

像素空间关系

3.1 像素间的基本关系

图像的基本组成单元是像素,像素在图像空间中按照某种规律排列,有一定的相互联系。常见的像素间的基本关系包括像素的邻域、邻接、连接、通路和连通以及像素集合间的邻接、连接和连通。

3.1.1 像素的邻域

像素的邻域是指一个像素的相邻像素构成的像素集。邻域的类型一般包括4邻域、对角邻域和8邻域,如图3-1所示。

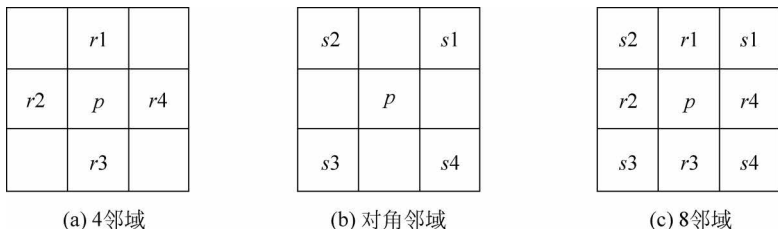


图 3-1 邻域的类型

1. 4邻域

像素 p 的4邻域是指由像素 p 的水平方向(即左右)和垂直方向(即上下)共4个相邻像素所组成的集合,记为 $N_4(p) = \{r1, r2, r3, r4\}$ 。若像素 p 的坐标为 (x, y) ,则像素 p 的4邻域像素的坐标分别为 $r1: (x-1, y)$ 、 $r2: (x, y-1)$ 、 $r3: (x+1, y)$ 和 $r4: (x, y+1)$,如图3-1(a)所示。

2. 对角邻域

像素 p 的对角邻域是指由像素 p 的对角方向(即左上、左下、右上、右下)共4个相邻像素所组成的集合,记为 $N_D(p) = \{s1, s2, s3, s4\}$ 。若像素 p 的坐标为 (x, y) ,则像素 p 的对角邻域像素的坐标分别为 $s1: (x-1, y+1)$ 、 $s2: (x-1, y-1)$ 、 $s3: (x+1, y-1)$ 和 $s4: (x+1, y+1)$ 。

$y+1$),如图 3-1(b)所示。

3. 8 邻域

像素 p 的 8 邻域是指由像素 p 的水平、垂直和对角方向(即左右、上下、左上、左下、右上、右下)共 8 个相邻像素所组成的集合,记为 $N_8(p)=\{r1,r2,r3,r4,s1,s2,s3,s4\}$ 。若像素 p 的坐标为 (x,y) ,则像素 p 的 8 邻域像素的坐标分别为 $r1:(x-1,y),r2:(x,y-1),r3:(x+1,y),r4:(x,y+1),s1:(x-1,y+1),s2:(x-1,y-1),s3:(x+1,y-1)$ 和 $s4:(x+1,y+1)$,如图 3-1(c)所示。

需要注意的是,如果像素 p 本身处于图像的边缘,则它的 4 邻域 $N_4(p)$,对角邻域 $N_D(p)$ 和 8 邻域 $N_8(p)$ 中的若干个像素将位于图像之外。

3.1.2 像素的邻接

像素的邻接是指一个像素与其邻域中的像素的接触关系。邻接的类型根据邻域的类型不同一般分为 4 邻接、对角邻接和 8 邻接。

1. 4 邻接

4 邻接是指一个像素与其 4 邻域中的像素的接触关系。若两个像素为 p 和 r ,则像素 p 与像素 r 满足 4 邻接可表示为

$$p \in N_4(r) \quad (3-1)$$

注意: 像素 p 与像素 r 满足 4 邻接等价于像素 r 与像素 p 满足 4 邻接,即式(3-2)成立。

$$p \in N_4(r) \Leftrightarrow r \in N_4(p) \quad (3-2)$$

2. 对角邻接

对角邻接是指一个像素与其对角邻域中的像素的接触关系。若两个像素为 p 和 r ,则像素 p 与像素 r 满足对角邻接可表示为

$$p \in N_D(r) \quad (3-3)$$

注意: 像素 p 与像素 r 满足对角邻接等价于像素 r 与像素 p 满足对角邻接,即式(3-4)成立。

$$p \in N_D(r) \Leftrightarrow r \in N_D(p) \quad (3-4)$$

3. 8 邻接

8 邻接是指一个像素与其 8 邻域中的像素的接触关系。若两个像素为 p 和 r ,则像素 p 与像素 r 满足 8 邻接可表示为

$$p \in N_8(r) \quad (3-5)$$

注意: 像素 p 与像素 r 满足 8 邻接等价于像素 r 与像素 p 满足 8 邻接,即式(3-6)成立。

$$p \in N_8(r) \Leftrightarrow r \in N_8(p) \quad (3-6)$$

需要注意的是,邻接仅考虑了像素间的空间关系,与像素的属性值无关。

3.1.3 像素的连接

两个像素的连接是指两个像素必须邻接(即接触)且它们的属性值必须满足某个特定的相似准则。属性值一般采用像素的灰度值。相似准则可以是灰度值相等,或者同在一个灰

度值集合中取值,记为 V 。例如,在一张二值图像中,定义两个灰度值为 1 的像素之间的连接,可以取相似准则为灰度值集合 $V=\{1\}$; 在一张 256 色的灰度图像中,定义灰度值为 100~105 的像素之间的连接,可以取相似准则为灰度值集合 $V=\{100,101,102,103,104,105\}$ 。

连接的类型根据邻域的类型的一般分为 4 连接、对角连接、8 连接以及混合连接。

1. 4 连接

4 连接是指两个像素 4 邻接且它们的属性值满足某个特定的相似准则。若两个像素为 p 和 r ,像素的属性值函数为 $f()$,相似准则为 V ,则 4 连接的条件可表示为

$$(p \in N_4(r)) \wedge (f(p) \in V) \wedge (f(r) \in V) \tag{3-7}$$

2. 对角连接

对角连接是指两个像素对角邻接且它们的属性值满足某个特定的相似准则。若两个像素为 p 和 r ,像素的属性值函数为 $f()$,相似准则为 V ,则对角连接的条件可表示为

$$(p \in N_D(r)) \wedge (f(p) \in V) \wedge (f(r) \in V) \tag{3-8}$$

3. 8 连接

8 连接是指两个像素 8 邻接且它们的属性值满足某个特定的相似准则。若两个像素为 p 和 r ,像素的属性值函数为 $f()$,相似准则为 V ,则 8 连接的条件可表示为

$$(p \in N_8(r)) \wedge (f(p) \in V) \wedge (f(r) \in V) \tag{3-9}$$

4. 混合连接

混合连接又称 m 连接,是指两个像素的属性值必须满足某个特定的相似准则且满足下列两个条件之一: ①两个像素 4 邻接; ②两个像素对角邻接且它们 4 邻域的交集在相似准则的意义下是空集。若两个像素为 p 和 r ,像素的属性值函数为 $f()$,相似准则为 V ,则混合连接的条件可表示为式(3-10)或式(3-11):

$$(p \in N_4(r)) \wedge (f(p) \in V) \wedge (f(r) \in V) \tag{3-10}$$

$$(p \in N_D(r)) \wedge (f(N_D(p) \cap N_D(r)) \cap V = \phi) \tag{3-11}$$

例 3.1 混合连接的判定。一张 4×4 的二值图像模板如图 3-2(a)所示,相似准则 $V=\{1\}$,当图像数据如图 3-2(b)和图 3-2(c)取值时,判定像素 p 和 r 是否满足混合连接的条件。

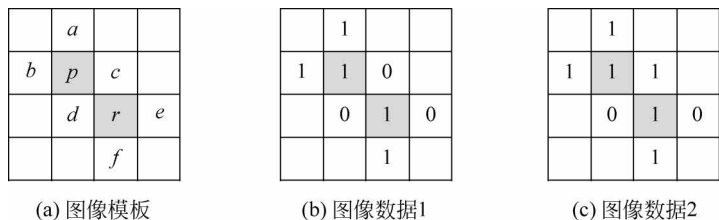


图 3-2 混合连接的判定

解: 由图像模板可知,像素 p 和 r 互为对角邻接,即 $p \in N_D(r)$,且像素 p 和 r 对角邻域的交集为: $N_D(p) \cap N_D(r) = \{c, d\}$ 。

(1) 当图像数据如图 3-2(b)取值时,有: $f(c)=0, f(d)=0$,则

$$f(N_D(p) \cap N_D(r)) = f(\{c, d\}) = f(c) \cup f(d) = \{0\} \cup \{0\} = \{0\}$$

因此, $f(N_D(p) \cap N_D(r)) \cap V = \{0\} \cap \{1\} = \emptyset$

所以,像素 p 和 r 满足混合连接的条件。

(2) 当图像数据如图 3-2(c)取值时,有: $f(c)=1, f(d)=0$, 则

$$f(N_D(p) \cap N_D(r)) = f(\{c, d\}) = f(c) \cup f(d) = \{1\} \cup \{0\} = \{0, 1\}$$

因此, $f(N_D(p) \cap N_D(r)) \cap V = \{0, 1\} \cap \{1\} = \{1\} \neq \emptyset$

所以,像素 p 和 r 不满足混合连接的条件。

混合连接可以认为是 8 连接的一种变型,引进混合连接是为了消除使用 8 连接时常出现的多路问题。

例 3.2 多路问题示例。一张 3×3 的二值图像模板如图 3-3(a)所示,相似准则 $V = \{1\}$,像素 a, b, c, d 的灰度值均取 1,像素 e 的灰度值取 0。试分析采用 8 连接和混合连接时像素 a, b, c, d 的连接情况。

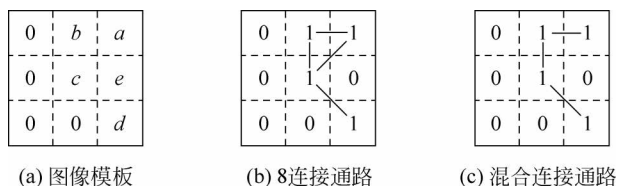


图 3-3 多路问题示例

解: 采用 8 连接时,像素 a, b, c, d 的连接情况如图 3-3(b)所示。此时,像素 a 和 c 之间存在两条 8 连接通路,即 $a-b-c$ 和 $a-c$,导致像素 a 和 c 之间产生多路问题。

采用混合连接时,像素 a, b, c, d 的连接情况如图 3-3(c)所示。此时,像素 a 和 b 之间满足混合连接条件,像素 b 和 c 之间满足混合连接条件,因此,像素 a 和 c 之间存在一条混合连接通路 $a-b-c$ 。

另外, $f(N_D(a) \cap N_D(c)) = f(\{b, e\}) = f(b) \cup f(e) = \{1\} \cup \{0\} = \{0, 1\}$

因此, $f(N_D(a) \cap N_D(c)) \cap V = \{0, 1\} \cap \{1\} = \{1\} \neq \emptyset$

所以,像素 a 和 c 之间不满足混合连接条件,即不存在混合连接通路 $a-c$ 。

因此,采用混合连接时,像素 a 和 c 之间不存在多路问题。

3.1.4 像素的通路

从一个具有坐标 (x, y) 的像素 p 到另一个具有坐标 (s, t) 的像素 q 的一条通路定义为由一系列具有坐标 $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ 的独立像素组成的集合,并且满足以下条件:

- (1) $(x_0, y_0) = (x, y), (x_n, y_n) = (s, t)$;
- (2) (x_i, y_i) 与 (x_{i-1}, y_{i-1}) 邻接;
- (3) $1 \leq i \leq n, n$ 为通路的长度。

