,例如:

- (1) 最简单的是将新增像素的幅度值取为 0,缺点是有可能导致图像边界处有明显的 discontinuity;
- (2) 将这些新增像素的幅度值取为其在原图像中 4-邻接像素的值(4 个角上新增像素的幅度值取为其在原图像中 8-邻接像素的值);
- (3) 将图像在水平和垂直方向上均看作是周期循环的,即认为图像最后一行之后是图像的第一行,图像最后一列之后是图像的第一列,从而将相应的行或列移过来;
- (4) 利用外插技术,根据接近边界处一行或多行(一列或多列)像素的幅度值以一定的规则进行外推得到图像边界外像素的幅度值。

需要指出上述这些方法都不是完美/理想的,因为实际上它们都是对图像边界外像素幅度值的一种估计而已。事实上,它们均对边界像素给予了特殊的权重,并会使图像的平均灰度发生小的改变。

4. 模板运算功能分类

利用模板运算也可对图像进行空域增强。如以 $f(x,y)$ 和 $g(x,y)$ 分别代表原始图像和增强图像,用 E_H 代表一个增强操作,则有

$$g(x,y) = E_H[f(x,y),N(x,y)] \tag{3.2.2}$$

式中, $N(x,y)$ 代表 $f(x,y)$ 在 (x,y) 的邻域内各像素的灰度值。

将模板运算用于图像空域增强一般称为空域滤波,根据其功能主要分成平滑滤波和锐化滤波两类(模板系数不同)。

(1) 平滑滤波

它能减弱或消除图像中的高频率分量,但不影响低频率分量。因为高频分量对应图像中的区域边缘等灰度值具有较大较快变化的部分,平滑滤波将这些分量滤去可减少局部灰度起伏,使图像变得比较平滑。实际中,平滑滤波还可用于消除噪声(噪声的空间相关性较弱,对应较高的空间频率),或在提取较大的目标前去除太小的细节或将目标内的小间断连接起来。

(2) 锐化滤波

它能减弱或消除图像中的低频率分量,但不影响高频率分量。因为低频分量对应图像中灰度值缓慢变化的区域,因而与图像的整体特性,如整体对比度和平均灰度值等有关。锐化滤波将这些分量滤去可使图像反差增加,边缘明显。实际中,锐化滤波可用于增强被模糊的细节或目标的边缘。

另一方面,空域滤波也常根据其运算特点分成线性的和非线性的两类。从统计的角度看,滤波是一种估计,它基于一组观察结果来估计未观察的量。线性滤波对观察结果进行线性组合,而非线性滤波则是对观察结果的逻辑组合[Dougherty 1994]。线性方法的理论基础比较成熟。在线性的方法中,常可将复杂的运算进行分解,计算比较方便,也容易并行实现。非线性的方法理论基础较弱,应用领域受一些限制,但有些非线性方法常比线性方法有更好的滤波效果。

结合上述两种分类方法,可将空间滤波增强技术分成 4 类,见表 3.2.1。

表 3.2.1 空域滤波增强技术分类

功能 \ 特点	线 性	非 线 性
	特 点	特 点
平滑	线性平滑	非线性平滑
锐化	线性锐化	非线性锐化

下面两节将分别介绍线性滤波和非线性滤波。

3.3 线 性 滤 波

线性滤波既可得到平滑的效果(图像反差减少),也可得到锐化的效果(图像反差增加),主要取决于所用模板的系数值。线性滤波均基于模板卷积进行。

3.3.1 线性平滑滤波

有很多种线性平滑滤波的方法。此时,平滑模板系数的取值均应为正,而且可以在中心比较大而周围比较小。

需要指出,很多时候可将图像看作一个随机过程(随机场)。如果它是遍历的,则可用空间平均(模板计算)替换时间平均(见 2.2.2 小节)。实际中图像并不是完全遍历的,只在其中亮度一致的区域是遍历的。所以,与时间平均法相比,空间平均法常会使图像中的边缘变得模糊。

下面介绍几种比较简单和典型的线性平滑滤波方法。

1. 邻域平均

最简单的平滑滤波是用一个像素邻域的平均值作为滤波结果,此时滤波模板的所有系数都取为 1。为保证输出图仍在原来的灰度值范围,在算得卷积值 R 后要将其除以系数总个数再行赋值。例如对 3×3 的模板来说,在算得 R 后要将其除以系数 9。邻域平均的一般表达式为

$$g(x, y) = \frac{1}{n^2} \sum_{(s, t) \in N(x, y)} f(s, t) \quad (3.3.1)$$

其中, $N(x, y)$ 对应 $f(x, y)$ 中 (x, y) 的 $n \times n$ 邻域,与模板 W 所覆盖的范围对应。

例 3.3.1 邻域平均平滑滤波的效果

参见图 3.3.1,其中图(a)为一幅原始的 8 比特灰度级图像,图(b)为叠加了均匀分布随机噪声的结果,图(c)~图(g)依次为用 3×3 , 5×5 , 7×7 , 9×9 和 11×11 的平滑模板对图(b)进行平滑滤波的结果。由这些图可见当所用平滑模板尺寸增大时,对噪声的消除效果有所增强。不过同时所得到的图像变得更为模糊,可视的细节逐步减少,且所需运算量也逐步增大。所以实际中需根据应用要求选取合适大小的模板。

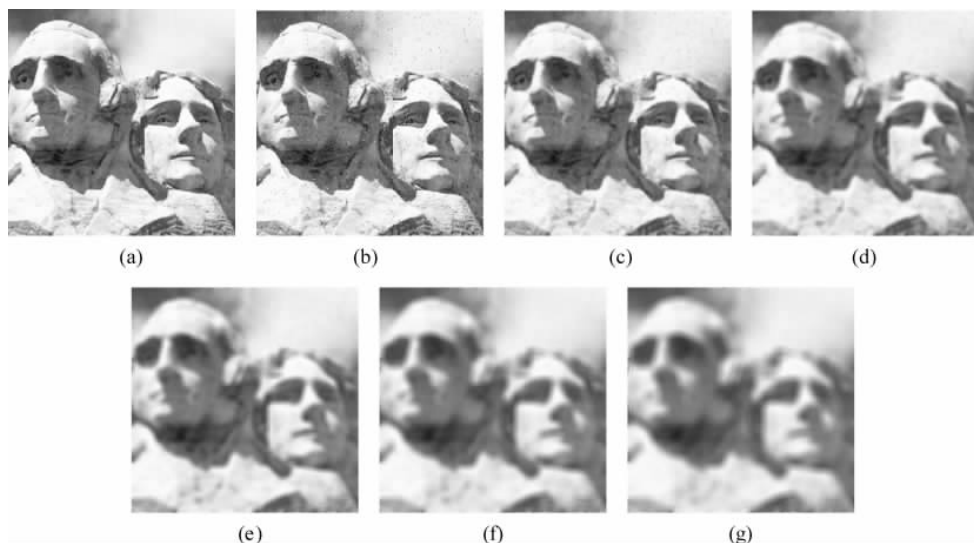


图 3.3.1 空域平滑滤波的效果

□

如果图像的尺寸是 $W \times H$, 模板的尺寸是 $n \times n$, 则均值滤波器的计算量为 $O(WHn^2)$ 。如果使用加法结合律, 先计算一个方向的和(并保存)再计算另一个方向的和, 就可以将计算量降到 $O(WH(n+n)=2WHn)$ 。如果使用递归的方式, 还可以降到 $O(WH)$, 此时计算量并不依赖于模板尺寸。

2. 加权平均

模板操作中, 模板中心周围的像素也参与滤波。一般认为离中心近的像素应对滤波结果有较大的贡献(它们与中心像素的相关性也大), 所以可将接近模板中心的系数取得比模板周边的系数大, 这相当于对邻域平均进行了加权。加权平均的一般表达式为

$$g(x, y) = \frac{\sum_{(s,t) \in N(x,y)} w(s, t) f(s, t)}{\sum_{(s,t) \in N(x,y)} w(s, t)} \quad (3.3.2)$$

