第六章 图像的几何变换

BACK

6.1 几何变换基础

6.1.1 概述

图像的几何变换，按照需要使图像产生大小、形状和位置的变化。

从变换的性质分，图像的几何变换有平移、比例缩放、旋转、反射和错切等基本变换，透视变换等复合变换，以及插值运算等。

除了插值运算外，常见的图像几何变换可以通过与之对应的矩阵线性变换来实现。

为了能够用统一的矩阵线性变换形式，表示和实现这些常见的图像几何变换，需要引入一种新的坐标，即齐次坐标。

6.1.2 齐次坐标

现设点$P_0(x_0, y_0)$进行平移后，移到$P(x, y)$，其中x方向的平移量为Δx，y方向的平移量为Δy。那么，点$P(x, y)$的坐标为

$$\begin{cases} x = x_0 + \Delta x \\ y = y_0 + \Delta y \end{cases}$$

如图6-1所示。这个变换用矩阵的形式可以表示为

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

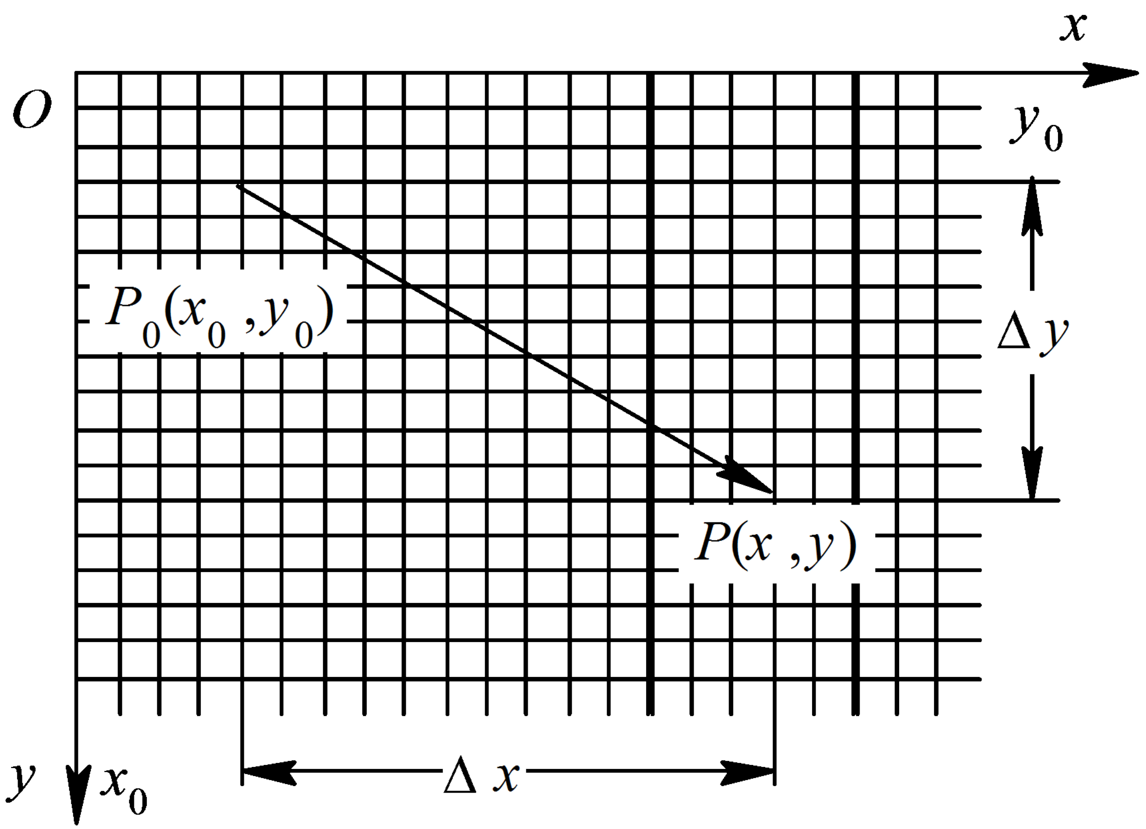

图6-1 点的平移

要实现平移变换，平面上点的变换矩阵

$$T=\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

需要使用2×3阶变换矩阵，取其形式为

$$T=\begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix}$$

此矩阵的第一、二列构成单位矩阵，第三列元素为平移常量。

同时要满足矩阵相乘时要求前者的列数与后者的行数相等的规则。

所以需要在点的坐标列矩阵［*x* *y*］^{T}中引入第三个元素，增加一个附加坐标，扩展为3×1的列矩阵［*x* *y* 1］^{T}，这样用三维空间点（*x*, *y*, 1）表示二维空间点（*x*, *y*），即采用一种特殊的坐标，可以实现平移变换，变换结果为

$$P = T \cdot P_0 = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 + \Delta x \\ y_0 + \Delta y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

式 $\begin{cases} x = x_0 + \Delta x \\ y = y_0 + \Delta y \end{cases}$ 符合上述平移后的坐标位置。由此可得平移变换矩阵为

$$T = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \end{bmatrix}$$

通常将2×3阶矩阵扩充为3×3阶矩阵，以拓宽功能。下面再验证一下点$P\ (x, y)$按照3×3的变换矩阵T平移变换的结果。

$$P = T \cdot P_0 = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_0 + \Delta x \\ y_0 + \Delta y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