根据邻接类型的不同,通路的类型也不同。如果像素间邻接的类型为 4 邻接、对角邻接或 8 邻接,则对应通路的类型为 4 通路、对角通路或 8 通路。

3.1.5 像素的连通

像素的连通是指像素通路上所有像素的属性值均满足某个特定的相似准则。根据连接类型的不同,连通的类型也不同。如果像素间连接的类型为 4 连接、对角连接、8 连接或混

合连接,则对应连通的类型为4连通、对角连通、8连通或混合连通。

像素连接可以看作是像素连通的一种特例。当 $n=1$ 时,两个连通的像素也是连接的。

3.1.6 像素集合的邻接

图像可以看作是像素的集合。像素集合根据一定的规则可以划分成若干个子集,每一个子集就构成了一个子图像。对于两个图像子集 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,则称两个图像子集 S 和 T 是邻接的。如果像素间邻接的类型为4邻接、对角邻接或8邻接,则对应图像子集的邻接类型为4邻接、对角邻接或8邻接。

3.1.7 像素集合的连接

两个图像子集 S 和 T 是连接的是指两个图像子集必须邻接且邻接像素的属性值必须满足某个特定的相似准则。也就是说,对于两个图像子集 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连接,则称两个图像子集 S 和 T 是连接的。如果像素间连接的类型为4连接、对角连接、8连接或混合连接,则对应图像子集的连接类型为4连接、对角连接、8连接或混合连接。

3.1.8 像素集合的连通

对于图像子集 S 中的两个像素 p 和 q 来说,如果存在一条完全由 S 中的像素组成的从 p 到 q 的通路,则称 p 在 S 中与 q 连通。对于 S 中的任一个像素 p ,所有与 p 相连通且又在 S 中的像素的集合(包括 p)称为 S 中的一个连通组元。图像里同一个连通组元中的任意两个像素相互连通,而不同连通组元中的像素互不连通。

图像中每个连通组元构成图像中的一个区域。图像可以认为是由一系列区域组成。区域的边界也称区域的轮廓,是该区域的一个子集,它将该区域与其他区域分开。组成一个区域边界的像素本身属于该区域而在其邻域中存在不属于该区域的像素。

3.2 像素间的距离

像素间的距离是指像素在空间的接近程度。设3个像素为 p 、 q 和 r ,坐标分别为 (x, y) 、 (s, t) 和 (u, v) ,则距离度量函数 D 必须满足下列三个条件:

- (1) $D(p, q) \geq 0$;
- (2) $D(p, q) = D(q, p)$;
- (3) $D(p, r) \leq D(p, q) + D(q, r)$ 。

其中,条件(1)表明两个像素之间的距离总是为正值,若 $D(p, q) = 0$ 当且仅当 $p = q$;条件(2)表明像素之间的距离与起点和终点的选择无关;条件(3)表明像素之间的最短距离是沿直线的。

常见的距离度量方法包括欧氏距离、城区距离、棋盘距离和混合距离。

3.2.1 欧氏距离

欧氏(Euclidean)距离记为 D_E ,设像素点 p 的坐标为 (x, y) ,像素点 q 的坐标为 (s, t) ,

则像素点 p 和 q 之间的欧氏距离的定义如下式所示：

$$D_E(p, q) = \sqrt{(x-s)^2 + (y-t)^2} \quad (3-12)$$

欧氏距离的几何意义为：距离坐标为 (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，如图 3-4 所示。

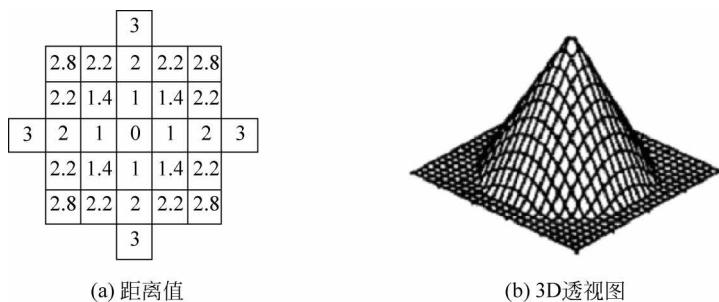


图 3-4 欧氏距离

其中，图 3-4(a)为距离坐标为 (x, y) 的像素的 D_E 距离小于或等于 3 的像素所构成的圆形区域，其值为各像素点距离中心像素点的距离值，该值已经过四舍五入处理；图 3-4(b)为距离值对应的 3D 透视图。

3.2.2 城区距离

城区(city-block)距离记为 D_4 ，设像素点 p 的坐标为 (x, y) ，像素点 q 的坐标为 (s, t) ，则像素点 p 和 q 之间的城区距离的定义如下式所示：

$$D_4(p, q) = |x-s| + |y-t| \quad (3-13)$$

城区距离的几何意义为：距离坐标为 (x, y) 的像素的 D_4 距离小于或等于某个值 d 的像素都包括在以 (x, y) 为中心的菱形中，如图 3-5 所示。其中，图 3-5(a)为距离坐标为 (x, y) 的像素的 D_4 距离小于或等于 3 的像素所构成的菱形区域，其值为各像素点距离中心像素点的距离值；图 3-5(b)为距离值对应的 3D 透视图。

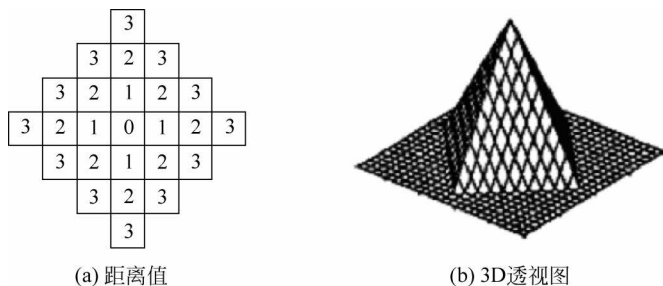


图 3-5 城区距离

由城区距离的定义可知，距离像素 p 的城区距离为 1 的像素就是像素 p 的 4 邻域像素。因此，像素 p 的 4 邻域可以通过城区距离定义为

$$N_4(p) = \{r \mid D_4(p, r) = 1\} \quad (3-14)$$

式中， r 为某个像素。

3.2.3 棋盘距离

棋盘(chessboard)距离记为 D_8 , 设像素点 p 的坐标为 (x, y) , 像素点 q 的坐标为 (s, t) , 则像素点 p 和 q 之间的棋盘距离的定义如下式所示:

$$D_8(p, q) = \max(|x - s|, |y - t|) \quad (3-15)$$

棋盘距离的几何意义为: 距离坐标为 (x, y) 的像素的 D_8 距离小于或等于某个值 d 的像素都包括在以 (x, y) 为中心的正方形中, 如图 3-6 所示。

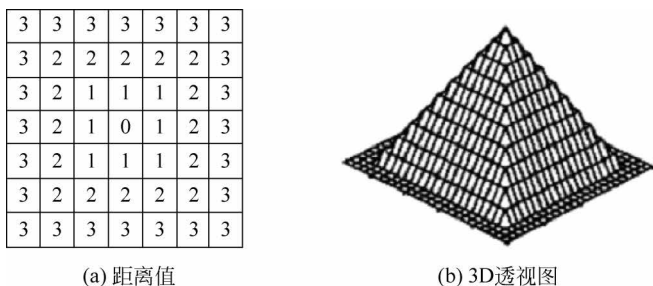


图 3-6 棋盘距离

其中, 图 3-6(a)为距离坐标为 (x, y) 的像素的 D_8 距离小于或等于 3 的像素所构成的正方形区域, 其值为各像素点距离中心像素点的距离值; 图 3-6(b)为距离值对应的 3D 透视图。

由棋盘距离的定义可知, 距离像素 p 的棋盘距离为 1 的像素就是像素 p 的 8 邻域像素。因此, 像素 p 的 8 邻域可以通过棋盘距离定义为式(3-16)。

$$N_8(p) = \{r \mid D_8(p, r) = 1\} \quad (3-16)$$

式中, r 为某个像素。

3.2.4 混合距离

像素点之间的欧氏距离(D_E)、城区距离(D_4)和棋盘距离(D_8)刻画了像素在空间的接近程度, 其大小仅与像素的坐标有关, 与像素本身及其邻近像素的属性值无关。两个像素点 p 和 q 之间的欧氏距离 $D_E(p, q)$ 等于它们之间的直线距离, 通常无法对应某条具体通路。城区距离 $D_4(p, q)$ 等于它们之间最短的 4 通路的长度。棋盘距离 $D_8(p, q)$ 等于它们之间最短的 8 通路的长度。像素间距离与通路示意图如图 3-7 所示。

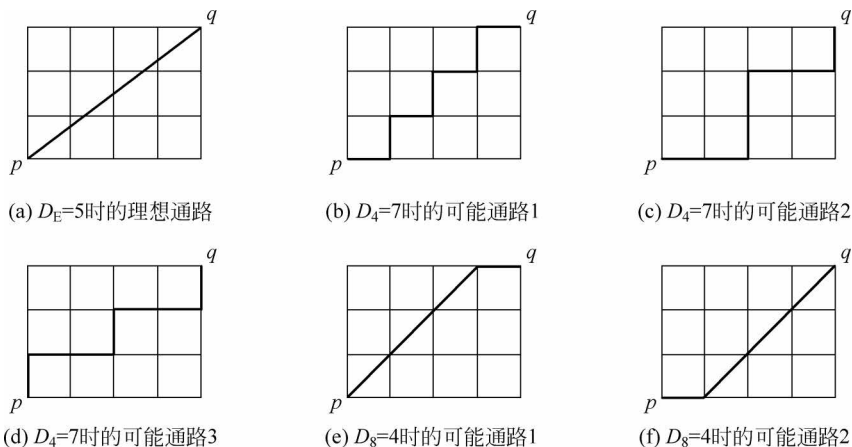


图 3-7 像素间距离与通路示意图

像素点之间的混合距离 D_m 同样刻画了像素在空间的接近程度,其大小不仅与像素的坐标有关,还与像素本身及其邻近像素的属性值有关。两个像素点 p 和 q 之间的混合距离 $D_m(p, q)$ 等于它们之间满足混合连通的通路长度。

例 3.3 混合距离示例。一张 3×3 的二值图像模板如图 3-8(a)所示,相似准则 $V = \{1\}$,像素 p 和 q 的灰度值均取 1。试求:当像素 s 和 t 的灰度值如图 3-8(b)、图 3-8(c)、图 3-8(d)和图 3-8(e)取值时,像素 p 和 q 之间的混合距离 $D_m(p, q)$ 。

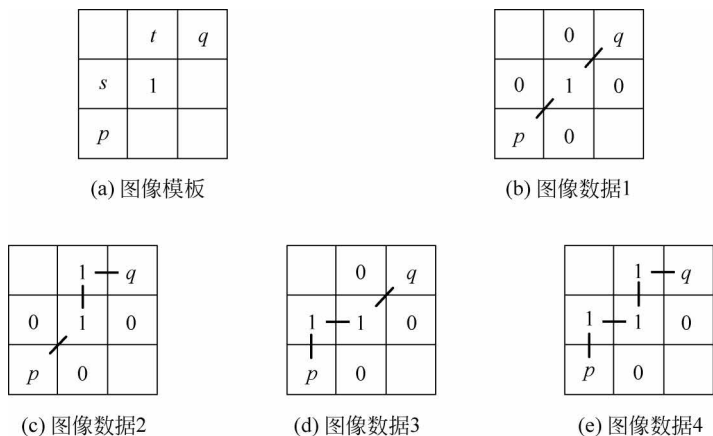


图 3-8 混合距离示例

解: 当像素 s 和 t 的灰度值如图 3-8(b)取值时,满足混合连通的通路如图所示, $D_m(p, q) = 2$ 。

当像素 s 和 t 的灰度值如图 3-8(c)取值时,满足混合连通的通路如图所示, $D_m(p, q) = 3$ 。

当像素 s 和 t 的灰度值如图 3-8(d)取值时,满足混合连通的通路如图所示, $D_m(p, q) = 3$ 。

当像素 s 和 t 的灰度值如图 3-8(e)取值时,满足混合连通的通路如图所示, $D_m(p, q) = 4$ 。

3.3 几何变换

几何变换指将图像的几何信息进行变换来获取新图像的变换方法,包括平移变换、放缩变换、旋转变换、镜像变换、剪切变换、透视变换等。二维图像和三维图像的几何变换原理基本相同,本节主要基于三维图像几何变换的原理进行介绍。