实用中, 为保证各模板系数均为整数以减少计算量, 常取模板周边最小的系数为 1, 而取模板内部的系数成比例增加, 直到中心系数取得最大值。

这里的增加比例可根据各系数位置与模板中心的距离来确定, 例如依次根据距离的倒数来确定各内部系数的值。图 3.3.2 给出这样得到的一个模板的示例。

1	2	1
2	4	2
1	2	1

图 3.3.2 一个加权平均模板

在邻域平均中, 可通过选取不同尺寸的模板获得不同的结果。在加权平均中, 除对同一尺寸模板中的不同位置采用不同系数外, 还可选取不同尺寸的模板。最后, 如果将小尺寸的模板(迭代)反复使用, 也可得到加权大尺寸模板的效果(见思考题和练习题 3-4)。

3. 高斯平均

高斯平均是加权平均的一种特例, 它根据高斯分布来确定各模板系数, 也将接近模板中心的系数取得比模板周边的系数大。例如一个 5×5 的高斯平均模板如下:

$$\frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

可以严格证明, 一个 2-D 的高斯卷积可分解(分离)为顺序执行的两个 1-D 高斯卷积, 即一个 2-D 高斯平均模板可拆分成两个 1-D 高斯平均模板。

例 3.3.2 高斯平均模板的分解

高斯平均使用其系数近似高斯分布的模板。实际中, 为较好地近似高斯分布, 高斯平均常使用较大的模板, 因而有可能需要很大的计算量。例如, 使用 30×30 的高斯卷积模板对 256×256 的图像进行平滑需要 64×10^6 次基本操作。为此, 可将 2-D 高斯平均模板分解为两个顺序使用的 1-D 高斯平均模板, 例如,

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \quad \square$$

一般情况下, 计算量为 $O(n^2)$ 的单个 $n \times n$ 模板被计算量为 $O(n)$ 的两个 $1 \times n$ 模板所替

换,节约因子是 $n/2$,所以对 $n>3$,总可以借助模板分解减少计算量。表 3.3.1 给出对几个小模板计算的情况,最后一列给出计算量减少的效果(与 n 成正比)。

表 3.3.1 将 $n \times n$ 高斯模板分解为两个 $1 \times n$ 模板可减少计算量

n	n^2	$2n$	$2n/n^2 = 2/n$
1	1	2	2
3	9	6	2/3
5	25	10	2/5
7	49	14	2/7
9	81	18	2/9

为了得到 1-D 的离散模板,可对高斯函数在整数位置 $-n, \dots, 0, \dots, +n$ 采样。具体可取 $n=2\sigma+1$ (σ 为高斯方差),而模板尺寸是 $S=2n+1$ 。例如,对 $\sigma=1.0$,模板尺寸最多为 7 就足够了;如果 $\sigma=2.0$,则模板尺寸需要至少为 11。

对高斯函数的简单近似可借助杨辉三角形进行。表 3.3.2 给出所得到的几个小模板的系数。

表 3.3.2 1-D 高斯模板系数

$g(i)$	σ	$g(i)$	σ
1	0	1 3 3 1	3/4
1 1	1/4	1 4 6 4 1	1
1 2 1	1/2	1 5 10 10 5 1	5/4

高斯滤波器是唯一满足以下各个条件的滤波器,即线性的,与位置无关的,旋转对称的(各向同性的),可以控制效果的(有一个控制参数),可以连续多次使用的,可分离的(2-D 可用 1-D 的来计算)。但高斯滤波器不能以递归的方式来执行。

线性滤波器都会改变图像中噪声的方差。一个线性滤波器 $h(x,y)$ 会使噪声方差乘以因子 $\sum \sum h(x,y)$ 。对高斯滤波器,这个因子是 $1/(4\pi\sigma^2)$ 。所以,要让一个高斯平均滤波器与一个 $n \times n$ 的邻域平均滤波器具有相同的效果,需取 $\sigma=n/(2\sqrt{\pi})$ 。例如,一个 5×5 的邻域平均滤波器与一个 $\sigma \approx 1.41$ 的高斯滤波器具有相同的滤波性能。

4. 边缘保持平滑

前面介绍的几种平均方法都是线性的,在消除噪声的同时也会模糊不同区域之间的边缘。为了避免局部平均的这种副作用,可以考虑选择只在不包含边缘的邻域中进行平均。

参见图 3.3.3,考虑一个像素(用 ● 表示)的 5×5 邻域,分别取该中心像素的 8 个不同的邻域。这 8 个邻域每个都包含 7 个像素(用阴影表示),但分成两种类型:近似五边形和近似六边形(用阴影上的粗线表示)。同类型的模板可通过绕中心像素旋转得到。对每个邻域都统计其像素的灰度方差,并将灰度方差最小的那个邻域中的像素均值赋给中心像素。

这里的基本思路是围绕要平滑的像素找到它周围不包含边缘的邻域(方差小的邻域中像素灰度级的变化小,包含边缘的概率也小),用这样的邻域进行平均可以避免模糊边缘或破坏区域边界的形状。如果使用具有较大邻域的模板就能进一步减少噪声的影响,但也有可能平滑掉尺寸较小的区域和损伤目标轮廓的细节。所以,一般是迭代地使用较小的模板,

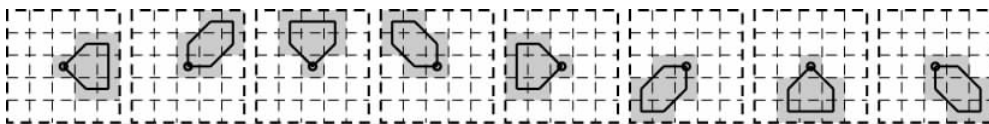


图 3.3.3 八个边缘保持平滑模板

直到像素值基本不再变化。

3.3.2 线性锐化滤波

有很多种线性锐化滤波的方法,下面介绍几种比较简单和典型的线性锐化滤波方法。

1. 拉普拉斯算子

线性锐化滤波可借助模板卷积实现。对应积分运算的模板卷积可以平滑图像,反过来对应微分运算的模板卷积可以锐化图像。锐化模板系数的取值应在中心为正而周围远离中心处为负。

拉普拉斯算子是一种各向同性的二阶微分算子,常用于线性锐化滤波。根据定义,

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.3.3)$$

两个分别沿 X 和 Y 方向的二阶偏导数均可借助差分来计算

$$\frac{\partial^2 f}{\partial x^2} = 2f(x, y) - f(x+1, y) - f(x-1, y) \quad (3.3.4)$$

$$\frac{\partial^2 f}{\partial y^2} = 2f(x, y) - f(x, y+1) - f(x, y-1) \quad (3.3.5)$$

将式(3.3.4)和式(3.3.5)代入式(3.3.3)得到

$$\nabla^2 f(x, y) = 4f(x, y) - f(x+1, y) - f(x-1, y) - f(x, y+1) - f(x, y-1) \quad (3.3.6)$$

据此得到的模板如图 3.3.4(a)所示,仅考虑了中心像素的 4-邻域。类似地,如果考虑 8-邻域,则得到如图 3.3.4(b)所示的模板,这相当于在式(3.2.1)中取 $k_0=8$,而取其余系数为-1。

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

(a)

-1	-1	-1
-1	8	-1
-1	-1	-1

(b)

图 3.3.4 两种拉普拉斯算子模板

以上两种模板的所有系数之和均为 0,这是为了使经过模板运算所得结果图像的均值不变。当这样的模板放在图像中灰度值是常数或变化很小的区域时,其卷积输出为 0 或很小。