从上式可以看出，引入附加坐标后，扩充了矩阵的第3行，并没有使变换结果受到影响。这种用n+1维向量表示n维向量的方法称为齐次坐标表示法。

因此，2D图像中的点坐标(x, y)通常表示成齐次坐标（Hx, Hy, H），其中H表示非零的任意实数，当$H=1$时，则$(x, y, 1)$就称为点(x, y)的规范化齐次坐标。

由点的齐次坐标（Hx, Hy, H）求点的规范化齐次坐标$(x, y, 1)$，可按如下公式进行：

$$x = \frac{Hx}{H} \qquad y = \frac{Hy}{H}$$

齐次坐标的几何意义相当于点(x, y)落在3D空间H＝1的平面上， 如图6-2所示。如果将XOY 平面内的三角形abc 的各顶点表示成齐次坐标$(x_i, y_i, 1)(i=1, 2, 3)$的形式，就变成$H＝1$平面内的三角形$a_1b_1c_1$的各顶点。

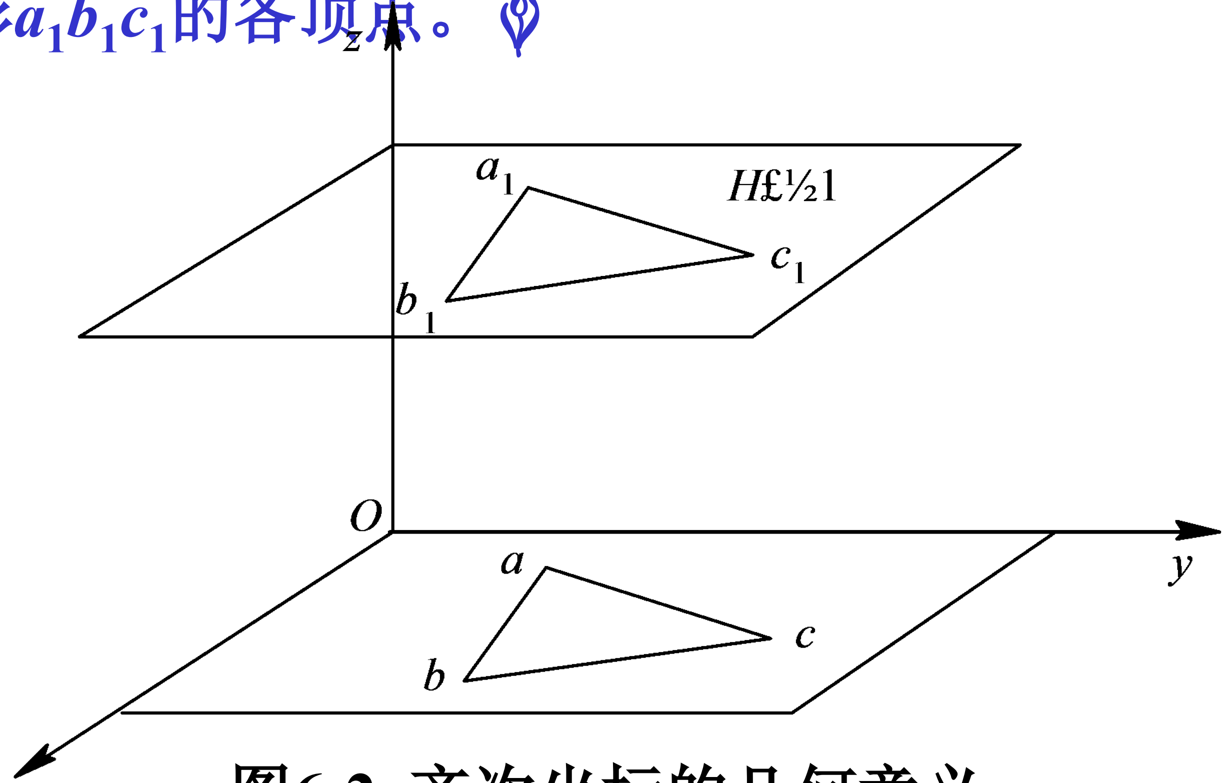

图6-2 齐次坐标的几何意义

齐次坐标在2D图像几何变换中的另一个应用是：如某点S(60 000， 40 000)在16位计算机上表示则大于32 767的最大坐标值，需要进行复杂的操作。但如果把*S*的坐标形式变成（*Hx*, *Hy*, *H*）形式的齐次坐标，则情况就不同了。在齐次坐标系中，设$H＝1／2$，则 (60 000，40 000)的齐次坐标为(1/2*x*, 1/2*y*, 1/2)，那么所要表示的点变为(30 000, 20 000, 1/2)，此点显然在16位计算机上二进制数所能表示的范围之内。