首先讨论坐标的齐次表示问题。齐次坐标是指把一个 n 维向量用一个 $n+1$ 维向量来表示。设空间中一个点的笛卡儿坐标为 (x, y, z) ,则其对应的齐次坐标为 (Hx, Hy, Hz, H) ;当 $H=1$ 时,坐标 $(x, y, z, 1)$ 称为坐标 (x, y, z) 对应的规范化齐次坐标。齐次坐标的表示方式是不唯一的,引入齐次坐标的主要目的是合并矩阵运算中的乘法和加法,它提供了一种利用矩阵运算把二维、三维甚至高维空间中的一个点集从一个坐标系变换到另一个坐标系的有效方法。

几何变换的本质是矩阵运算。设空间中一个点的笛卡儿坐标为 (x, y, z) ,基于某个条件将其变换到新的坐标 (x', y', z') ,则几何变换公式可表示为如式(3-17)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{A}\mathbf{P} \quad (3-17)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; \mathbf{A} 称为几何变换矩阵, 它唯一地确定了几何变换的结果。

图像由像素点构成, 对单个点的变换可以推广到对 m 个点的变换, 从而实现对图像的几何变换。设 m 个点的笛卡儿坐标如式(3-18)所示, 该式可表示为如式(3-19)所示的矩阵形式, 基于某个条件将其变换到如式(3-20)所示的新的坐标, 则几何变换公式可表示为如式(3-21)所示的矩阵形式:

$$\mathbf{P}_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix}, \quad \dots, \quad \mathbf{P}_m = \begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix} \quad (3-18)$$

$$\mathbf{P} = \begin{bmatrix} x_1 & x_2 & \dots & x_m \\ y_1 & y_2 & \dots & y_m \\ z_1 & z_2 & \dots & z_m \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (3-19)$$

$$\mathbf{P}' = \begin{bmatrix} x'_1 & x'_2 & \dots & x'_m \\ y'_1 & y'_2 & \dots & y'_m \\ z'_1 & z'_2 & \dots & z'_m \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (3-20)$$

$$\mathbf{P}' = \begin{bmatrix} x'_1 & x'_2 & \dots & x'_m \\ y'_1 & y'_2 & \dots & y'_m \\ z'_1 & z'_2 & \dots & z'_m \\ 1 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \dots & x_m \\ y_1 & y_2 & \dots & y_m \\ z_1 & z_2 & \dots & z_m \\ 1 & 1 & \dots & 1 \end{bmatrix} = \mathbf{A}\mathbf{P} \quad (3-21)$$

式中, \mathbf{A} 称为几何变换矩阵, 它唯一地确定了几何变换的结果。

3.3.1 平移变换

平移变换(Translation Transformation)是一种刚体变换(Rigid-Body Transformation), 指将图像沿某方向平移来获取新图像的变换方法。设空间中一个点的笛卡儿坐标为 (x, y, z) , 基于平移向量 (a, b, c) 将其平移到新的坐标 (x', y', z') , 则平移变换公式可表示为如式(3-22)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{T}\mathbf{P} = \begin{bmatrix} x + a \\ y + b \\ z + c \\ 1 \end{bmatrix} \quad (3-22)$$

式中, $\mathbf{P}=(x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}'=(x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; a, b, c 称为平移系数, 它们共同构成了平移向量 (a, b, c) ; \mathbf{T} 称为平移变换矩阵。

平移变换的效果图如图 3-9 所示。

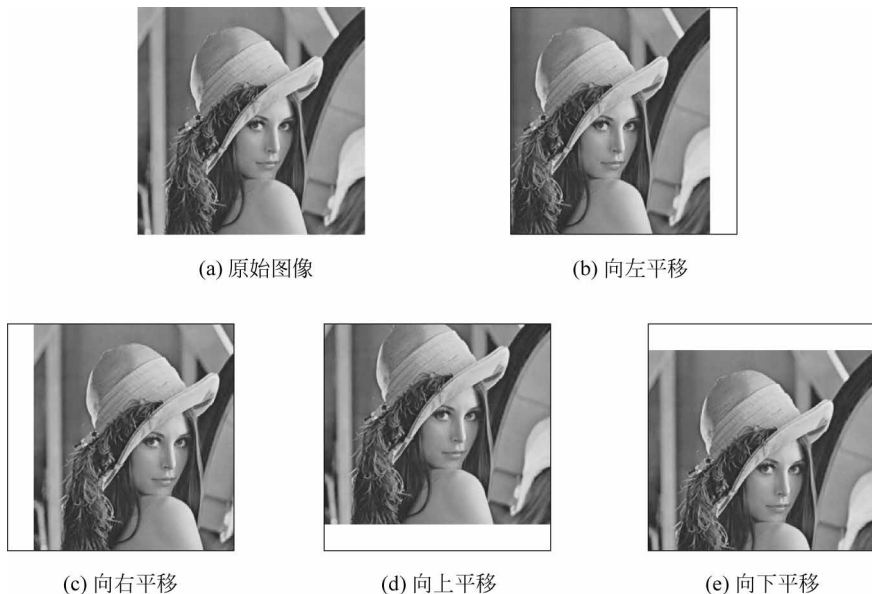


图 3-9 平移变换

3.3.2 放缩变换

放缩变换 (Scale Transformation) 也称为尺度变换, 指将图像在某方向按比例放缩来获取新图像的变换方法。放缩变换改变了图像的尺寸, 即改变了图像像素点间的距离。放缩变换一般沿坐标轴方向进行, 也可分解为沿坐标轴方向进行。

设空间中一个点的笛卡儿坐标为 (x, y, z) , 基于放缩向量 (a, b, c) 将其放缩到新的坐标 (x', y', z') , 则放缩变换公式可表示为如式 (3-23) 所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{S}\mathbf{P} = \begin{bmatrix} ax \\ by \\ cz \\ 1 \end{bmatrix} \quad (3-23)$$

式中, $\mathbf{P}=(x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}'=(x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; a, b, c 称为放缩系数, 它们共同构成了放缩向量 (a, b, c) ; 当 $a=b=c$ 时, 称为尺度放缩, 否则, 称为拉伸放缩; \mathbf{S} 称为放缩变换矩阵。

需要注意的是, 当放缩系数 a, b, c 不为整数时, 原始图像中某些像素放缩后的坐标可能不为整数, 导致变换后的图像中出现“孔洞”现象, 此时, 需要经过取整或插值等操作来进行失真校正。未经失真校正的放缩变换效果图如图 3-10 所示, 经过失真校正的放缩变换效果图如图 3-11 所示。



图 3-10 未经失真校正的放缩变换



图 3-11 经过失真校正的放缩变换

```

% F3_11.m

I1 = imread('lena.bmp');
I2 = imresize(I1,0.5);
I3 = imresize(I1,1.5);

figure, imshow(I1), xlabel('(a) 原图像');
figure, imshow(I2), xlabel('(b) 缩小后的图像');
figure, imshow(I3), xlabel('(c) 放大后的图像');
    
```

3.3.3 旋转变换

旋转变换(Rotation Transformation)是一种刚体变换(Rigid-Body Transformation),指将图像以某点为轴进行旋转来获取新图像的变换方法。首先考虑 2D 平面上一个点绕原点的旋转。设平面上一个点的笛卡儿坐标为 (x_0, y_0) , 将该点绕原点 O 顺时针旋转角度 a 后到达新的坐标 (x_1, y_1) , 则旋转变换示意图如图 3-12 所示。其中, r 为该点到原点 O 的距离; b 为旋转前 r 与 x 轴之间的夹角。

旋转前,式(3-24)成立:

$$\begin{aligned} x_0 &= r \cos b \\ y_0 &= r \sin b \end{aligned} \tag{3-24}$$

旋转后,式(3-25)成立:

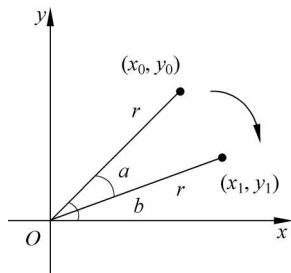


图 3-12 2D 旋转变换示意图

$$\begin{cases} x_1 = r \cos(b - a) = r \cos b \cos a + r \sin b \sin a = x_0 \cos a + y_0 \sin a \\ y_1 = r \sin(b - a) = r \sin b \cos a - r \cos b \sin a = -x_0 \sin a + y_0 \cos a \end{cases} \quad (3-25)$$

式(3-25)可表示为如式(3-26)所示的矩阵形式:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos a & \sin a \\ -\sin a & \cos a \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (3-26)$$

最简单的 3D 旋转变换是一个点绕坐标轴的旋转,包括分别绕 Z、Y、X 坐标轴旋转。

1. 一个点绕 Z 坐标轴旋转

设空间中一个点的笛卡儿坐标为 (x, y, z) , 将该点绕 Z 坐标轴旋转 γ 角度到达新的坐标 (x', y', z') , 则旋转变换公式可表示为如式(3-27)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \gamma & \sin \gamma & 0 & 0 \\ -\sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{R}_\gamma \mathbf{P} = \begin{bmatrix} x \cos \gamma + y \sin \gamma \\ -x \sin \gamma + y \cos \gamma \\ z \\ 1 \end{bmatrix} \quad (3-27)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; γ 为该点绕 Z 坐标轴旋转的角度, 定义为在右手坐标系下从旋转轴正向看原点是顺时针的; \mathbf{R}_γ 称为旋转变换矩阵。

2. 一个点绕 Y 坐标轴旋转

设空间中一个点的笛卡儿坐标为 (x, y, z) , 将该点绕 Y 坐标轴旋转 β 角度到达新的坐标 (x', y', z') , 则旋转变换公式可表示为如式(3-28)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{R}_\beta \mathbf{P} = \begin{bmatrix} x \cos \beta + z \sin \beta \\ y \\ -x \sin \beta + z \cos \beta \\ 1 \end{bmatrix} \quad (3-28)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; β 为该点绕 Y 坐标轴旋转的角度, 定义为在右手坐标系下从旋转轴正向看原点是顺时针的; \mathbf{R}_β 称为旋转变换矩阵。

3. 一个点绕 X 坐标轴旋转

设空间中一个点的笛卡儿坐标为 (x, y, z) , 将该点绕 X 坐标轴旋转 α 角度到达新的坐标 (x', y', z') , 则旋转变换公式可表示为如式(3-29)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{R}_\alpha \mathbf{P} = \begin{bmatrix} x \\ y \cos \alpha + z \sin \alpha \\ -y \sin \alpha + z \cos \alpha \\ 1 \end{bmatrix} \quad (3-29)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; α 为该点绕 X 坐标轴旋转的角度, 定义为在右手坐标系下从旋转轴正向看原点是顺时针的; \mathbf{R}_α 称为旋转变换矩阵。