使用这样的模板会将输出图的平均灰度值变为 0,这样图中就会有一部分像素灰度值小于 0。在图像处理中,一般只考虑大于或等于 0 的灰度值,所以还需将输出图灰度值范围通过变换变回到 $[0, L-1]$ 区间才能正确显示出来。

拉普拉斯算子增强了图像中的灰度不连续边缘,而减弱了对应图像中灰度值缓慢变化区域的对比度,将这样的结果叠加到原始图像上,就可以得到锐化后的图像。

2. 高频提升滤波

图像锐化的效果可以通过叠加图像微分结果取得,也可以通过减除图像积分结果取得。设原始图像为 $f(x, y)$,其平滑后的图像为 $g(x, y)$,则用原始图像减去平滑或模糊图像就得到非锐化掩模,如将非锐化掩模加到原始图像上就能锐化图像。更进一步,如果把原始图像

乘以一个放大系数 A 再减去平滑图像就可实现**高频提升滤波**：

$$h_b(x,y) = Af(x,y) - g(x,y) = (A-1)f(x,y) + h(x,y) \quad (3.3.7)$$

式中,当 $A=1$,得到的就是非锐化掩模 $h(x,y)$,它对应平滑时丢失的锐化分量。当 $A>1$ 时,加权的原始图像与非锐化掩模相加,能使原始图像得到锐化的效果,其中 $A=2$ 的特例被称为**非锐化掩模化**。可见非锐化掩模化包括 3 个步骤：①平滑原始图像；②从原始图像中减去平滑后的结果；③将上述结果加到原始图像上。

如果要用如图 3.3.4 所示的 3×3 模板来实现高频提升滤波,则对图 3.3.4(a),其中心系数取值应是 $k_0>4$ ；而对图 3.3.4(b),其中心系数取值应是 $k_0>8$ 。

例 3.3.3 拉普拉斯算子与高频提升滤波的比较

图 3.3.5 给出拉普拉斯算子与高频提升滤波的一个比较示例。图(a)为一幅实验的原始图像(对 Lena 图像模糊而得到),图(b)为对其用拉普拉斯算子进行处理得到的结果(已进行了尺度变换以显示),图(c)为采用高频提升滤波得到的结果($A=2$),图(d)为在此基础上又用直接灰度变换方法对灰度值范围进行扩展得到的最终结果。



图 3.3.5 拉普拉斯算子与高频提升滤波的比较

□

3.4 非线性滤波

虽然线性滤波计算简单,但它常不能区分图像中有用的内容和无用的噪声。参见如图 3.4.1 所示的 1-D 傅里叶空间示意图(横轴对应频率,竖轴对应频率系数值),如果对原始图像加上白噪声,两者的频谱将直接叠加,在有噪声图像的各个频率处,其幅度既包括原始图像的贡献也包括噪声的贡献。任何线性滤波都可描述成将原始图像的傅里叶变换和滤波模板的傅里叶变换相乘,结果是在每个频率处有用内容的强度和无用噪声的水平都以相同的因子减弱。这样,图像信噪比将保持原值,并不增加。

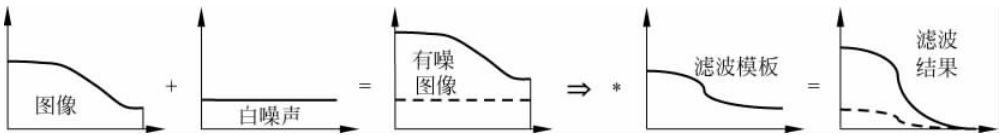


图 3.4.1 线性滤波不能区分图像中有用的特征和噪声

上述问题常可借助非线性滤波来解决。**非线性滤波**主要沿 3 个方向发展：逻辑、几何、代数[Dougherty 1994]。它们也可分别定义为基于集合的、基于形状的、基于排序的 3 种。这 3 种非线性滤波是密切相关的。其中,基于形状的非线性滤波是围绕数学形态学(见中册

第 13 章和第 14 章)进行的,而基于形状的形态操作建立在集合论的基础上,并可在一定条件下简化成传统的逻辑表达。下面仅介绍基于排序的非线性滤波。

3.4.1 非线性平滑滤波

基于排序的非线性滤波中,中值滤波是个典型,已被成功地用于保留所需的图像结构并同时消除(脉冲)噪声。

1. 中值滤波原理

中值滤波依靠模板排序来实现。为简便起见,先考虑用 1-D 信号来介绍其原理。设模板尺寸为 M , $M=2r+1$, r 为模板半径,给定 1-D 信号序列 $\{f_i\}$, $i=1,2,\dots,N$,则中值滤波输出为

$$g_j = \text{median}[f_{j-r}, f_{j-r+1}, \dots, f_j, \dots, f_{j+r}] \quad (3.4.1)$$

式中,median 代表取中值,即对模板覆盖的信号序列按数值大小进行排序,并取排序后处在中间位置的值,且有 $1 < j-r < j+r < N$ 。换句话说, $\{f_i\}$ 中有一半值大于 g_j ,而另一半值小于 g_j 。上式定义的操作常称为游程中值,可以通过滑动奇数长度的模板来实现。如果记模板中的采样分别为 $f_1, f_2, \dots, f_{2r+1}$,排序后的采样依次为 $f_{(1)}, f_{(2)}, \dots, f_{(2r+1)}$ 。将 $f_{(i)}$ 记为第 i 阶统计,那么中值就是 $r+1$ 阶的统计值[Dougherty 1994]。

中值滤波适合用于对脉冲噪声的消除。脉冲噪声会导致受影响的像素值发生明显变化,成为野点。平均滤波不加区别地平均对应模板的所有像素值,而中值滤波则忽略野点,所以不会模糊图像。为区别中值滤波和平均滤波,可以考虑它们的零脉冲响应和理想阶跃响应。中值滤波可以完全消除孤立的脉冲而不对通过的理想边缘产生任何影响。图 3.4.2 给出一对示例,上面是原始信号序列,下面是中值滤波结果,其中窗口长度为 3。图(a)表示消除孤立的脉冲而不对边缘产生影响,图(b)表示接近边缘的脉冲会使边缘偏移。

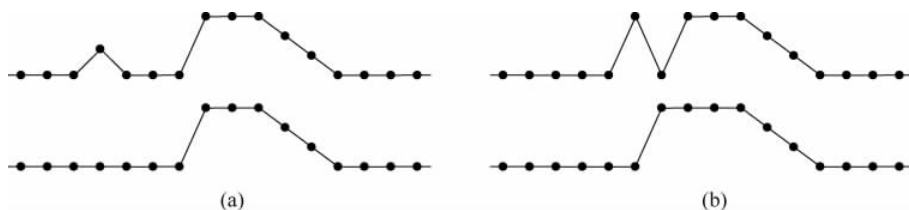


图 3.4.2 1-D 中值滤波示例

能被中值滤波完全除去的脉冲的最大长度依赖于模板尺寸 $S=2r+1$ 。考虑一个长度记为 L 的信号 $f(i)$

$$f(i) \begin{cases} = 0 & i < 0 \\ \neq 0 & i = 0 \\ \neq 0 & i = L-1 \\ = 0 & i \geq L \end{cases} \quad (3.4.2)$$

容易看出,如果 $L \leq r$ 那么输出将完全是 0,即脉冲全被消除了。然而,如果信号仅包含长度至少为 $r+1$ 的常数段,那么用长度小于等于 $2r+1$ 的中值滤波模板对信号滤波并不会使信号发生任何变化。不受中值滤波影响的信号称为根信号。一个信号是一个长度为 $2r+1$ 的中值滤波的根信号的充分条件是该信号局部单调变化且阶数为 $r+1$,即该信号的每个长

度为 $r+1$ 的段均为单调的。

例 3.4.1 中值滤波的一些特性示例

考虑对图像中的一个 1-D 剖面分别使用 3 个元素的中值滤波和 5 个元素的中值滤波得到(? 代表未定):

原始: 1 2 3 0 2 2 3 1 1 2 2 **9** 2 **2** 8 8 8 7

3 元素中值: ? 2 2 2 2 2 2 1 1 2 2 2 2 **2** 8 8 8 ?