因此，采用齐次坐标，并将变换矩阵改成3×3阶的形式后，便可实现所有2D图像几何变换的基本变换。

6.1.3 二维图像几何变换的矩阵

利用齐次坐标及改成3×3阶形式的变换矩阵，实现2D图像几何变换的基本变换的一般过程是：

1、 将2×n阶的二维点集矩阵 $\begin{bmatrix} x_{0i} \\ y_{0i} \end{bmatrix}_{2\times n} \xrightarrow{\text{表示成齐次坐标}} \begin{bmatrix} x_{0i} \\ y_{0i} \\ 1 \end{bmatrix}_{3\times n}$

2、然后乘以相应的变换矩阵即可完成。即

变换后的点集矩阵 = 变换矩阵T×变换前的点集矩阵

（图像上各点的新齐次坐标） （图像上各点的原齐次坐标）

设变换矩阵*T*为

$$T = \begin{bmatrix} a & b & p \\ c & d & q \\ l & m & s \end{bmatrix}$$

则上述变换可以用公式表示为

$$\begin{bmatrix} Hx_1' & Hx_2' & \mathrm{L} & Hx_n' \\ Hy_1' & Hy_2' & \mathrm{L} & Hy_n' \\ H & H & \mathrm{L} & H \end{bmatrix}_{3\times n} = T \times \begin{bmatrix} x_1 & x_2 & \mathrm{L} & x_n \\ y_1 & y_2 & \mathrm{L} & y_n \\ 1 & 1 & \mathrm{L} & 1 \end{bmatrix}_{3\times n}$$

图像上各点的新齐次坐标规范化后的点集矩阵为

$$\begin{bmatrix} x_1' & x_2' & \cdots & x_n' \\ y_1' & y_2' & \cdots & y_n' \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{3\times n}$$

引入齐次坐标后，表示2D图像几何变换的3×3矩阵的功能就完善了，可以用它完成2D图像的各种几何变换。下面讨论3×3阶变换矩阵中各元素在变换中的功能。几何变换的3×3矩阵的一般形式为

$$T=\begin{bmatrix} a & b & p \\ c & d & q \\ l & m & s \end{bmatrix}$$

3×3的阶矩阵T可以分成四个子矩阵。其中，$\begin{bmatrix} a & b \\ c & d \end{bmatrix}_{2\times 2}$ **这一子矩阵可使图像实现恒等、比例、反射（或镜像）、错切和旋转变换。［*l m*］这一行矩阵可以使图像实现平移变换。［*p q*］^{T}这一列矩阵可以使图像实现透视变换，但当*p*=0，*q*=0时它无透视作用。［*s*］这一元素可以使图像实现全比例变换。例如，将图像进行全比例变换，即**

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix} \times \begin{bmatrix} x_{0i} \\ y_{0i} \\ 1 \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ s \end{bmatrix}$$

将齐次坐标 $\begin{bmatrix} x_i \\ y_i \\ s \end{bmatrix}$ 规范化后，$\begin{bmatrix} \frac{x_{0i}}{s} \\ \frac{y_{0i}}{s} \\ 1 \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$。由此可见，当s>1时，图像按比例缩小；当0<$s$<1时，整个图像按比例放大；当$s$=1时，图像大小不变。

6.2 图像比例缩放

6.2.1 图像比例缩放变换

图像比例缩放是指将给定的图像在x轴方向按比例缩放f_x倍，在y轴方向按比例缩放f_y倍，从而获得一幅新的图像。如果$f_x=f_y$，即在x轴方向和y轴方向缩放的比率相同，称这样的比例缩放为图像的全比例缩放。如果$f_x \neq f_y$，图像的比例缩放会改变原始图像的像素间的相对位置，产生几何畸变。设原图像中的点$P_0(x_0,y_0)$比例缩放后，在新图像中的对应点为$P(x, y)$，则$P_0(x_0,y_0)$和$P(x, y)$之间的对应关系如图6-3所示。