需要注意的是, 原始图像中某些像素旋转后的坐标可能不为整数, 导致变换后的图像中出现“孔洞”现象, 此时, 需要经过取整或插值等操作来进行失真校正。未经失真校正的旋转

变换效果图如图 3-13 所示,经过失真校正的旋转变换效果图如图 3-14 所示。



图 3-13 未经失真校正的旋转变换



图 3-14 经过失真校正的旋转变换

```
% F3_14.m

I1 = imread('lena.bmp');
I2 = imrotate(I1, -30);
I3 = imrotate(I1,30);

subplot(1,3,1), imshow(I1), xlabel('(a) 原始图像');
subplot(1,3,2), imshow(I2), xlabel('(b) 顺时针旋转');
subplot(1,3,3), imshow(I3), xlabel('(c) 逆时针旋转');
```

一般的 3D 旋转变换是一个点绕任意一个中心点旋转,此时的情况相对复杂。设空间中一个点 P 绕一个中心点 C 旋转可由三个变换的级联来实现:第一个变换将中心点 C 平移到坐标原点,第二个变换将点 P 绕坐标原点旋转,第三个变换将中心点 C 平移回原来的初始位置。

3.3.4 镜像变换

镜像变换(Mirror Transformation)是一种刚体变换(Rigid-Body Transformation),包括水平镜像和垂直镜像。

1. 水平镜像

水平镜像指将图像左半部分和右半部分以图像垂直中轴线为中心进行镜像对换来获取新图像的变换方法。设空间中一个点的笛卡儿坐标为 (x, y, z) ,将其水平镜像到新的坐标 (x', y', z') ,则水平镜像变换公式可表示为如式(3-30)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & w \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{M}_x \mathbf{P} = \begin{bmatrix} -x + w \\ y \\ z \\ 1 \end{bmatrix} \quad (3-30)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; w 为图像的宽度; \mathbf{M}_x 称为水平镜像变换矩阵。

2. 垂直镜像

垂直镜像指将图像上半部分和下半部分以图像水平中轴线为中心进行镜像对换来获取新图像的变换方法。设空间中一个点的笛卡儿坐标为 (x, y, z) , 将其垂直镜像到新的坐标 (x', y', z') , 则垂直镜像变换公式可表示为如式(3-31)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{M}_y \mathbf{P} = \begin{bmatrix} x \\ -y + h \\ z \\ 1 \end{bmatrix} \quad (3-31)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; h 为图像的高度; \mathbf{M}_y 称为垂直镜像变换矩阵。

镜像变换的效果图如图 3-15 所示。



图 3-15 镜像变换

```
% F3_15.m

I = imread('lena.bmp');
I1 = flipdim(I,2);
I2 = flipdim(I,1);

subplot(1,3,1), imshow(I), xlabel('(a) 原始图像');
subplot(1,3,2), imshow(I1), xlabel('(b) 水平镜像图像');
subplot(1,3,3), imshow(I2), xlabel('(c) 垂直镜像图像');
```

3.3.5 剪切变换

剪切变换 (Shear Transformation) 也称为错切变换, 刻画了类似四边形不稳定性的性质, 包括水平剪切和垂直剪切。

1. 水平剪切

水平剪切指将图像一条水平边固定, 并沿水平方向拉长图像来获取新图像的变换方法。

设空间中一个点的笛卡儿坐标为 (x, y, z) ,将其水平剪切到新的坐标 (x', y', z') ,则水平剪切变换公式可表示为如式(3-32)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{H}_x \mathbf{P} = \begin{bmatrix} x \\ ax + y \\ z \\ 1 \end{bmatrix} \quad (3-32)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; a 为水平剪切系数; \mathbf{H}_x 称为水平剪切变换矩阵。

2. 垂直剪切

垂直剪切指将图像一条垂直边固定,并沿垂直方向拉长图像来获取新图像的变换方法。设空间中一个点的笛卡儿坐标为 (x, y, z) ,将其水平剪切到新的坐标 (x', y', z') ,则水平剪切变换公式可表示为如式(3-33)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & b & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{H}_y \mathbf{P} = \begin{bmatrix} x + by \\ y \\ z \\ 1 \end{bmatrix} \quad (3-33)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; b 为垂直剪切系数; \mathbf{H}_y 称为垂直剪切变换矩阵。

剪切变换的效果图如图 3-16 所示。



图 3-16 剪切变换

```
% F3_16.m

I = imread('lena.bmp');

T = [ 1   0   0
      0.5 1   0
      0   0   1]
tform = maketform('affine',T);
J1 = imtransform(I,tform);

T = [ 1  0.5  0
      0  1    0
      0  0    1]
tform = maketform('affine',T);
J2 = imtransform(I,tform);
```

```

T = [ 1   0.5  0
      0.5  1   0
      0   0   1]
tform = maketform('affine',T);
J3 = imtransform(I,tform);

subplot(2,2,1),imshow(I),xlabel('(a) 原始图像');
subplot(2,2,2),imshow(J1),xlabel('(b) 水平剪切');
subplot(2,2,3),imshow(J2),xlabel('(c) 垂直剪切');
subplot(2,2,4),imshow(J3),xlabel('(d) 水平垂直剪切');

```

3.3.6 透视变换

透视变换(Perspective Transformation)指利用透视中心、像点、目标点三点共线的条件,按透视旋转定律使透视面绕透视轴旋转某一角度,破坏原有的投影光线束,仍能保持透视面上投影几何图形不变的变换方法。透视变换原理如图 3-17 所示。

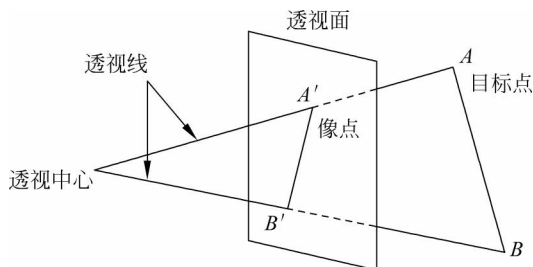


图 3-17 透视变换原理

设空间中一个点的笛卡儿坐标为 (x, y, z) ,将其透视变换到新的坐标 (x', y', z') ,则透视变换公式可表示为如式(3-34)所示的矩阵形式:

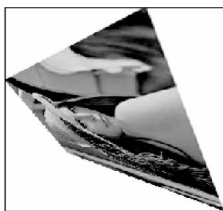
$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{E}\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ ax + by + cz + 1 \end{bmatrix} \quad (3-34)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; a, b, c 为透视变换系数,透视面为 $ax + by + cz = 0$; \mathbf{E} 称为透视变换矩阵。

透视变换的效果图如图 3-18 所示。



(a) 原始图像



(b) 透视图像1



(c) 透视图像2

图 3-18 透视变换


```

% F3_18.m

I = imread('lena.bmp');
udata = [0 1];
vdata = [0 1]
T = [ 0 0
      1 0
      1 1
      0 1]
T1 = [-4 2
      -8 -3
      1 -8
      6 6]
T2 = [ 6 6
      1 -8
      -8 -3
      -4 2]

tform = maketform('projective',T,T1);
[B1,xdata,ydata] = imtransform(I,tform,'bicubic','udata',udata,'vdata',vdata,'size',size(I),
'fill',255);

tform = maketform('projective',T,T2);
[B2,xdata,ydata] = imtransform(I,tform,'bicubic','udata',udata,'vdata',vdata,'size',size(I),
'fill',255);

subplot(1,3,1),imshow(I),axis on,xlabel('(a) 原始图像');
subplot(1,3,2),imshow(B1),axis on,xlabel('(b) 透视图像 1');
subplot(1,3,3),imshow(B2),axis on,xlabel('(c) 透视图像 2');

```

3.3.7 反变换

几何变换的反变换是指执行与对应几何变换相反操作的变换。许多几何变换都有对应的反变换,具体描述如下。

1. 平移变换的反变换

设空间中一个点的笛卡儿坐标为 (x, y, z) ,基于平移向量 (a, b, c) 已被平移到新的坐标 (x', y', z') ,则平移变换的反变换公式可表示为如式(3-35)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -a \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{T}^{-1} \mathbf{P}' = \begin{bmatrix} x' - a \\ y' - b \\ z' - c \\ 1 \end{bmatrix} \quad (3-35)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; $-a$ 、 $-b$ 、 $-c$ 称为反平移系数,它们共同构成了反平移向量 $(-a, -b, -c)$; \mathbf{T}^{-1} 称为平移变换的反变换矩阵。

2. 放缩变换的反变换

设空间中一个点的笛卡儿坐标为 (x, y, z) ,基于放缩向量 (a, b, c) 已被放缩到新的坐

标 (x', y', z') , 则放缩变换的反变换公式可表示为如式(3-36)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{a} & 0 & 0 & 0 \\ 0 & \frac{1}{b} & 0 & 0 \\ 0 & 0 & \frac{1}{c} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{S}^{-1} \mathbf{P}' = \begin{bmatrix} x' \\ a \\ y' \\ b \\ z' \\ c \\ 1 \end{bmatrix} \quad (3-36)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; $1/a, 1/b, 1/c$ 称为反放缩系数, 它们共同构成了反放缩向量 $(1/a, 1/b, 1/c)$; \mathbf{S}^{-1} 称为放缩变换的反变换矩阵。

3. 旋转变换的反变换

一个点绕 Z 坐标轴旋转的反变换可定义为: 设空间中一个点的笛卡儿坐标为 (x, y, z) , 该点绕 Z 坐标轴旋转 γ 角度后已被旋转到新的坐标 (x', y', z') , 则旋转变换的反变换公式可表示为如式(3-37)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(-\gamma) & \sin(-\gamma) & 0 & 0 \\ -\sin(-\gamma) & \cos(-\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{R}_\gamma^{-1} \mathbf{P}' = \begin{bmatrix} x' \cos \gamma - y' \sin \gamma \\ x' \sin \gamma + y' \cos \gamma \\ z' \\ 1 \end{bmatrix} \quad (3-37)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; $-\gamma$ 为该点绕 Z 坐标轴旋转的角度, 定义为在右手坐标系下从旋转轴正向看原点是顺时针的; \mathbf{R}_γ^{-1} 称为旋转变换的反变换矩阵。

一个点绕 Y 坐标轴旋转的反变换可定义为: 设空间中一个点的笛卡儿坐标为 (x, y, z) , 该点绕 Y 坐标轴旋转 β 角度后已被旋转到新的坐标 (x', y', z') , 则旋转变换的反变换公式可表示为如式(3-38)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(-\beta) & 0 & \sin(-\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(-\beta) & 0 & \cos(-\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{R}_\beta^{-1} \mathbf{P}' = \begin{bmatrix} x' \cos \beta - z' \sin \beta \\ y' \\ x' \sin \beta + z' \cos \beta \\ 1 \end{bmatrix} \quad (3-38)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; $-\beta$ 为该点绕 Y 坐标轴旋转的角度, 定义为在右手坐标系下从旋转轴正向看原点是顺时针的; \mathbf{R}_β^{-1} 称为旋转变换的反变换矩阵。

一个点绕 X 坐标轴旋转的反变换可定义为: 设空间中一个点的笛卡儿坐标为 (x, y, z) , 该点绕 X 坐标轴旋转 α 角度后已被旋转到新的坐标 (x', y', z') , 则旋转变换的反变换公式可表示为如式(3-39)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(-\alpha) & \sin(-\alpha) & 0 \\ 0 & -\sin(-\alpha) & \cos(-\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{R}_\alpha^{-1} \mathbf{P}' = \begin{bmatrix} x' \\ y' \cos \alpha - z' \sin \alpha \\ y' \sin \alpha + z' \cos \alpha \\ 1 \end{bmatrix} \quad (3-39)$$