5 元素中值: ? ? 2 2 2 2 2 2 2 2 2 2 **2** 8 8 8 ? ?

由此例可见:

(1) 中值滤波可很好地消除孤立尖峰(原始的 9 没有在对应结果中留下痕迹);

(2) 中值滤波有可能导致尖峰附近的边缘偏移,且元素多的更易偏移(原始 2 和 8 之间的边缘在 5 元素中值输出中更接近为 9 的尖峰,该边缘与尖峰不在同一个 3 元素中值的模板中);

(3) 中值滤波趋向于产生为常数的片断(5 元素中值输出连续的 2)。

□

例 3.4.2 一个消除二值图像中椒盐噪声的简单算法

设原始受椒盐噪声影响的二值图像为 $f(x, y)$, 噪声消除后的图像为 $g(x, y)$, 考虑一个像素的 8-邻域 $N(x, y)$, 则一个消除二值图像中椒盐噪声的简单算法如下:

(1) 计算: $s = \sum_{\substack{(p,q) \in N(x,y) \\ (p,q) \neq (x,y)}} f(p, q)$

(2) 判断: 如果 $s=0, g(x, y)=0$; 如果 $s=8, g(x, y)=1$; 否则 $g(x, y)=f(x, y)$ 。

上述算法可以调整,以推广到消除二值图像中目标区域边界上的毛刺:

(1) 计算: $s = \sum_{\substack{(p,q) \in N(x,y) \\ (p,q) \neq (x,y)}} f(p, q)$

(2) 判断: 如果 $s \leq 1, g(x, y)=0$; 如果 $s \geq 7, g(x, y)=1$; 否则 $g(x, y)=f(x, y)$ 。

□

2. 2-D 中值滤波

一个用于图像的 2-D 中值滤波的输出可写为

$$g_{\text{median}}(x, y) = \text{median}_{(s,t) \in N(x,y)} [f(s, t)] \quad (3.4.3)$$

对一个所用尺寸为 $n \times n$ 的中值滤波模板,其输出值应大于等于模板中 $(n^2-1)/2$ 个像素的值,又应小于等于模板中 $(n^2-1)/2$ 个像素的值。例如使用一个 3×3 的模板,中值是第 5 大的那个。一般情况下,图像中尺寸小于模板尺寸一半的过亮或过暗区域将会在滤波后被消除掉。所以中值滤波的主要功能就是让与周围像素灰度值的差比较大的像素改取与周围像素值接近的值,这样它对孤立的噪声像素的消除能力是很强的。又由于它不是简单的取均值,所以产生的模糊比较少。换句话说,中值滤波既能消除噪声又能较好地保持图像的细节。

例 3.4.3 邻域平均和中值滤波的效果比较

图 3.4.3 对比给出对同一幅图像分别用邻域平均和中值滤波处理的结果。仍考虑图 3.3.1(b)中叠加了均匀分布随机噪声的图像。图 3.4.3(a)和(c)分别给出用 3×3 和 5×5 模板进行邻域平均处理得到的结果,而图 3.4.3(b)和(d)分别为用 3×3 和 5×5 模板进行中值滤波处理得到的结果。两相比较可见中值滤波的效果要比邻域平均处理的低通滤波效

果好,主要特点是滤波后图像中的轮廓比较清晰。

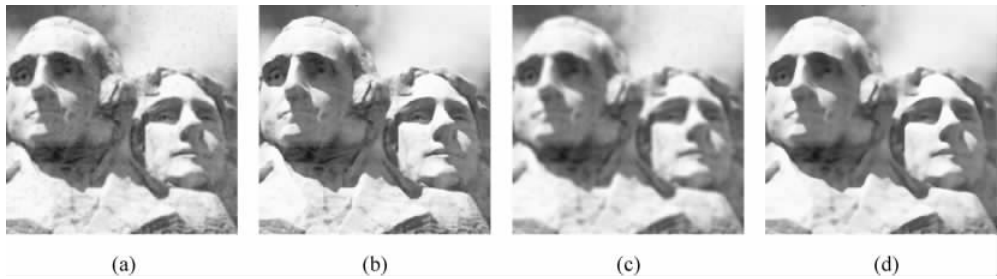


图 3.4.3 邻域平均和中值滤波的比较

□

3. 中值滤波的模板

中值滤波的效果不仅与所用模板的尺寸有关,也与模板中参与(排序)运算的像素个数有关。当使用给定尺寸的模板时,可以仅利用其中的一部分来计算以减少计算量。图 3.4.4 给出一些模板尺寸为 5×5 时的例子。在图 3.4.4 中,图 3.4.4(a)~图 3.4.4(c)都只使用了 9 个像素,其中图 3.4.4(a)和图 3.4.4(b)可看作通过分别延伸 4-邻域和对角邻域而得到,而图 3.4.4(c)则考虑了 16-邻域(见[章 2005b])。图 3.4.4(d)~图 3.4.4(f)均使用了 13 个像素,其中图 3.4.4(d)是将图 3.4.4(c)与 4-邻域像素结合,图 3.4.4(e)使用了与中心像素的 D_4 距离小于或等于 2 的像素,图 3.4.4(f)则使用了与中心像素的欧氏距离为 $2 \sim 2.5$ 的像素。有实验表明,当使用超过 $9 \sim 13$ 个像素的模板来消除图像中的噪声时,计算量的增加比消噪效果的改善更明显,所以常可使用稀疏的模板来减少运算量。

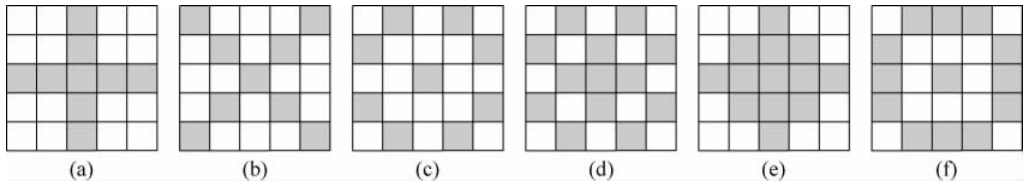


图 3.4.4 一些用于中值滤波的模板

另外,中值滤波的效果还与所用模板的形状或模板中参与运算的像素所构成图案的形状有关[Dougherty 1994]。一般来说方形的模板对图像的细节最不敏感:它会滤除细线并消除边缘上的角点。它也常产生讨厌的条纹(常数灰度的区域),这是中值滤波的一个缺点。十字交叉模板能保留较细的水平线和垂直线,但有可能将对角线滤除掉。X 形状的模板仅保留对角线。使用十字交叉模板得到的效果看起来比较好,因为水平线和垂直线都在人类视觉中起着重要的作用。

4. 2-D 中值计算

为提高计算速度,实际中常试图顺序使用两个 1-D 中值滤波模板以实现类似于 2-D 中值滤波模板的效果,但它们并不严格相等(因为中值计算是非线性的)。由于使用两个 1-D 中值滤波模板有一个次序问题,所以计算 2-D 中值的方法有 3 种:①直接使用 2-D 模板;②先使用第 1 个 1-D 模板再使用第 2 个 1-D 模板;③先使用第 2 个 1-D 模板再使用第 1 个 1-D 模板。这 3 种方法的结果可能相同,也可能不相同。