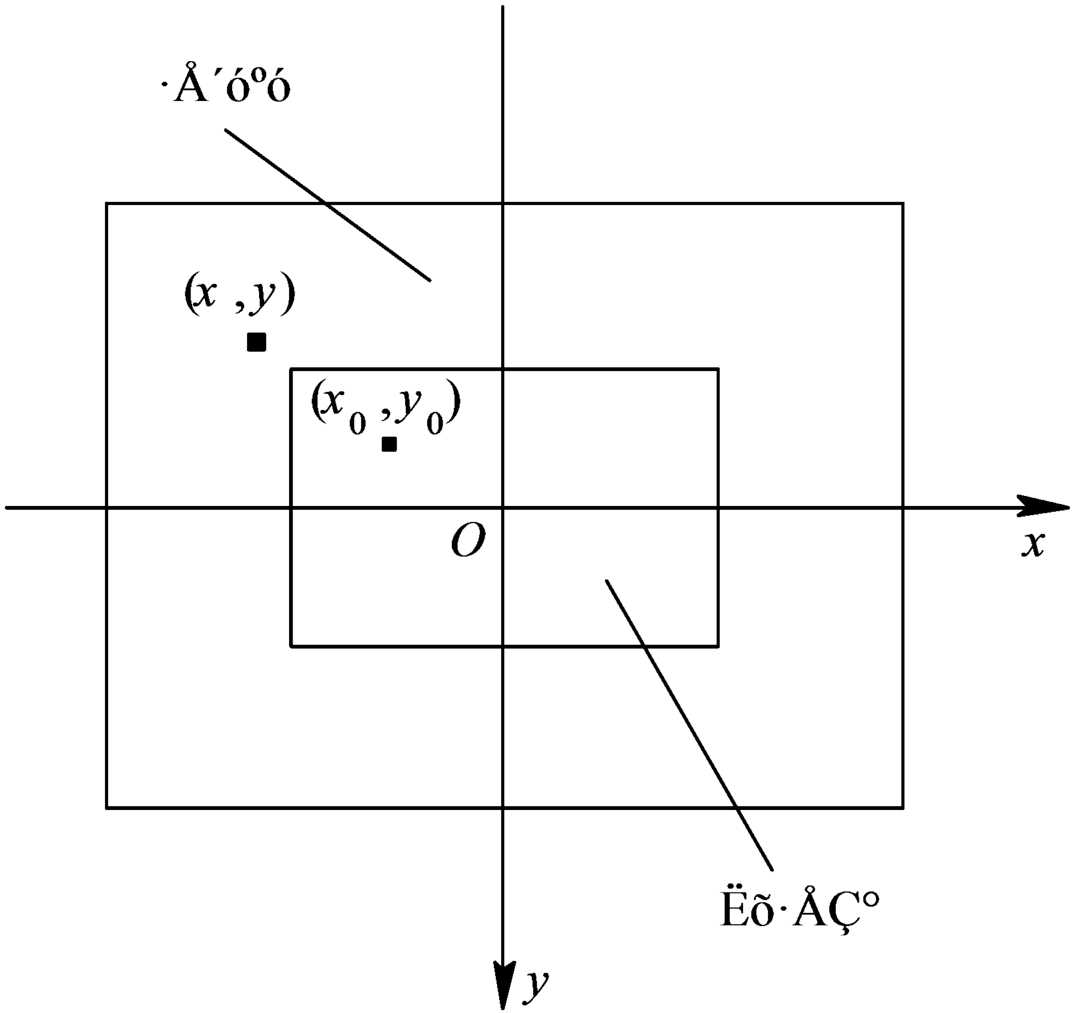

图6-3 比例缩放

比例缩放前后两点$P_0(x_0, y_0)$、$P(x, y)$之间的关系用矩阵形式可以表示为

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} fx & 0 & 0 \\ 0 & fy & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \qquad (6\text{-}1)$$

公式（6-1）的逆运算为

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{fx} & 0 & 0 \\ 0 & \frac{1}{fy} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

即

$$\begin{cases} x_0 = \dfrac{x}{fx} \\ y_0 = \dfrac{y}{fy} \end{cases}$$

比例缩放所产生的图像中的像素可能在原图像中找不到相应的像素点，这样就必须进行插值处理。插值处理常用的方法有两种，一种是直接赋值为和它最相近的像素值，另一种是通过一些插值算法来计算相应的像素值。前一种方法计算简单，但会出现马赛克现象；后者处理效果要好些，但是运算量也相应增加。在下面的算法中直接采用了前一种做法。实际上，这也是一种插值算法，称为最邻近插值法（Nearest Neighbor Interpolation）。

首先讨论图像的比例缩小。最简单的比例缩小是当 $fx=fy=1/2$时，图像被缩到一半大小，图像缩小之后，因为承载的信息量小了，所以画布可相应缩小。此时， 只需在原图像基础上，每行隔一个像素取一点，每隔一行进行操作，即取原图的偶（奇）数行和偶（奇）数列构成新的图像，如图6-4所示。