式中, $\mathbf{P}=(x, y, z, 1)^{\mathrm{T}}$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}'=(x', y', z', 1)^{\mathrm{T}}$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; $-\alpha$ 为该点绕 X 坐标轴旋转的角度, 定义为在右手坐标系下从旋转轴正向看原点是顺时针的; \mathbf{R}_α^{-1} 称为旋转变换的反变换矩阵。

4. 镜像变换的反变换

水平镜像的反变换可定义为: 设空间中一个点的笛卡儿坐标为 (x, y, z) , 已被水平镜像到新的坐标 (x', y', z') , 则水平镜像变换的反变换公式可表示为如式(3-40)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & w \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{M}_x^{-1} \mathbf{P}' = \begin{bmatrix} -x' + w \\ y' \\ z' \\ 1 \end{bmatrix} \quad (3-40)$$

式中, $\mathbf{P}=(x, y, z, 1)^{\mathrm{T}}$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}'=(x', y', z', 1)^{\mathrm{T}}$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; w 为图像的宽度; \mathbf{M}_x^{-1} 称为水平镜像变换的反变换矩阵。由此可知, 水平镜像变换矩阵 \mathbf{M}_x 与水平镜像变换的反变换矩阵 \mathbf{M}_x^{-1} 相同。

垂直镜像的反变换可定义为: 设空间中一个点的笛卡儿坐标为 (x, y, z) , 已被垂直镜像到新的坐标 (x', y', z') , 则垂直镜像变换的反变换公式可表示为如式(3-41)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{M}_y^{-1} \mathbf{P}' = \begin{bmatrix} x' \\ -y' + h \\ z' \\ 1 \end{bmatrix} \quad (3-41)$$

式中, $\mathbf{P}=(x, y, z, 1)^{\mathrm{T}}$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}'=(x', y', z', 1)^{\mathrm{T}}$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; h 为图像的高度; \mathbf{M}_y^{-1} 称为垂直镜像变换的反变换矩阵。由此可知, 垂直镜像变换矩阵 \mathbf{M}_y 与垂直镜像变换的反变换矩阵 \mathbf{M}_y^{-1} 相同。

5. 剪切变换的反变换

水平剪切的反变换可定义为: 设空间中一个点的笛卡儿坐标为 (x, y, z) , 已被水平剪切到新的坐标 (x', y', z') , 则水平剪切变换的反变换公式可表示为如式(3-42)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -a & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{H}_x^{-1} \mathbf{P}' = \begin{bmatrix} x' \\ -ax' + y' \\ z' \\ 1 \end{bmatrix} \quad (3-42)$$

式中, $\mathbf{P}=(x, y, z, 1)^{\mathrm{T}}$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}'=(x', y', z', 1)^{\mathrm{T}}$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; $-a$ 为反水平剪切系数; \mathbf{H}_x 称为水平剪切变换的反变换矩阵。

垂直剪切的反变换可定义为: 设空间中一个点的笛卡儿坐标为 (x, y, z) , 已被垂直剪切到新的坐标 (x', y', z') , 则垂直剪切变换的反变换公式可表示为如式(3-43)所示的矩阵形式:

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -b & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{H}_y^{-1} \mathbf{P}' = \begin{bmatrix} x' - by' \\ y' \\ z' \\ 1 \end{bmatrix} \quad (3-43)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; $-b$ 为反垂直剪切系数; \mathbf{H}_y 称为垂直剪切变换的反变换矩阵。

3.3.8 复合变换

复合变换是指多个几何变换复合而成的变换,其变换矩阵为各个几何变换的变换矩阵相乘,如式(3-44)所示。

$$\mathbf{P}' = \mathbf{A}_1 \mathbf{A}_2 \cdots \mathbf{A}_n \mathbf{P} = \mathbf{A} \mathbf{P} \quad (3-44)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$ 为各个几何变换的变换矩阵; \mathbf{A} 称为复合变换矩阵。

例如,设空间中一个点的笛卡儿坐标为 (x, y, z) ,依次进行平移、放缩、绕 Z 轴旋转将其变换到新的坐标 (x', y', z') ,则复合变换公式可表示为如式(3-45)所示的矩阵形式:

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_2 & 0 & 0 & 0 \\ 0 & b_2 & 0 & 0 \\ 0 & 0 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_1 \\ 0 & 1 & 0 & b_1 \\ 0 & 0 & 1 & c_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \\ = \mathbf{R}_\gamma (\mathbf{S}(\mathbf{T}\mathbf{P})) = \mathbf{R}_\gamma \mathbf{S} \mathbf{T} \mathbf{P} = \mathbf{A} \mathbf{P} \quad (3-45)$$

式中, $\mathbf{P} = (x, y, z, 1)^T$ 为点 (x, y, z) 对应的规范化齐次坐标; $\mathbf{P}' = (x', y', z', 1)^T$ 为变换后点 (x', y', z') 对应的规范化齐次坐标; a_1, b_1, c_1 称为平移系数,它们共同构成了平移向量 (a_1, b_1, c_1) , \mathbf{T} 称为平移变换矩阵; a_2, b_2, c_2 称为放缩系数,它们共同构成了放缩向量 (a_2, b_2, c_2) , \mathbf{S} 称为放缩变换矩阵; γ 为该点绕 Z 坐标轴旋转的角度,定义为在右手坐标系下从旋转轴正向看原点是顺时针的, \mathbf{R}_γ 称为旋转变换矩阵; \mathbf{A} 称为复合变换矩阵。

例 3.4 平面上一个点的坐标为 $(1, 1)$, 基于平移向量 $(1, -1)$ 依次进行平移、顺时针旋转 45° 和反平移变换, 试求该点变换后的坐标。

解: 该变换是一个复合变换, 求解过程为

$$\mathbf{P}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \mathbf{T}^{-1} \mathbf{R}_\gamma \mathbf{T} \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & -a \\ 0 & 1 & 0 & -b \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \\ = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & -a \\ -\sin\gamma & \cos\gamma & 0 & -b \\ 0 & 0 & 1 & -c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & a\cos\gamma + b\sin\gamma - a \\ -\sin\gamma & \cos\gamma & 0 & -a\sin\gamma + b\cos\gamma - b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{aligned}
 &= \begin{bmatrix} \cos 45 & \sin 45 & 0 & \cos 45 - \sin 45 - 1 \\ -\sin 45 & \cos 45 & 0 & -\sin 45 - \cos 45 + 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & -1 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 & 1 - \sqrt{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} \sqrt{2} - 1 \\ 1 - \sqrt{2} \\ 0 \\ 1 \end{bmatrix}
 \end{aligned}$$

故坐标为(1,1)的点变换后的坐标为 $(\sqrt{2}-1, 1-\sqrt{2})$ 。

3.4 几何失真校正

几何失真是指图像在获取或显示过程中产生的畸变,也称几何畸变。例如,使用长焦镜头或使用变焦镜头的长焦端时容易产生枕形失真(Pincushion Distortion),使用广角镜头或使用变焦镜头的广角端时容易产生桶形失真(Barrel Distortion),使用广角镜头还容易产生透视失真(Perspective Distortion)等,如图 3-19 所示。

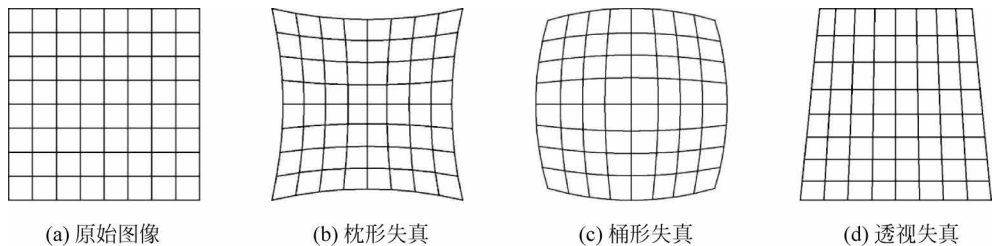


图 3-19 几何失真

几何失真校正是指将存在几何失真的畸变图像校正成为无几何失真的原始图像,通常以一张基准图像或一组基准点为基准去校正畸变图像。基准图像通常由没有畸变或畸变较小的摄像系统获得,畸变图像通常由畸变较大的摄像系统获得。

几何失真校正的校正方法包括直接校正法和间接校正法。校正步骤包括空间变换和灰度插值。

3.4.1 直接校正法

直接校正法也称为向前映射法,它首先由空间变换函数 $h_1(x, y)$ 、 $h_2(x, y)$ 推导出反变换函数 $h'_1(x', y')$ 、 $h'_2(x', y')$,然后依次计算出每一个畸变图像像素坐标 (x', y') 对应的校正坐标 (x, y) ;但 (x, y) 一般不为整数,无法直接将畸变坐标 (x', y') 处的灰度值赋值给校正坐标 (x, y) ,而是将 (x', y') 处的灰度值分配给校正坐标 (x, y) 周围的四个像素,据此获

得校正图像。直接校正法原理图如图 3-20 所示。

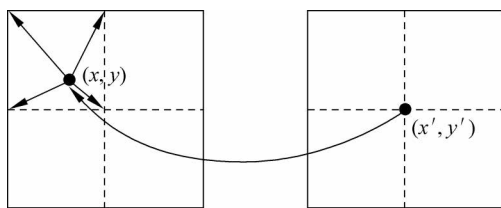


图 3-20 直接校正法原理图

直接校正法存在很多不足：

- (1) 畸变图像像素可能会被映射到校正图像之外,因此计算效率不高;
- (2) 校正图像像素的灰度值由畸变图像像素的贡献之和决定,导致分配时存在较多的寻址,特别是在高阶插值时;
- (3) 生成校正图像的像素分布不规则,容易出现像素挤压、疏密不均等现象,需要通过灰度插值方法生成规则的栅格图像。

3.4.2 间接校正法

间接校正法也称为向后映射法,它假设经过校正的图像像素坐标在基准坐标系中为等距网格的交叉点,从网格交叉点的坐标 (x, y) 出发计算出畸变图像上的对应坐标 (x', y') ;但 (x', y') 一般不为整数,无法直接将畸变坐标 (x', y') 处的灰度值赋值给校正坐标 (x, y) ,而是将 (x', y') 周围像素的灰度值进行插值以得到校正坐标 (x, y) 处的灰度值,据此获得校正图像。间接校正法原理图如图 3-21 所示。

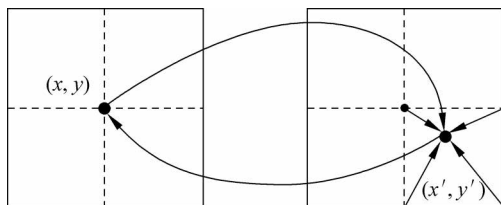


图 3-21 间接校正法原理图

间接校正法生成的校正图像由逐个像素通过一步插值得到,灰度插值效率较高,在实际应用中被广泛使用。间接校正法的步骤包括空间变换和灰度插值,如下式所示:

$$\begin{cases} x' = h_1(x, y) \\ y' = h_2(x, y) \\ f(x, y) = g(x', y') \end{cases} \quad (3-46)$$

式中, (x, y) 为无失真坐标系中像素的坐标; (x', y') 为失真坐标系中像素的坐标; $h_1(x, y)$ 、 $h_2(x, y)$ 为无失真坐标系向失真坐标系变换的空间变换函数; $f(x, y)$ 为无失真的原始图像; $g(x', y')$ 为存在失真的畸变图像。