借助图 3.4.5 给出一个上述 3 种方法的最终结果都不相同的简单例子,其中图(a)代表

一幅原始图像；图(b)是直接使用 3×3 的 2-D 中值滤波模板得到的结果；图(c)和图(d)是先使用 1×3 的 1-D 中值滤波模板和再使用 3×1 的 1-D 中值滤波模板所得到的结果；图(e)和图(f)是先使用 3×1 的 1-D 中值滤波模板和再使用 1×3 的 1-D 中值滤波模板所得到的结果。比较图(b)、图(d)和图(f)，可见最终结果全不相同。

$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
(a)	(b)	(c)	(d)	(e)	(f)

图 3.4.5 实现 2-D 中值滤波的 3 种方法的结果

最后要说明，这种不严格分解并不只是非线性滤波模板的特性，有些线性滤波模板也不能准确分解。究其原因是在一个 $n \times n$ 模板中的独立系数个数是 n^2 ，这比在 $n \times 1$ 或 $1 \times n$ 分量模板中的独立系数个数要多得多。

例 3.4.4 中值滤波器的分解计算

前面讨论了将 2-D 高斯平均模板分解为两个 1-D 高斯平均模板以减少计算量。对 2-D 中值滤波模板也可进行类似分解以减少计算量。但要注意，此时分解结果是有近似的。考虑原始图像为

0	0	0	0	0	0
0	0	1	1	2	0
0	0	2	2	0	0
0	0	0	0	0	0

考虑直接使用 3×3 的 2-D 中值滤波器，则结果为

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

如果先使用 1×3 的中值滤波器，再使用 3×1 的中值滤波器，结果依次为

0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	1	1	0	0
0	0	2	2	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0

如果先使用 3×1 的中值滤波器，再使用 1×3 的中值滤波器，结果依次为

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

可见，这里分别使用两次 1-D 中值滤波器的两种情况给出的结果相同，但直接使用 3×3 的 2-D 中值滤波器与它们的结果不一样。

现在再考虑原始图像为(注意它与上一个原始图像仅有一个像素不同)

```

0 0 0 0 0 0
0 0 1 1 2 0
0 0 0 2 0 0
0 0 0 0 0 0

```

如果直接使用 3×3 的 2-D 中值滤波器,则结果为

```

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

```

如果先使用 1×3 的中值滤波器,再使用 3×1 的中值滤波器,结果依次为

```

0 0 0 0 0 0  0 0 0 0 0 0
0 0 1 1 1 0  0 0 0 1 0 0
0 0 0 2 0 0  0 0 0 1 0 0
0 0 0 0 0 0  0 0 0 0 0 0

```

如果先使用 3×1 的中值滤波器,再使用 1×3 的中值滤波器,结果依次为

```

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

```

这里,分别使用两次 1-D 中值滤波器的两种情况给出的结果不相同,但直接使用 3×3 的 2-D 中值滤波器与使用两次 1-D 中值滤波器中的一种情况的结果相同。实际中,三种方法的结果也可能都相同或都不相同。□

从本质上说,很多滤波器这样分解都只能近似也可这样解释:在 $n \times n$ 模板中的独立参数是个 n^2 ,而在两个 1-D 滤波器模板中的独立参数只有 $2n$ 个。

中值滤波器是不可分离的。但使用优化算法可使它的运行时间与可分离滤波器基本相同。假如图像的尺寸是 $W \times H$,模板的尺寸是 $n \times n$,则计算时间为 $O(WHn)$ 。使用中值滤波器不能预测图像中的边缘是否会变化,也不能确定变化程度,还很难估计噪声消除的效果。

5. 序统计滤波

中值滤波实际上是一类更广泛的滤波——百分比(percentile)滤波的一个特例[Jähne 1997]。百分比滤波均基于模板的排序来工作,所以是一种序统计滤波。在中值滤波中,选取了灰度序列中位于 50%位置的像素作为滤波结果;如果选取了灰度序列中位于 0%位置的像素,就得到最小值(min)作为滤波结果;如果选取了灰度序列中位于 100%位置的像素,就得到最大值(max)作为滤波结果。这两种滤波的输出可分别表示为

$$g_{\max}(x, y) = \max_{(s, t) \in N(x, y)} [f(s, t)] \quad (3.4.4)$$

$$g_{\min}(x, y) = \min_{(s, t) \in N(x, y)} [f(s, t)] \quad (3.4.5)$$

中值滤波、最大值滤波和最小值滤波都适合于消除椒盐噪声(见 5.2.2 小节)。最大值

滤波可用来检测图像中最亮的点,并可减弱暗(低取值)的椒噪声;而最小值滤波可用来检测图像中最暗的点,并可减弱亮(高取值)的盐噪声。

如果在一幅具有较暗目标和较亮背景图像进行最大值滤波,则在滤波结果图中,较亮的背景区域将会扩张,而较暗的目标区域将会缩小。如果目标原来尺寸就较小,则目标甚至会被消除掉。最大值滤波还趋向于在图像中产生具有(高)常数灰度的区域。与此对应,最小值滤波会趋向于在图像中产生具有(低)常数灰度的区域。根据需要,可将最大值滤波和最小值滤波结合使用。例如**中点滤波**就是取最大值和最小值中点的那个值作为滤波的输出:

$$g_{\text{mid}}(x, y) = \frac{1}{2} \left\{ \max_{(s, t) \in N(x, y)} [f(s, t)] + \min_{(s, t) \in N(x, y)} [f(s, t)] \right\} = \frac{1}{2} \{ g_{\text{max}}(x, y) + g_{\text{min}}(x, y) \} \quad (3.4.6)$$

这种滤波方式结合了排序滤波和平均滤波两种方式的优点。它对多种随机分布的噪声,如高斯噪声和均匀噪声,都比较有效。

6. 最频值滤波

最频值代表一个分布中最有可能出现的值。均值、中值和最频值都是相对于模板覆盖的区域来说的。类似均值或中值滤波,**最频值滤波**表示用最频值作为滤波的输出。

使用最频值滤波不仅可以消除噪声(尤其是脉冲噪声),还可以锐化目标边缘。这是因为在接近边缘处的邻域中,最频值滤波会将最频值移动到更靠近边缘中心的位置,从而使边缘更加尖锐。这可以如下解释:在任何边缘的背景一边像素主要具有背景灰度,所以最频值滤波的输出是背景的灰度值;而在任何边缘的前景一边像素主要具有前景灰度,所以最频值滤波的输出是前景的灰度值。这样在边缘上某个特定点,局部灰度分布的主要峰会从背景变到前景或从前景变到背景,从而趋向于增强边缘。这与会模糊边缘的平均滤波不同。平均滤波会产生背景灰度和前景灰度混合的边缘剖面,从而导致减少两个区域间的局部对比度。

一个区域中的灰度分布可利用该区域的直方图来表示,而均值、中值和最频值都与直方图密切相关。一个区域的直方图的均值也给出了该区域中灰度的均值。一个区域的直方图的中值也给出了该区域中灰度的中值。一个区域的直方图的最频值就是统计值最大的灰度值。如果直方图是对称的且仅有一个峰,那么均值、中值和最频值都相同。如果直方图中仅有一个峰、但左右不对称,那么最频值对应那个峰,而中值总比均值更接近最频值。

例 3.4.5 均值、中值和最频值的位置关系

图 3.4.6 给出一幅图像的直方图以显示均值、中值和最频值三者之间的位置关系。

图 3.4.6 中,最频值的位置是 $7 (= \arg\{\max[H(z)]\})$,中值的位置是 $6(1+3+4+5+6=9+8+2)$,而均值的位置是 $5.69(=(1 \times 1 + 2 \times 3 + 3 \times 4 + 4 \times 5 + 5 \times 6 + 6 \times 7 + 7 \times 9 + 8 \times 8 + 9 \times 2) / (1+2+3+4+5+6+7+8+9) = 256/45))$ 。可见,中值的位置比均值的位置更接近最频值的位置。□