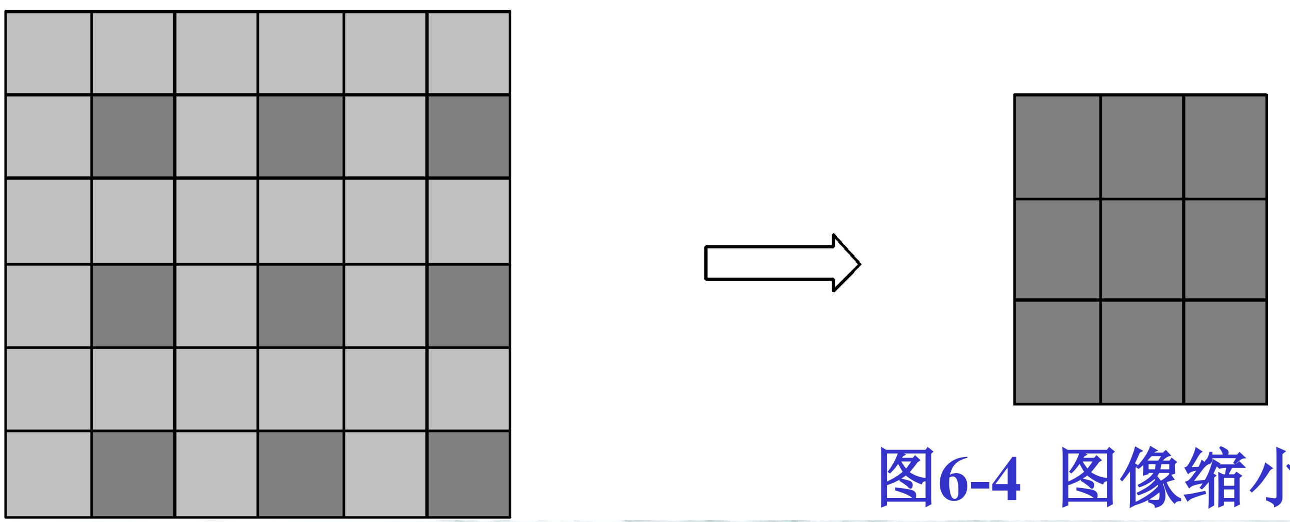

图6-4 图像缩小一半

如果图像按任意比例缩小，则需要计算选择的行和列。

如果$M \times N$大小的原图像$F(x, y)$缩小为$kM \times kN$大小（$k<1$）的新图像$I(x, y)$时，则

$$I(x, y)=F(\mathrm{int}(c \times x), \mathrm{int}(c \times y))$$

其中，$c=1 / k$。由此公式可以构造出新图像，如图6-5所示。

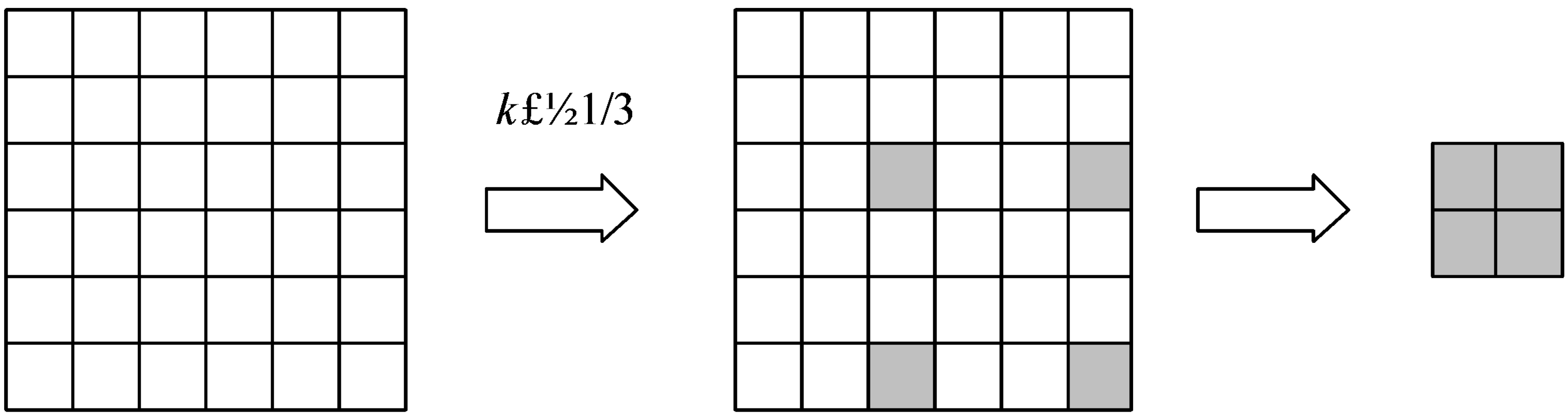

图6-5 图像按任意比例缩小

当$f_x \neq f_y(f_x, f_y > 0)$时，图像不按比例缩小，这种操作因为在x方向和y方向的缩小比例不同，一定会带来图像的几何畸变。图像不按比例缩小的方法是：如果$M \times N$大小的旧图$F(x，y)$缩小为$k_1M \times k_2N$（$k_1<1$，$k_2<1$）大小的新图像$I(x，y)$时，则

$$I(x, y)=F(\mathrm{int}(c_1 \times x), \mathrm{int}(c_2 \times y))$$

其中

$$c1 = \frac{1}{k_1}, c_2 = \frac{1}{k_2}$$

由此公式可以构造出新图像。

图像在缩小操作中，是在现有的信息里如何挑选所需要的有用信息。

在图像的放大操作中，则需要对尺寸放大后所多出来的空格填入适当的像素值，这是信息的估计问题，所以较图像的缩小要难一些。当$fx=fy=2$时，图像被按全比例放大2倍， 放大后图像中的(0，0)像素对应于原图中的(0，0)像素；(0，1)像素对应于原图中的(0，0.5)像素，该像素不存在，可以近似为(0，0)也可以近似 (0，1)； (0，2)像素对应于原图像中的(0，1)像素；(1，0)像素对应于原图中的(0.5，0)，它的像素值近似于(0， 0)或(1，0)像素；(2，0)像素对应于原图中的(1，0)像素，依此类推。其实这是将原图像每行中的像素重复取值一遍，然后每行重复一次。图6-6是原始图像，图6-7和图6-8是分别采用上述两种近似方法放大后的图像。