3.4.3 空间变换

空间变换是指对畸变图像像素坐标位置进行重新排列以恢复原始图像像素坐标位置的

空间关系。空间变换的关键是确定空间变换函数,该函数可以通过先验知识得到,但实际上往往难以求得,通常需要采用后验校正的方法来确定。后验校正方法的基本思想是通过控制点(即基准图像上的正确像素点和畸变图像上的失真像素点)之间的对应关系来近似空间变换函数,一般采用如下式所示的多项式拟合的形式:

$$\begin{cases} x' = h_1(x, y) = \sum_{i=0}^N \sum_{j=0}^{N-i} a_{ij} x^i y^j \\ y' = h_2(x, y) = \sum_{i=0}^N \sum_{j=0}^{N-i} b_{ij} x^i y^j \end{cases} \quad (3-47)$$

式中, (x, y) 为无失真坐标系中像素的坐标; (x', y') 为失真坐标系中像素的坐标; $h_1(x, y)$ 、 $h_2(x, y)$ 为无失真坐标系向失真坐标系变换的空间变换函数; a_{ij} 、 b_{ij} 为拟合多项式的各项待定系数; N 为拟合多项式的次数。

当 $N=1$ 时,空间变换函数为线性拟合,可粗略地近似几何畸变,如式(3-48)所示;当 $N=2$ 时,空间变换函数为二次拟合,可较精确地近似几何畸变,如式(3-49)所示;当 $N>3$ 时,空间变换函数为高阶拟合。一般地,二次或高阶拟合被用来补偿由实际图像系统造成的空间失真。

$$\begin{cases} x' = a_0 + a_1 x + a_2 y \\ y' = b_0 + b_1 x + b_2 y \end{cases} \quad (3-48)$$

$$\begin{cases} x' = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 \\ y' = b_0 + b_1 x + b_2 y + b_3 xy + b_4 x^2 + b_5 y^2 \end{cases} \quad (3-49)$$

为了求解方程组,可根据待定系数的个数确定基准图像和畸变图像上控制点的对数,这些控制点构成了图像中的一个多边形区域;将图像分成一系列覆盖全图的多边形区域的集合,对每个区域都找到足够的控制点,便可计算出待拟合多项式的系数,确定空间变换函数,从而复原原始图像。失真图像和校正图像四边形区域对应点示意图如图 3-22 所示。

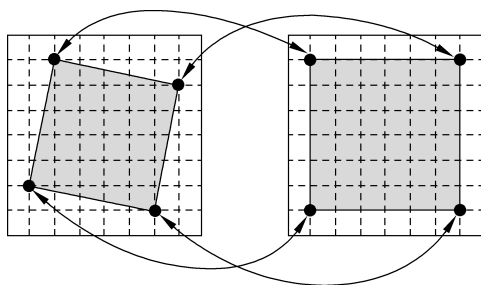


图 3-22 失真图像和校正图像四边形区域对应点示意图

例 3.5 确定线性拟合时的空间变换函数。

解: 采用线性拟合时,确定空间变换函数就是要确定 a_0 、 a_1 、 a_2 、 b_0 、 b_1 、 b_2 共 6 个多项式系数,因此需要建立 6 个方程联立求解。为此,找出基准图像和畸变图像上相对应的 3 对控制点 (x_1, y_1) 、 (x_2, y_2) 、 (x_3, y_3) 和 (x'_1, y'_1) 、 (x'_2, y'_2) 、 (x'_3, y'_3) , 分别代入式(3-48)后得到式(3-50)~式(3-52):

$$\begin{cases} x'_1 = a_0 + a_1 x_1 + a_2 y_1 \\ y'_1 = b_0 + b_1 x_1 + b_2 y_1 \end{cases} \quad (3-50)$$

$$\begin{cases} x'_2 = a_0 + a_1 x_2 + a_2 y_2 \\ y'_2 = b_0 + b_1 x_2 + b_2 y_2 \end{cases} \quad (3-51)$$

$$\begin{cases} x'_3 = a_0 + a_1 x_3 + a_2 y_3 \\ y'_3 = b_0 + b_1 x_3 + b_2 y_3 \end{cases} \quad (3-52)$$

写成矩阵形式后得到式(3-53)和式(3-54):

$$\begin{cases} x'_1 = a_0 + a_1 x_1 + a_2 y_1 \\ x'_2 = a_0 + a_1 x_2 + a_2 y_2 \\ x'_3 = a_0 + a_1 x_3 + a_2 y_3 \end{cases} \Rightarrow \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \quad (3-53)$$

$$\begin{cases} y'_1 = b_0 + b_1 x_1 + b_2 y_1 \\ y'_2 = b_0 + b_1 x_2 + b_2 y_2 \\ y'_3 = b_0 + b_1 x_3 + b_2 y_3 \end{cases} \Rightarrow \begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \quad (3-54)$$

通过联立方程组求解或矩阵求逆运算,解出 $a_0, a_1, a_2, b_0, b_1, b_2$ 共6个多项式系数,即可确定空间变换函数 $h_1(x, y)$ 和 $h_2(x, y)$ 。空间变换函数一旦确定,即可复原原始图像中此三点连线所包围的三角形区域内的各像素点;以此类推,每三对控制点一组重复进行上述操作,即可实现全图像的空间校正。

例 3.6 确定二次拟合时的空间变换函数。

解: 采用二次拟合时,确定空间变换函数就是要确定 $a_0, a_1, a_2, a_3, a_4, a_5, b_0, b_1, b_2, b_3, b_4, b_5$ 共12个多项式系数,因此需要建立12个方程联立求解。为此,找出原始图像和畸变图像上相对应的6对控制点 $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6)$ 和 $(x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3), (x'_4, y'_4), (x'_5, y'_5), (x'_6, y'_6)$,分别代入式(3-49)后得到式(3-55)~式(3-60):

$$\begin{cases} x'_1 = a_0 + a_1 x_1 + a_2 y_1 + a_3 x_1 y_1 + a_4 x_1^2 + a_5 y_1^2 \\ y'_1 = b_0 + b_1 x_1 + b_2 y_1 + b_3 x_1 y_1 + b_4 x_1^2 + b_5 y_1^2 \end{cases} \quad (3-55)$$

$$\begin{cases} x'_2 = a_0 + a_1 x_2 + a_2 y_2 + a_3 x_2 y_2 + a_4 x_2^2 + a_5 y_2^2 \\ y'_2 = b_0 + b_1 x_2 + b_2 y_2 + b_3 x_2 y_2 + b_4 x_2^2 + b_5 y_2^2 \end{cases} \quad (3-56)$$

$$\begin{cases} x'_3 = a_0 + a_1 x_3 + a_2 y_3 + a_3 x_3 y_3 + a_4 x_3^2 + a_5 y_3^2 \\ y'_3 = b_0 + b_1 x_3 + b_2 y_3 + b_3 x_3 y_3 + b_4 x_3^2 + b_5 y_3^2 \end{cases} \quad (3-57)$$

$$\begin{cases} x'_4 = a_0 + a_1 x_4 + a_2 y_4 + a_3 x_4 y_4 + a_4 x_4^2 + a_5 y_4^2 \\ y'_4 = b_0 + b_1 x_4 + b_2 y_4 + b_3 x_4 y_4 + b_4 x_4^2 + b_5 y_4^2 \end{cases} \quad (3-58)$$

$$\begin{cases} x'_5 = a_0 + a_1 x_5 + a_2 y_5 + a_3 x_5 y_5 + a_4 x_5^2 + a_5 y_5^2 \\ y'_5 = b_0 + b_1 x_5 + b_2 y_5 + b_3 x_5 y_5 + b_4 x_5^2 + b_5 y_5^2 \end{cases} \quad (3-59)$$

$$\begin{cases} x'_6 = a_0 + a_1x_6 + a_2y_6 + a_3x_6y_6 + a_4x_6^2 + a_5y_6^2 \\ y'_6 = b_0 + b_1x_6 + b_2y_6 + b_3x_6y_6 + b_4x_6^2 + b_5y_6^2 \end{cases} \quad (3-60)$$

合并坐标方程后得到式(3-61)和式(3-62):

$$\begin{cases} x'_1 = a_0 + a_1x_1 + a_2y_1 + a_3x_1y_1 + a_4x_1^2 + a_5y_1^2 \\ x'_2 = a_0 + a_1x_2 + a_2y_2 + a_3x_2y_2 + a_4x_2^2 + a_5y_2^2 \\ x'_3 = a_0 + a_1x_3 + a_2y_3 + a_3x_3y_3 + a_4x_3^2 + a_5y_3^2 \\ x'_4 = a_0 + a_1x_4 + a_2y_4 + a_3x_4y_4 + a_4x_4^2 + a_5y_4^2 \\ x'_5 = a_0 + a_1x_5 + a_2y_5 + a_3x_5y_5 + a_4x_5^2 + a_5y_5^2 \\ x'_6 = a_0 + a_1x_6 + a_2y_6 + a_3x_6y_6 + a_4x_6^2 + a_5y_6^2 \end{cases} \quad (3-61)$$

$$\begin{cases} y'_1 = b_0 + b_1x_1 + b_2y_1 + b_3x_1y_1 + b_4x_1^2 + b_5y_1^2 \\ y'_2 = b_0 + b_1x_2 + b_2y_2 + b_3x_2y_2 + b_4x_2^2 + b_5y_2^2 \\ y'_3 = b_0 + b_1x_3 + b_2y_3 + b_3x_3y_3 + b_4x_3^2 + b_5y_3^2 \\ y'_4 = b_0 + b_1x_4 + b_2y_4 + b_3x_4y_4 + b_4x_4^2 + b_5y_4^2 \\ y'_5 = b_0 + b_1x_5 + b_2y_5 + b_3x_5y_5 + b_4x_5^2 + b_5y_5^2 \\ y'_6 = b_0 + b_1x_6 + b_2y_6 + b_3x_6y_6 + b_4x_6^2 + b_5y_6^2 \end{cases} \quad (3-62)$$

写成矩阵形式后得到式(3-63)和式(3-64):

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \\ x'_5 \\ x'_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2y_2 & x_2^2 & y_2^2 \\ 1 & x_3 & y_3 & x_3y_3 & x_3^2 & y_3^2 \\ 1 & x_4 & y_4 & x_4y_4 & x_4^2 & y_4^2 \\ 1 & x_5 & y_5 & x_5y_5 & x_5^2 & y_5^2 \\ 1 & x_6 & y_6 & x_6y_6 & x_6^2 & y_6^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} \quad (3-63)$$

$$\begin{bmatrix} y'_1 \\ y'_2 \\ y'_3 \\ y'_4 \\ y'_5 \\ y'_6 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & x_1^2 & y_1^2 \\ 1 & x_2 & y_2 & x_2y_2 & x_2^2 & y_2^2 \\ 1 & x_3 & y_3 & x_3y_3 & x_3^2 & y_3^2 \\ 1 & x_4 & y_4 & x_4y_4 & x_4^2 & y_4^2 \\ 1 & x_5 & y_5 & x_5y_5 & x_5^2 & y_5^2 \\ 1 & x_6 & y_6 & x_6y_6 & x_6^2 & y_6^2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} \quad (3-64)$$

通过联立方程组求解或矩阵求逆运算,解出 $a_0, a_1, a_2, a_3, a_4, a_5, b_0, b_1, b_2, b_3, b_4, b_5$ 共 12 个多项式系数,即可确定空间变换函数 $h_1(x, y)$ 和 $h_2(x, y)$ 。空间变换函数一旦确定,即可复原原始图像中此六点连线所包围的六边形区域内的各像素点;以此类推,每 6 对像素点一组重复进行上述操作,即可实现全图像的空间校正。