直接检测最频值有可能由于噪声的影响而不准确。如果中值已确定了,则可借助中值位置来进一步确定最频值位置。这里可使用截断中值滤波的方法来进行。首先根据中值把拖尾较长的那部分截断一些使得与未截断部分同样长,然后计算剩下部分的中值,再如上进

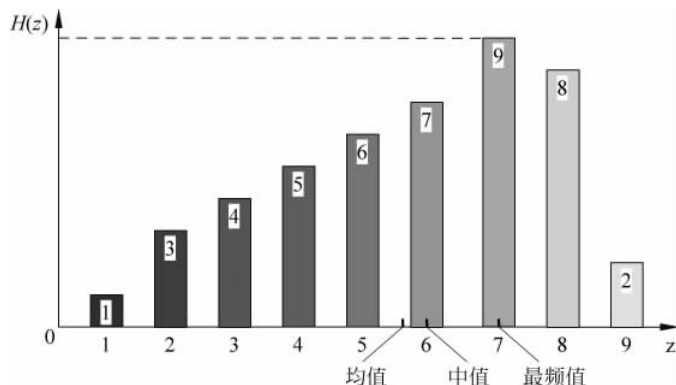


图 3.4.6 显示均值、中值和最频值位置关系的直方图

行截断,如此迭代就会逐渐逼近最频值位置。这里就利用了中值位置比均值位置更接近最频值位置的关系。

3.4.2 非线性锐化滤波

借助非线性滤波也可以获得对图像锐化的结果。此时的模板运算采用模板卷积进行。

1. 基于梯度的锐化滤波

图像处理中最常用的微分方法是利用**梯度**(基于一阶微分)。对一个连续函数 $f(x, y)$, 其梯度是一个矢量,由分别沿 X 和 Y 方向的两个偏导数分量组成:

$$\nabla f = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T = [G_X \quad G_Y]^T \quad (3.4.7)$$

在离散空间,微分用差分实现,两个常用的差分卷积模板见图 3.4.7(未标系数可取 0),分别计算沿 X 和 Y 两个方向的差分。

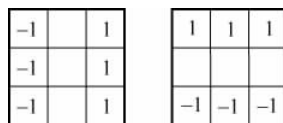


图 3.4.7 两个差分模板

实际锐化滤波中,常只使用这个矢量的幅度(即矢量的模)。参见 3.1.3 小节,矢量的模可分别以 2 为范数计算(对应欧氏距离),以 1 为范数计算(对应城区距离)或以 ∞ 为范数计算(对应棋盘距离)[Zhang 1993]:

$$|\nabla f_{(2)}| = \text{mag}(\nabla f) = [G_X^2 + G_Y^2]^{1/2} \quad (3.4.8)$$

$$|\nabla f_{(1)}| = |G_X| + |G_Y| \quad (3.4.9)$$

$$|\nabla f_{(\infty)}| = \max\{|G_X|, |G_Y|\} \quad (3.4.10)$$

注意,上述这些组合模板计算结果的方法本身都是非线性的。

2. 最大-最小锐化变换

最大-最小锐化变换是一种将最大值滤波和最小值滤波结合使用的图像增强技术,可以锐化模糊的边缘并让模糊的目标清晰起来[Ritter 2001]。这种方法可以迭代进行,在每次迭代中将一个模板覆盖区域里的中心像素值与该区域里的最大值和最小值进行比较,然后将中心像素值用与其较接近的极值(最大值或最小值)替换。

最大-最小锐化变换 S 定义为

$$S[f(x,y)] = \begin{cases} g_{\max}(x,y) & \text{如果 } g_{\max}(x,y) - f(x,y) \leq f(x,y) - g_{\min}(x,y) \\ g_{\min}(x,y) & \text{其他} \end{cases} \quad (3.4.11)$$

在图像增强中,可将这个计算过程迭代进行:

$$S^{n+1}[f(x,y)] = S\{S^n[f(x,y)]\} \quad (3.4.12)$$

3. 锐化滤波模板的通用性质

还有一些其他锐化滤波模板可实现对图像锐化的结果,但它们都具有以下性质。

(1) 零位移

锐化滤波模板通过检测和加强图像中对应边缘的部分来增强图像的视觉效果,但不应该改变图像中边缘的位置。一阶微分滤波模板应是反(anti)对称的[Jähne 1997]。因为锐化滤波需要用两个分别沿 X 和 Y 方向的模板,所以反对称的卷积模板 $G_X(x,y)$ 和 $G_Y(x,y)$ 分别应该满足:

$$G_X(-x,y) = -G_X(x,y) \quad (3.4.13)$$

$$G_Y(x,-y) = -G_Y(x,y) \quad (3.4.14)$$

如果卷积模板中的系数个数是奇数,则处在中心的系数应为 0。以图 3.2.1(b)中的模板为例,如果用于沿 X 方向,应有 $k_4 = -k_2, k_5 = -k_1, k_6 = -k_8$; 如果用于沿 Y 方向,应有 $k_4 = -k_6, k_3 = -k_7, k_2 = -k_8$ 。这实际上就是图 3.4.7 给出的一对模板。

(2) 消除均值(suppression of mean value)

任意阶的微分滤波模板都不应该对常数值有响应。这个条件指所有系数的和应该为 0,即

$$\sum_{i=0}^{N-1} k_i = 0 \quad (3.4.15)$$

(3) 对称性质

零位移的条件暗示一阶微分算子一般有奇数个(反)对称的系数。这样,沿一个方向的卷积计算可简化为

$$g_i = \sum_{j=1}^N k_j (f_{i-j} - f_{i+j}) \quad (3.4.16)$$

对 $2r+1$ 个模板系数,只需要 r 次乘法。但加法的次数仍是 $2r-1$ 。

3.4.3 线性和非线性混合滤波

线性滤波和非线性滤波各有优点和缺点,实际中也常将它们结合使用,以取长补短。下面给出一个结合线性和非线性滤波以加快运算的方法。

当使用比较大尺寸的模板时,实现非线性滤波会需要很大的计算量。解决这个问题的一种方法是结合使用线性滤波(一般比较快速)和非线性滤波(如排序统计滤波),使组合后滤波的效果接近所期望的要求,但在计算复杂度方面有较大改进[Dougherty 1994]。

在混合滤波中,常将线性滤波运算和中值滤波运算混合串联起来,先对较大的区域进行计算量较小的线性滤波操作,然后再计算线性滤波输出的中值,作为混合滤波的最终输出。

考虑一个 1-D 信号 $f(i)$ 。用子结构 $H_1, H_2, \dots, H_M$ 组成的线性中值混合滤波定义为

$$g(i) = \text{MED}[H_1(f(i)), H_2(f(i)), \dots, H_M(f(i))] \quad (3.4.17)$$

式中, $H_1, H_2, \dots, H_M$ (M 为奇数) 是线性滤波模板。选择子滤波模板 H_i 使得在噪声消除和保留信号间取得一个可接受的妥协, 并保持 M 足够小以简化计算。作为一个例子, 考虑下面的结构:

$$g(i) = \text{MED}[H_L(f(i)), H_C(f(i)), H_R(f(i))] \quad (3.4.18)$$

式中, 滤波模板 H_L, H_C 和 H_R 都是平滑滤波模板。下标 L、C 和 R 分别代表左、中、右, 指示相对于当前输出值的滤波模板的对应位置, 如图 3.4.8 所示。