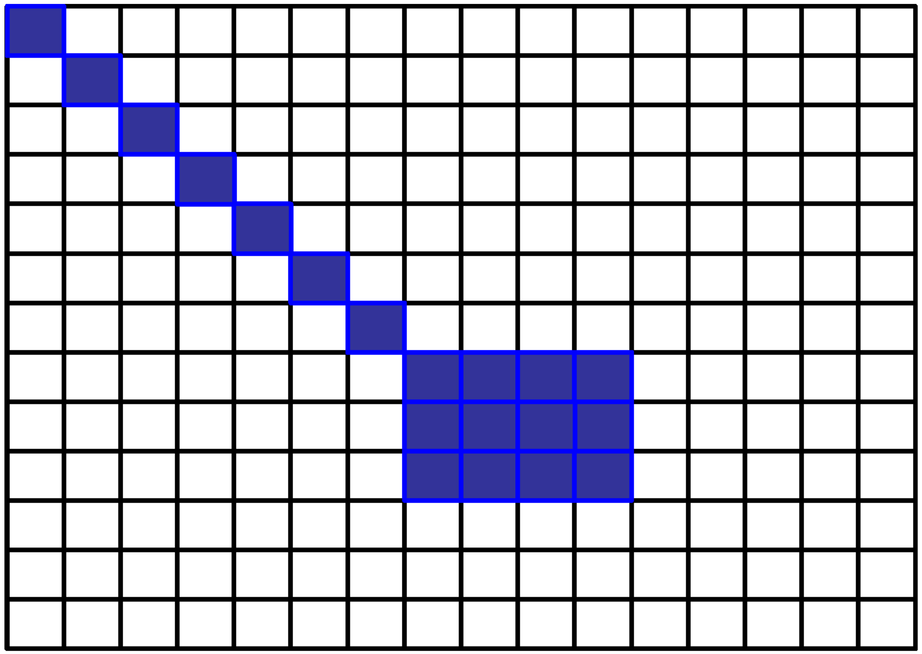

图6-6 放大前的图像

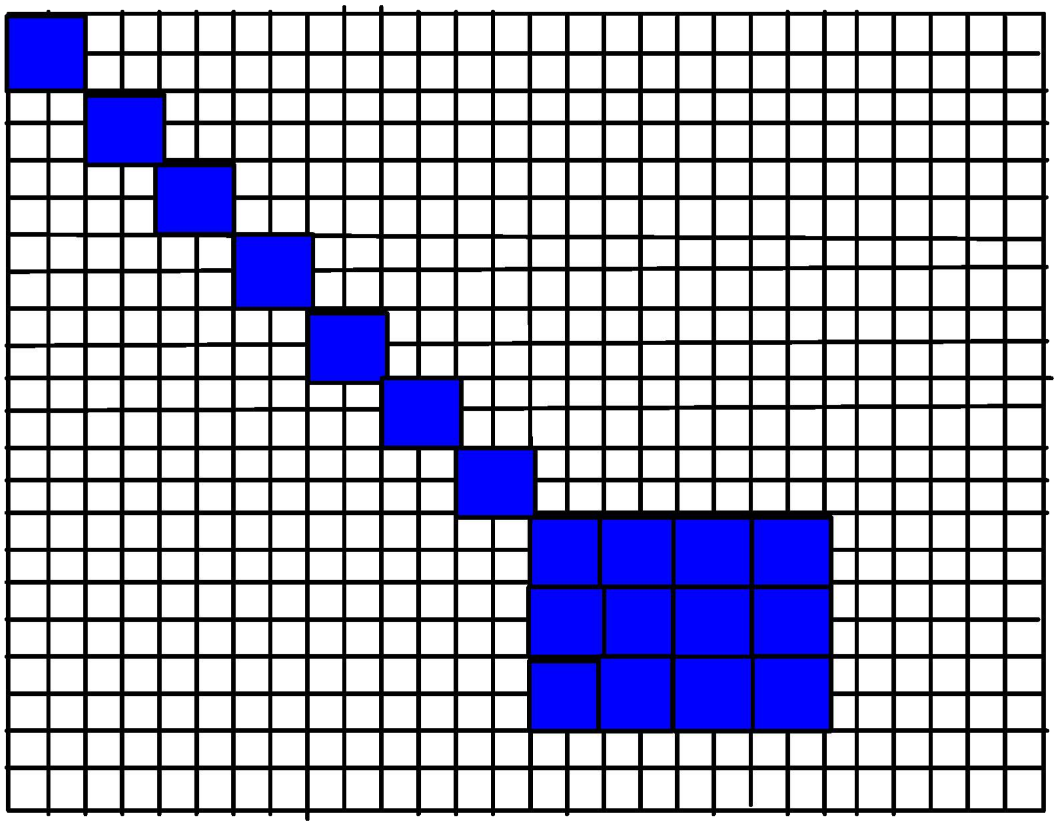

图6-7 按最近邻域法放大两倍的图像

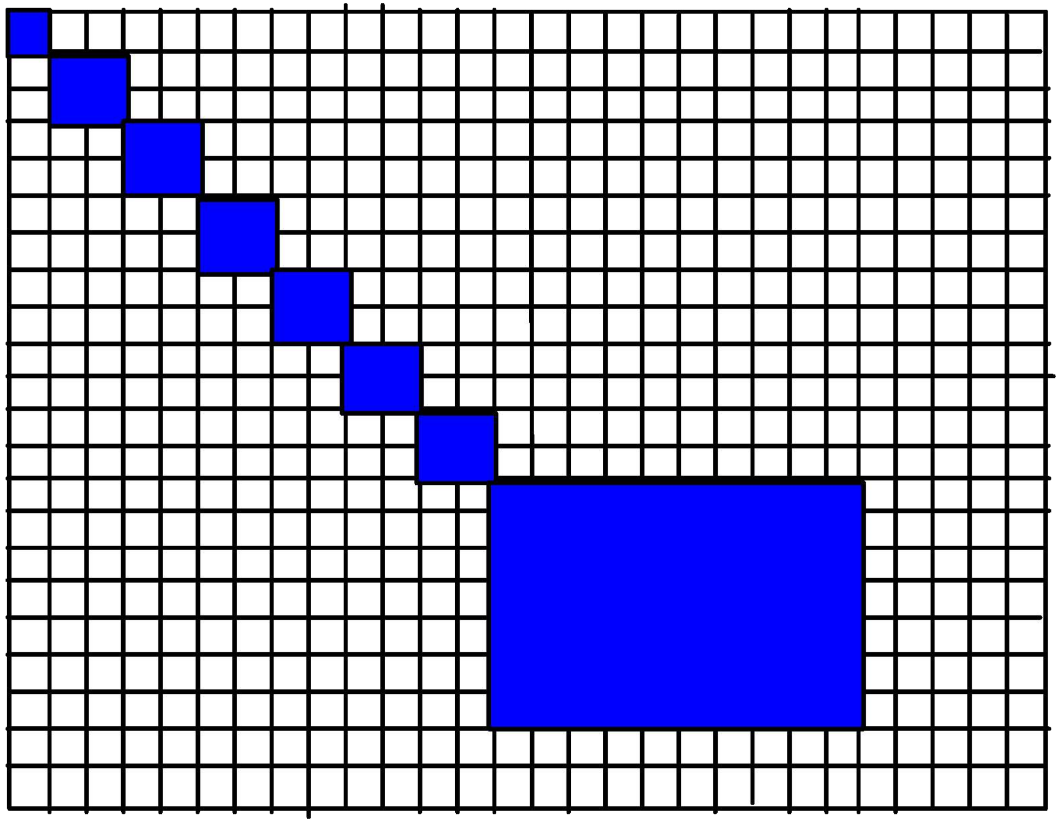

图6-8 按插值法放大两倍的图像

一般地，按比例将原图像放大k倍时，如果按照最近邻域法则需要将一个像素值添在新图像的$k\times k$的子块中，如图6-9所示。显然，如果放大倍数太大， 按照这种方法处理会出现马赛克效应。当$fx\neq fy(fx, fy>0)$时，图像在x方向和y方向不按比例放大，此时， 这种操作由于x方向和y方向的放大倍数不同，一定带来图像的几何畸变。放大的方法是将原图像的一个像素添到新图像的一个$k_1\times k_2$的子块中去。为了提高几何变换后的图像质量，常采用线性插值法。该方法的原理是，当求出的分数地址与像素点不一致时，求出周围四个像素点的距离比，根据该比率，由四个邻域的像素灰度值进行线性插值， 如图6-10所示。