3.4.4 灰度插值

灰度插值是指对图像映射位置及其周围像素的灰度值进行插值操作以复原原始图像像素的灰度值。基于间接矫正法复原原始图像时,对原始图像中的每一个像素点 (x, y) ,计算

出畸变图像上的对应坐标 (x', y') ，基于 (x', y') 处的灰度值确定原始图像中像素点 (x, y) 的灰度值。如果计算出的畸变图像上的对应坐标 (x', y') 为整数，则原始图像对应像素点 (x, y) 的灰度值与其保持一致；如果不为整数，则需要进行灰度插值操作。常用的灰度插值方法包括最近邻插值法、双线性插值法和双三次插值法等。

1. 最近邻插值法

最近邻插值法 (Nearest Neighbor Interpolation) 也称为零阶插值，是指将距离映射位置最近的像素点的灰度值作为插值结果的方法，其原理图如图 3-23 所示。

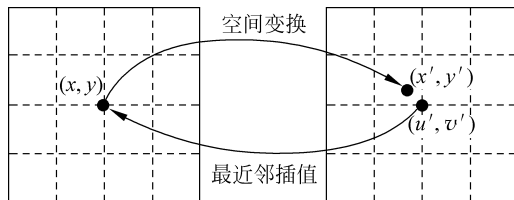


图 3-23 最近邻插值法原理图

设原始图像 $f(x, y)$ 中某像素点 (x, y) 经过变换后在畸变图像 $g(x', y')$ 上的映射位置为 (x', y') ，则最近邻插值法如下式所示：

$$f(x, y) = g(x', y') = g(u', v') \quad (3-65)$$

式中，像素点 (u', v') 为距离映射位置 (x', y') 最近的像素点，即 u', v' 满足下式条件：

$$\begin{cases} x' - \frac{1}{2} < u' < x' + \frac{1}{2} \\ y' - \frac{1}{2} < v' < y' + \frac{1}{2} \end{cases} \quad (3-66)$$

最近邻插值法简单易行，计算量小，但细微结构粗糙，容易出现块状效应。最近邻插值法的效果图如图 3-24 所示。

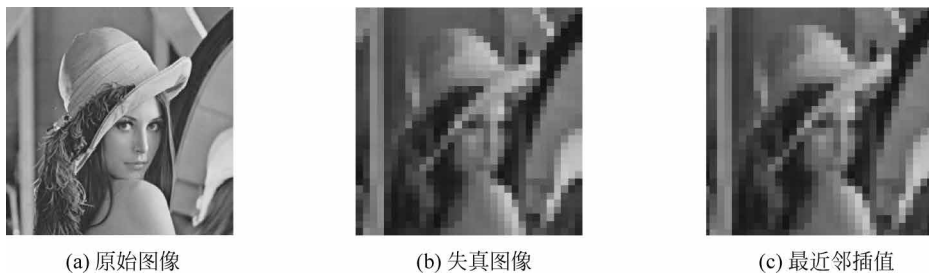


图 3-24 最近邻插值法效果图

```
% F3_24.m

I = imread('lena.bmp');
subplot(2,2,1), imshow(I), xlabel('(a) 原始图像');

times = 8;
f = imresize(I, 1/times);
subplot(2,2,2), imshow(f), xlabel('(b) 失真图像');
```

```

% 方法 1
g1 = interp2(double(f), 'nearest');
% g1 = im2uint8(mat2gray(g1));
subplot(2,2,3), imshow(uint8(g1)), xlabel('(c) 最近邻插值(方法 1)');

% 方法 2
[m,n] = size(f);
[x,y] = meshgrid(1:m,1:n);
[xi,yi] = meshgrid(1:m*times,1:n*times);
g2 = interp2(x,y,double(f),xi/times,yi/times,'nearest');
% g2 = im2uint8(mat2gray(g2));
subplot(2,2,4), imshow(uint8(g2)), xlabel('(d) 最近邻插值(方法 2)');

```

2. 双线性插值法

双线性插值法(Bilinear Interpolation)是指将映射位置周围 4 个像素点的灰度值在水平和垂直两个方向上进行插值以获取插值结果的方法,其原理图如图 3-25 所示。

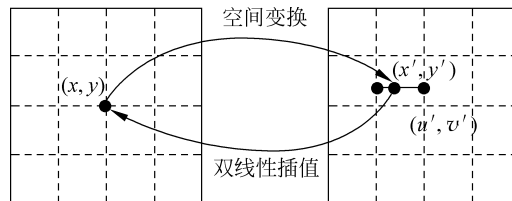


图 3-25 双线性插值法原理图

设原始图像 $f(x, y)$ 中某像素点 (x, y) 经过变换后在畸变图像 $g(x', y')$ 上的映射位置为 $G(x', y')$, 该映射位置周围 4 个像素点的坐标为 $A([x'], [y'])$ 、 $B([x'], [y'] + 1)$ 、 $C([x'] + 1, [y'])$ 和 $D([x'] + 1, [y'] + 1)$, 符号 $[]$ 表示取整, 则双线性插值法计算图如图 3-26 所示。

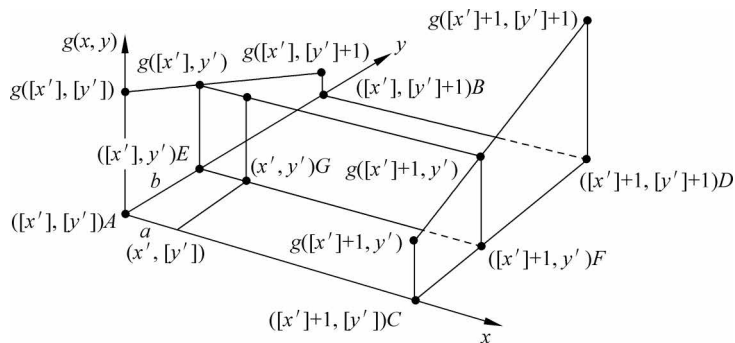


图 3-26 双线性插值法计算图

由图可知下式成立:

$$\begin{cases} a = x' - [x'] \\ b = y' - [y'] \end{cases} \quad (3-67)$$

因此, 点 E 处的灰度值 $g(E)$ 可表示为

$$g(E) = g(A) + b(g(B) - g(A)) = (1 - b)g(A) + bg(B) \quad (3-68)$$

点 F 处的灰度值 $g(F)$ 可表示为

$$g(F) = g(C) + b(g(D) - g(C)) = (1 - b)g(C) + bg(D) \quad (3-69)$$

点 G 处的灰度值 $g(G)$ 可表示为

$$\begin{aligned} g(G) &= g(E) + a(g(F) - g(E)) = (1 - a)g(E) + ag(F) \\ &= (1 - a)[(1 - b)g(A) + bg(B)] + a[(1 - b)g(C) + bg(D)] \\ &= (1 - a)(1 - b)g(A) + (1 - a)bg(B) + a(1 - b)g(C) + abg(D) \end{aligned} \quad (3-70)$$

因此,双线性插值法可表示为

$$f(x, y) = g(x', y') = g(G) \quad (3-71)$$

当 $x = [x']$ 时,有 $a = 0$,此时,双线性插值法可表示为

$$\begin{aligned} f(x, y) &= g(x', y') = g(G) = (1 - b)g(A) + bg(B) \\ &= (1 - y' + [y'])g([x'], [y']) + (y' - [y'])g([x'], [y'] + 1) \end{aligned} \quad (3-72)$$

当 $y = [y']$ 时,有 $b = 0$,此时,双线性插值法可表示为

$$\begin{aligned} f(x, y) &= g(x', y') = g(G) = (1 - a)g(A) + ag(C) \\ &= (1 - x' + [x'])g([x'], [y']) + (x' - [x'])g([x'] + 1, [y']) \end{aligned} \quad (3-73)$$

当 $x = [x'], y = [y']$ 时,有 $a = 0, b = 0$,此时,双线性插值法可表示为

$$f(x, y) = g(x', y') = g(G) = g(A) = g([x'], [y']) \quad (3-74)$$

双线性插值法具有低通滤波器的性质,它削弱了高频信息,导致图像轮廓模糊。双线性插值法的效果图如图 3-27 所示。

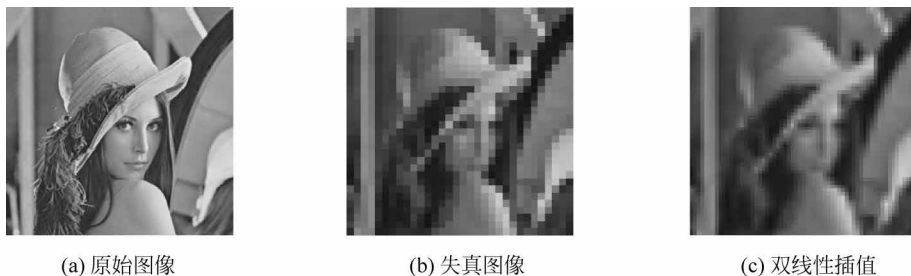


图 3-27 双线性插值法效果图

```
% F3_27.m

I = imread('lena.bmp');
subplot(2,2,1), imshow(I), xlabel('(a) 原始图像');

times = 4;
f = imresize(I, 1/times);
subplot(2,2,2), imshow(f), xlabel('(b) 失真图像');

% 方法 1
g1 = interp2(double(f), 'linear');
% g1 = im2uint8(mat2gray(g1));
subplot(2,2,3), imshow(uint8(g1)), xlabel('(c) 双线性插值(方法 1)');
```

```

% 方法 2
[m,n] = size(f);
[x,y] = meshgrid(1:m,1:n);
[xi,yi] = meshgrid(1:m*times,1:n*times);
g2 = interp2(x,y,double(f),xi/times,yi/times,'linear');
% g2 = im2uint8(mat2gray(g2));
subplot(2,2,4), imshow(uint8(g2)), xlabel('(d) 双线性插值(方法 2)');

```

3. 双三次插值法

双三次插值法(Bicubic Interpolation)是指将映射位置周围 16 个像素点的灰度值在水平和垂直两个方向上进行插值以获取插值结果的方法,其原理图如图 3-28 所示:

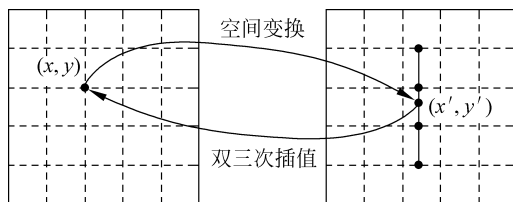


图 3-28 双三次插值法原理图

设原始图像 $f(x, y)$ 中某像素点 (x, y) 经过变换后在畸变图像 $g(x', y')$ 上的映射位置为 $G(x', y')$, 该映射位置周围 16 个像素点分别为 G_0, G_1, \dots, G_{15} , 符号 $[]$ 表示取整, 则双三次插值法计算图如图 3-29 所示。