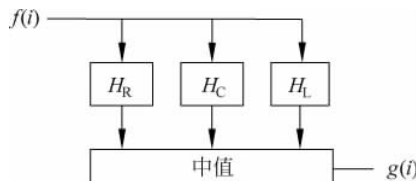


图 3.4.8 利用子滤波模板实现基本的线性和中值混合滤波

最简单的结构包括使用相同的平均滤波模板 H_L 和 H_R 以及直通的模板 $H_C[f(i)] = f(i)$ 。此时整个滤波运算可表示为

$$g(i) = \text{MED}\left[\frac{1}{k} \sum_{i=1}^k f(i-k), f(i), \frac{1}{k} \sum_{i=1}^k f(i+k)\right] \quad (3.4.19)$$

用这样的结构进行滤波与采用标准的中值有很相似的滤波效果, 但计算要快得多。如果利用迭代游程求和, 它的计算复杂度与窗口尺寸无关, 是一个常数。

在实际的 2-D 图像应用中, 常取子滤波模板的个数为 5。例如, 下列滤波运算

$$g(x, y) = \text{MED}\left\{\frac{1}{2}[f(x, y-2) + f(x, y+2)], \frac{1}{2}[f(x, y-1) + f(x, y+1)], f(x, y), \frac{1}{2}[f(x+2, y) + f(x-2, y)], \frac{1}{2}[f(x+1, y) + f(x-1, y)]\right\} \quad (3.4.20)$$

对应图 3.4.9(a) 所示的模板。图 3.4.9(b) 和 (c) 给出其他两个典型的模板。

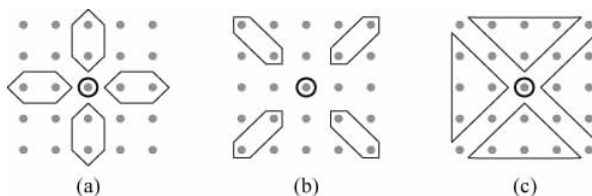


图 3.4.9 用于线性和中值混合滤波的模板

3.5 局部增强

以上几节在介绍增强方法时都把一幅图像作为一个整体, 在计算变换或滤波参数时都基于整幅图像的统计量, 并对整幅图进行统一增强操作。在实际应用中常常需要对图像某些局部区域的细节进行增强。这些局部区域内的像素数量相对于整幅图的像素数量往往较小, 在计算整幅图的变换或滤波参数时其影响常被忽略掉, 而用从整幅图算得的变换或滤波参数并不能保证在这些所关心的局部区域都得到所需要的增强效果。

为解决这类问题,需要根据所关心的局部区域的特性来计算变换或滤波参数,并将结果仅用于所关心的局部区域,以得到所需的相应的局部增强效果。由此可见,局部增强方法比全局增强方法在具体进行增强操作前多了一个选择确定局部区域的步骤,而对每个局部区域仍可采用前几节介绍的增强方法进行增强。

1. 直方图变换局部增强

直方图变换是空域增强中最常采用的方法,它也很容易用于图像的局部增强。只需先将图像分成一系列小区域(子图像),此时直方图均衡化或规定化都可以基于小区域内的像素分布进行,从而使得各个小区域得到不同的增强效果。由于增强是对各个局部区域分别进行的,增强效果在小区域的边界上有可能不协调,所以实际中常将图像划分成为有些互相重叠的小区域以减小这种影响。

例 3.5.1 局部均衡化增强示例

局部增强常用于一些要求特定增强效果的场合,一个示例如图 3.5.1 所示。图(a)为一幅整体灰度偏暗的图像,图(b)为用全局直方图均衡化得到的结果。由图(b)可见,均衡化以后,整图的对比度会增加较多,但同时噪声也被放大了许多倍。换句话说,原图像中一些灰度值仅有小起伏的区域现在变得黑白分明,比较生硬。图(c)为将原图分为互不重叠的尺寸为 7×7 的子图像,然后进行局部直方图均衡化得到的结果,整个图像与图(b)比较相对柔和。

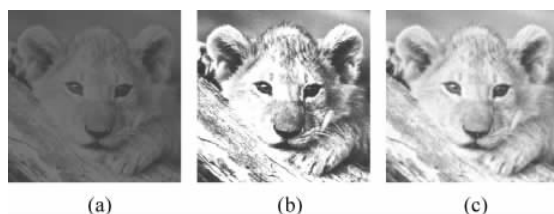


图 3.5.1 局部增强效果示例

□

与前面介绍的直方图均衡化方法相比,虽然那里均衡化后可使动态范围增加,但增强图像中的灰度级数并不会相较增强以前变得更多,只是灰度级间的距离增加。而局部增强在增加动态范围的同时也可以增加全图中的灰度级数,避免了像素间的生硬过渡,视觉效果较柔和。

2. 基于均值和方差的局部增强

局部增强除可借助先将图像分成子图像再对每个子图像分别增强外,也可在对整幅图像增强时直接利用局部信息以实现对不同的局部或不同的像素进行不同的增强。这里常用的局部信息主要是每个像素的邻域内所有像素的灰度均值和灰度方差,其中灰度均值是一个平均亮度的测度,而灰度方差是一个亮度反差(对比度)的测度。一种简单的方法是利用灰度均值和灰度方差来选择需增强的像素。例如,设需要增强图像 $f(x,y)$ 中灰度比较大但方差比较小的区域,可借助下式对每个像素进行计算来得到输出图 $g(x,y)$:

$$g(x,y) = \begin{cases} Ef(x,y) & M \leq km(x,y) \text{ 且 } \sigma(x,y) \leq lS \\ f(x,y) & \text{其他} \end{cases} \quad (3.5.1)$$

式中, $m(x,y)$ 和 $\sigma(x,y)$ 分别是以像素 (x,y) 为中心的邻域内像素的灰度均值和均方差值;

M 是 $f(x, y)$ 的灰度均值; S 是 $f(x, y)$ 的均方差值; k 和 l 都是比例常数(可都取 < 0.5); E 是增强系数 (> 2)。逻辑运算前的不等式用于选择相对于全图比较亮的区域, 逻辑运算后的不等式用于选择相对于全图方差比较小的区域。

另一种常用的方法也是利用每个像素的邻域内所有像素的均值和方差这两个特性进行的。具体来说, 如要把输入图 $f(x, y)$ 根据各个像素的邻域特性增强成输出图 $g(x, y)$, 需要在每个像素位置 (x, y) 进行如下变换:

$$g(x, y) = A(x, y)[f(x, y) - m(x, y)] + m(x, y) \quad (3.5.2)$$

其中

$$A(x, y) = k \frac{M}{\sigma(x, y)} \quad 0 < k < 1 \quad (3.5.3)$$

称为局部增益函数。

在以上两式中, $m(x, y)$ 、 $\sigma(x, y)$ 、 M 和 k 的意义均如前。注意 $A(x, y)$ 、 $m(x, y)$ 和 $\sigma(x, y)$ 的值都与所选邻域的位置和大小有关。将 $A(x, y)$ 与 $f(x, y)$ 和 $m(x, y)$ 的差值相乘能放大图像的局部变化。因为 $A(x, y)$ 反比于均方差, 所以在图像中对对比度较小的区域得到的增益反而较大, 这样就可以取得局部对比度增强的效果。式(3.5.2)中最后又将 $m(x, y)$ 加回去是为了恢复原区域的平均灰度值。实际中为了平衡图像中孤立区域里灰度值的偏移, 常常只将 $m(x, y)$ 的一部分(如乘以一个小于 1 的常数)加回去, 并将 $A(x, y)$ 限制在一定的范围内。

总结和复习

为更好地学习, 下面对各小节给予概括小结并提供一些进一步的参考资料; 另外给出一些思考题和练习题以帮助复习(文后对加星号的题目还提供了解答)。

1. 各节小结和文献介绍

3.1 节概括介绍了像素间的联系, 包括像素的邻域, 像素间的邻接、连接和连通, 以及像素间的距离等概念。城区距离和棋盘距离都是数字图像中特有的, 它们的特点在等距离轮廓中表现得比较明显。在正方形像素间的连接中, 除了 4-连接和 8-连接外, 有时还需要使用 m -连接[章 2006b]; 如果将 4-邻域和 8-邻域扩展到 16-邻域, 则还需要使用 M -连接[章 2000c]。另外, 对距离的讨论还可见文献[Basseville 1989], 对范数的详细讨论可见文献[数 2000]。

3.2 节讨论模板运算(包括模板卷积和模板排序), 其中特别介绍了在图像边界处进行模板操作时所采取的方法, 这是在实际应用中会遇到的具体问题。考虑到模板运算是一种通用的手段, 所以对它可实现的增强功能进行了概述, 最后对利用模板操作可实现的图像空域增强技术进行了分类。本节内容在各种图像书籍(如文献[Pratt 2007]、[Gonzalez 2008])中均有介绍。

3.3 节讨论线性滤波的原理和基本方法。这里滤波是指那些借助模板进行的、利用像素及其邻域像素性质的图像处理方法。线性滤波的方法很多, 除了本节介绍的一些常用典型技术外, 还可参见各种图像处理教材, 如[Pratt 2007]、[Gonzalez 2008]、[Zhang 2009b]等。

3.4 节讨论非线性滤波的原理和基本方法。非线性滤波与线性滤波既有联系也有区别,功能和效果也不尽相同。本节主要介绍了对消除灰度图像中脉冲噪声比较有效的中值滤波,对二值图像的中值滤波可见文献[Marchand 2000],一种基于比特的快速中值滤波方法可见文献[Davies 2005]、[Davies 2012]。更多的非线性滤波器可见文献[Mitra 2001],利用各向异性扩散(anisotropic diffusion)进行图像增强可见文献[Nikolaidis 2001]。将不同的滤波器相结合可以取长补短。除这里介绍的结合线性和非线性滤波以减少运算量并保持滤波效果的方法外,5.2.5 小节还将介绍一种根据噪声类型选择对应的有效滤波器以发挥它们的各自特点提高滤波效果的方法。另一个混合滤波器的实例可见文献[王 2012]。

3.5 节介绍的局部增强方案并不是一种独立的图像增强方案,它是对已有增强技术的一种扩展应用。由于它利用的是图像里局部区域的统计特性,因此其增强效果会在所关心的局部区域更明显和有效。

2. 思考题和练习题

3-1 设有两个图像子集如图题 3-1 所示,如果 $V=\{1\}$,判断:

(1) 子集 P 和子集 Q 是否: ①4-邻接; ②8-邻接。

(2) 子集 P 和子集 Q 是否: ①4-连接; ②8-连接。

(3) 子集 P 和子集 Q 是否: ①4-连通; ②8-连通。

(4) 如果将子集 P 和子集 Q 以外的所有像素看成另一个子集 R ,子集 P 和子集 Q 是否与子集 R : ①4-连接; ②8-连接。

***3-2** 考虑如图题 3-2 所示图像子集:

	P					Q				
0	1	0	1	0		1	0	1	1	0
1	1	0	1	0		1	1	0	1	0
0	0	1	0	1		0	1	1	0	0
0	1	1	0	0		1	0	1	0	1
1	0	0	1	1		0	0	0	0	1

图题 3-1

1	1	2	$l(q)$
2	0	0	2
1	2	1	1
$(p)l$	0	1	2

图题 3-2

(1) 令 $V=\{0,1\}$,计算 p 和 q 之间 4-通路和 8-通路的长度;

(2) 令 $V=\{1,2\}$,仍计算上述两个长度。

3-3 空间滤波器在具体实现时需要让模板的中心移过图像中的每个位置,计算模板系数和对应像素的乘积并对它们求和。对所有模板系数均为 1 的低通滤波器,可使用称为盒滤波器或移动平均的算法程序,即每次只计算在模板移动中其值变化的部分。

(1) 写出对一个 $n \times n$ 的滤波器实现上述快速算法的主要步骤。

(2) 用直接算法和用上述快速算法所需的计算次数之比称为计算效益。忽略图像边界的影响,计算当 n 为 3~11 的奇数时的计算效益,并画出计算效益为 n 的函数的曲线。注意 $1/n^2$ 的尺度系数对两种算法是公共的,所以这里可不考虑。

3-4 将小尺寸的模板反复使用,相当于用对应小尺寸的模板进行卷积得到的结果去进行滤波,所以可以得到加权大尺寸模板的效果。现在考虑有两个 3×3 的平均滤波器模板,计算其效果与将它们依次使用所对应的加权大尺寸模板。

3-5 将 M 幅图像相加求平均可以起到消除噪声的效果,用一个 $n \times n$ 的模板进行平滑滤波也可以起到消除噪声的效果,试比较这两种方法的消噪效果。

3-6 讨论用于空间滤波的平滑滤波器和锐化滤波器的相同点、不同点以及联系。

***3-7** 设有从图像中获得的一个 1-D 剖面: 0 0 1 2 2 3 4 9 5 6 6 7 8 8 9。

(1) 使用一个有 3 个元素的模板对该 1-D 剖面进行中值滤波,给出滤波结果。

(2) 从滤波结果看,中值滤波除可消除脉冲噪声还有什么特点?

3-8 试给出与图 3.4.8(b)和(c)两个模板的滤波运算对应的表达式。

3-9 考虑先将一幅灰度图像用 3×3 平均滤波器平滑一次,再进行如下增强:

$$g(x, y) = \begin{cases} G[f(x, y)] & G[f(x, y)] \geq T \\ f(x, y) & \text{其他} \end{cases}$$

其中, $G[f(x, y)]$ 是 $f(x, y)$ 在 (x, y) 处的梯度; T 是非负的阈值。

(1) 比较原始图像和增强图像,在哪些地方图像会得到增强?

(2) 改变阈值 T 的数值,这会对增强效果带来哪些影响?

3-10 考虑如下的增强算法: 在每个像素位置,计算其水平方向上左边一个和右边一个位置的两个像素的灰度差 H ,计算其垂直方向上高一个和低一个位置的两个像素的灰度差 V 。如果 $V > H$,将该像素的灰度变为水平方向上两个像素的灰度和的平均值,否则,将该像素的灰度变为垂直方向上两个像素的灰度和的平均值。

(1) 讨论该算法的效果特点。

(2) 如果反复利用该算法,可获得什么效果?

3-11 给定如下 2-D 原始图像(令左上角像素的坐标为 $(0, 0)$),假设取 $k=0.6$,邻域取 3×3 ,计算 $(1, 1)$, $(2, 2)$ 和 $(3, 3)$ 点经局部增强后的灰度值。

14	11	21	16	20
24	32	64	32	24
25	26	40	32	30
32	30	33	30	32
11	33	22	11	22

3-12 如果仅需要增强图像中灰度比较小且方差也比较小的区域,此时可根据下式进行:

$$g(x, y) = \begin{cases} Ef(x, y) & M_w \leq aM_f \quad \text{且} \quad bS_f \leq S_w \leq cS_f \\ f(x, y) & \text{其他} \end{cases}$$

式中, a, b, c, E 均为系数(一般 $a < 0.5, b < c < 0.5, 2 < E < 5$), M_w 和 S_w 分别是以 (x, y) 为中心的图像窗口 W (常可取 3×3) 中的灰度均值和灰度方差, M_f 和 S_f 分别是图像 $f(x, y)$ 的灰度均值和灰度方差。逻辑运算前的不等式用于选择相对于全图比较偏暗的区域,逻辑运算后的不等式用于选择相对于全图方差比较小的区域。

(1) 分析上述增强方法的原理,为什么两个不等式可帮助选择不同区域?

(2) 如果需要增强图像中灰度比较大但方差比较小的区域,应如何调整上式,各系数如何选择?