由图可知下式成立:

$$\begin{cases} a = x' - [x'] \\ b = y' - [y'] \end{cases} \quad (3-75)$$

双三次插值法首先基于式(3-76)计算出点 A_0, A_1, A_2, A_3 处的灰度值, 再基于式(3-77)计算出最终的插值结果。

$$g(A_i) = \sum_{j=0}^3 g(G_{4i+j})s(d_y(G_{4i+j}, G)), \quad i = 0, 1, 2, 3 \quad (3-76)$$

$$g(x', y') = \sum_{i=0}^3 g(A_i)s(d_x(A_i, G)) \quad (3-77)$$

式中, 两点 $P_1(x_1, y_1)$ 和 $P_2(x_2, y_2)$ 之间的距离函数 $d()$ 定义为如式(3-78)所示; 理论上为了逼近最佳插值函数 $y = \sin x / x$, 三次多项式函数 $s(x)$ 定义如式(3-79)所示, 其函数曲线如图 3-30 所示。

$$\begin{cases} d_x(P_1, P_2) = |x_1 - x_2| \\ d_y(P_1, P_2) = |y_1 - y_2| \end{cases} \quad (3-78)$$

$$s(x) = \begin{cases} 1 - 2|x|^2 + |x|^3, & |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3, & 1 \leq |x| < 2 \\ 0, & |x| > 2 \end{cases} \quad (3-79)$$

由此可知,点 A_0, A_1, A_2, A_3 处的灰度值由式(3-80)~式(3-83)计算,它们可表示为如式(3-84)所示的矩阵形式。

$$\begin{aligned} g(A_0) &= \sum_{j=0}^3 g(G_{0+j})s(d_y(G_{0+j}, G)) \\ &= g(G_0)s(d_y(G_0, G)) + g(G_1)s(d_y(G_1, G)) + g(G_2)s(d_y(G_2, G)) + \\ &\quad g(G_3)s(d_y(G_3, G)) \\ &= g(G_0)s(1+b) + g(G_1)s(b) + g(G_2)s(1-b) + g(G_3)s(2-b) \end{aligned} \quad (3-80)$$

$$\begin{aligned} g(A_1) &= \sum_{j=0}^3 g(G_{4+j})s(d_y(G_{4+j}, G)) \\ &= g(G_4)s(d_y(G_4, G)) + g(G_5)s(d_y(G_5, G)) + g(G_6)s(d_y(G_6, G)) + \\ &\quad g(G_7)s(d_y(G_7, G)) \\ &= g(G_4)s(1+b) + g(G_5)s(b) + g(G_6)s(1-b) + g(G_7)s(2-b) \end{aligned} \quad (3-81)$$

$$\begin{aligned} g(A_2) &= \sum_{j=0}^3 g(G_{8+j})s(d_y(G_{8+j}, G)) \\ &= g(G_8)s(d_y(G_8, G)) + g(G_9)s(d_y(G_9, G)) + g(G_{10})s(d_y(G_{10}, G)) + \\ &\quad g(G_{11})s(d_y(G_{11}, G)) \\ &= g(G_8)s(1+b) + g(G_9)s(b) + g(G_{10})s(1-b) + g(G_{11})s(2-b) \end{aligned} \quad (3-82)$$

$$\begin{aligned} g(A_3) &= \sum_{j=0}^3 g(G_{12+j})s(d_y(G_{12+j}, G)) \\ &= g(G_{12})s(d_y(G_{12}, G)) + g(G_{13})s(d_y(G_{13}, G)) + g(G_{14})s(d_y(G_{14}, G)) + \\ &\quad g(G_{15})s(d_y(G_{15}, G)) \\ &= g(G_{12})s(1+b) + g(G_{13})s(b) + g(G_{14})s(1-b) + g(G_{15})s(2-b) \end{aligned} \quad (3-83)$$

$$\begin{bmatrix} g(A_0) \\ g(A_1) \\ g(A_2) \\ g(A_3) \end{bmatrix} = \begin{bmatrix} g(G_0) & g(G_1) & g(G_2) & g(G_3) \\ g(G_4) & g(G_5) & g(G_6) & g(G_7) \\ g(G_8) & g(G_9) & g(G_{10}) & g(G_{11}) \\ g(G_{12}) & g(G_{13}) & g(G_{14}) & g(G_{15}) \end{bmatrix} \begin{bmatrix} s(1+b) \\ s(b) \\ s(1-b) \\ s(2-b) \end{bmatrix} \quad (3-84)$$

最终的插值结果由式(3-85)计算,它们可表示为如式(3-86)所示的矩阵形式。

$$\begin{aligned} g(x', y') &= \sum_{i=0}^3 g(A_i)s(d_x(A_i, G)) \\ &= g(A_0)s(d_x(A_0, G)) + g(A_1)s(d_x(A_1, G)) + g(A_2)s(d_x(A_2, G)) + \\ &\quad g(A_3)s(d_x(A_3, G)) \\ &= g(A_0)s(1+a) + g(A_1)s(a) + g(A_2)s(1-a) + g(A_3)s(2-a) \end{aligned} \quad (3-85)$$

$$\mathbf{g}(x', y') = \begin{bmatrix} s(1+a) \\ s(a) \\ s(1-a) \\ s(2-a) \end{bmatrix}^T \begin{bmatrix} g(A_0) \\ g(A_1) \\ g(A_2) \\ g(A_3) \end{bmatrix} \quad (3-86)$$

因此,双三次插值法可表示为

$$\mathbf{g}(x', y') = \begin{bmatrix} s(1+a) \\ s(a) \\ s(1-a) \\ s(2-a) \end{bmatrix}^T \begin{bmatrix} g(G_0) & g(G_1) & g(G_2) & g(G_3) \\ g(G_4) & g(G_5) & g(G_6) & g(G_7) \\ g(G_8) & g(G_9) & g(G_{10}) & g(G_{11}) \\ g(G_{12}) & g(G_{13}) & g(G_{14}) & g(G_{15}) \end{bmatrix} \begin{bmatrix} s(1+b) \\ s(b) \\ s(1-b) \\ s(2-b) \end{bmatrix} = \mathbf{AGB} \quad (3-87)$$

双三次插值法计算精度高,较好地保持了图像的边缘细节,但计算量较大。双三次插值法的效果图如图 3-31 所示。

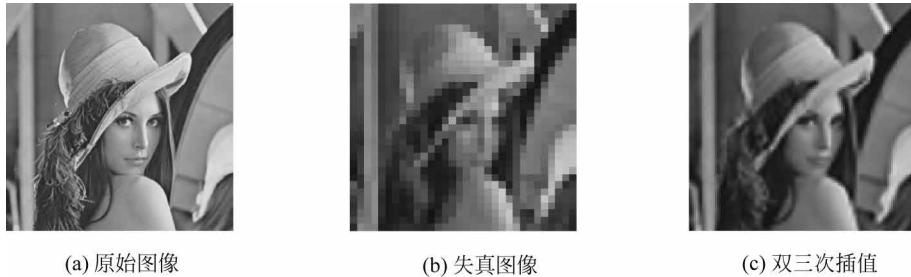


图 3-31 双三次插值法效果图

```
% F3_31.m

I = imread('lena.bmp');
subplot(2,2,1), imshow(I), xlabel('(a) 原始图像');

times = 4;
f = imresize(I, 1/times);
subplot(2,2,2), imshow(f), xlabel('(b) 失真图像');

% 方法 1
g1 = interp2(double(f), 'cubic');
% g1 = im2uint8(mat2gray(g1));
subplot(2,2,3), imshow(uint8(g1)), xlabel('(c) 双三次插值(方法 1)');

% 方法 2
[m,n] = size(f);
[x,y] = meshgrid(1:m, 1:n);
[xi,yi] = meshgrid(1:m*times, 1:n*times);
g2 = interp2(x,y, double(f), xi/times, yi/times, 'cubic');
% g2 = im2uint8(mat2gray(g2));
subplot(2,2,4), imshow(uint8(g2)), xlabel('(d) 双三次插值(方法 2)');
```

习题

- 3-1 试简述邻域的定义和类型。
- 3-2 试简述邻接的定义和类型。

3-3 试简述连接的定义和类型。

3-4 两个图像子集 P 和 Q 如图 3-32 所示,相似准则 $V = \{1\}$,试求:

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

图 3-32 图像子集 P 和 Q

- (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 m 连接。
- (3) 子集 P 和子集 Q 是否: ①4 连通; ②8 连通; ③ m 连通。
- (4) 若将子集 P 和子集 Q 以外的所有像素看成另一个子集 R ,试指出子集 P 和子集 Q 是否与子集 R : ①4 连接; ②8 连接; ③ m 连接。

3-5 图像子集如图 3-33 所示,试求:

- (1) 若相似准则 $V = \{0,1\}$,试计算 p 和 q 之间 4 通路、8 通路和 m 通路的长度;
- (2) 若相似准则 $V = \{1,2\}$,试计算 p 和 q 之间 4 通路、8 通路和 m 通路的长度。

3-6 试简述欧氏距离、城区距离和棋盘距离的几何意义。

3-7 像素之间的位置关系如图 3-34 所示,试求:

	3	1	2	1(q)
	2	2	0	2
	1	2	1	1
(p)	1	0	1	2

图 3-33 图像子集

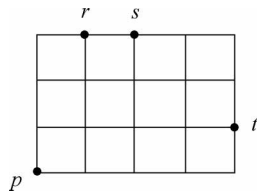


图 3-34 像素位置关系

- (1) 像素 p 和像素 r 之间的欧氏距离 D_E 、城区距离 D_4 和棋盘距离 D_8 ;
- (2) 像素 p 和像素 s 之间的欧氏距离 D_E 、城区距离 D_4 和棋盘距离 D_8 ;
- (3) 像素 p 和像素 t 之间的欧氏距离 D_E 、城区距离 D_4 和棋盘距离 D_8 。

3-8 像素之间的位置关系如图 3-35 所示,相似准则 $V = \{1\}$,试求:

				r
	0	1	0	1
	1	1	1	1
p	1	0	1	0

(a) 像素 p 和像素 r

				s
	1	1	1	1
	0	1	0	1
p	1	0	1	0

(b) 像素 p 和像素 s

				t
	1	1	1	1
	0	1	0	1
p	1	0	0	1

(c) 像素 p 和像素 t

图 3-35 像素位置关系

- (1) 像素 p 和像素 r 之间的混合距离 D_m ;
- (2) 像素 p 和像素 s 之间的混合距离 D_m ;

(3) 像素 p 和像素 t 之间的混合距离 D_m 。

3-9 试简述几何变换的分类及含义。

3-10 若平移变换矩阵 T 和尺度变换矩阵 S 如图 3-36 所示,试计算:

$$T = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) 平移变换矩阵 T

(b) 尺度变换矩阵 S

图 3-36 矩阵

(1) 对空间点 $P(1,2,3)$ 先进行平移变换、再进行尺度变换所得到的结果;

(2) 对空间点 $P(1,2,3)$ 先进行尺度变换、再进行平移变换所得到的结果。

3-11 若平移变换矩阵 T 和尺度变换矩阵 S 如图 3-37 所示,试计算:

$$T = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad S = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(a) 平移变换矩阵 T

(b) 尺度变换矩阵 S

图 3-37 矩阵

(1) 对空间点 $P(4,5,6)$ 先进行平移变换、再进行尺度变换、最后进行反平移变换所得到的结果;

(2) 对空间点 $P(4,5,6)$ 先进行尺度变换、再进行平移变换、最后进行反尺度变换所得到的结果。

3-12 试求能将图像顺时针旋转 45° 的旋转变换矩阵,并计算利用该矩阵对图像点 $P(1,0)$ 进行旋转变换后所得到的结果。

3-13 试简述几何失真校正的直接校正法的原理及不足。

3-14 试简述几何失真校正的间接校正法的原理及步骤。

3-15 试简述空间变换的概念及原理。

3-16 试简述灰度插值的概念及基于间接校正法进行灰度插值的原理。

3-17 试简述最近邻插值法、双线性插值法和双三次插值法的概念。