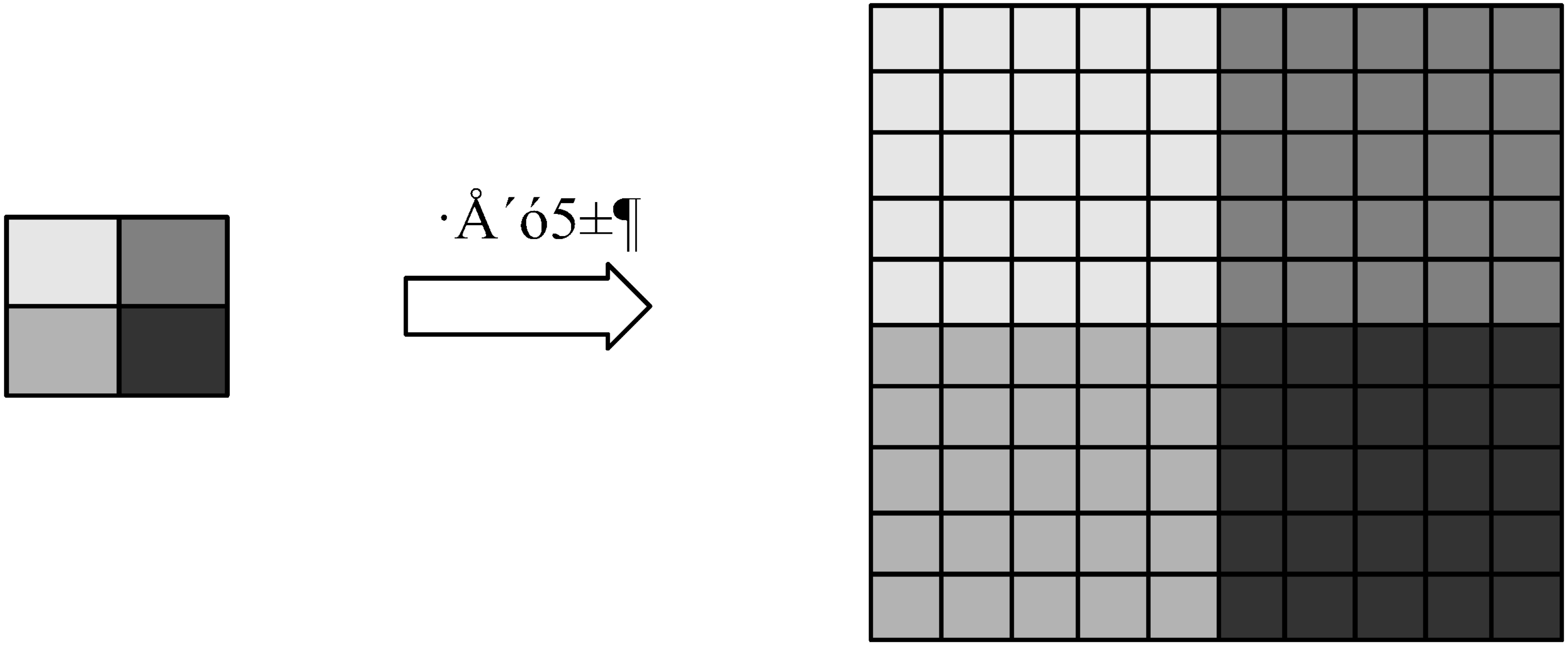

图6-9 按最近邻域法放大五倍的图像

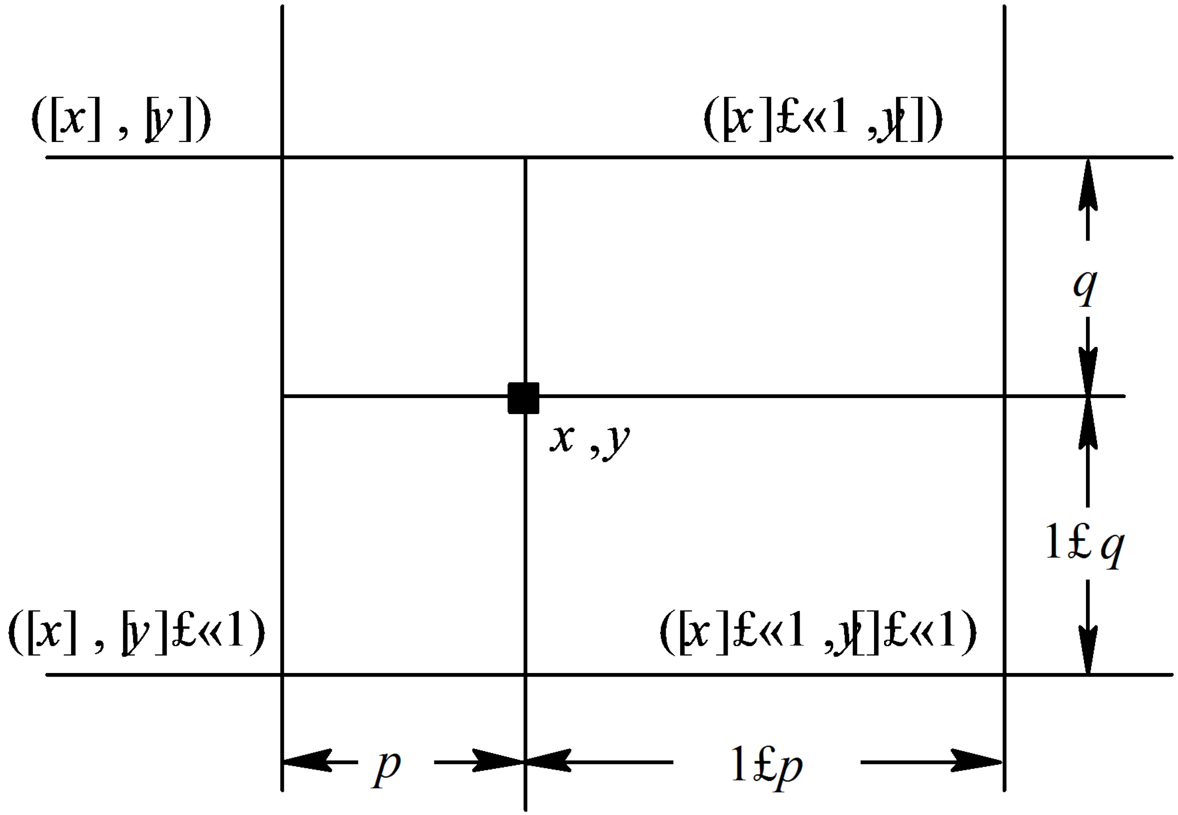

图6-10 线性插值法示意图

简化后的灰度值计算式如下:

$$g(x,y)=(1-q)\{(1-p)\times g([x],[y])+p\times g([x]+1,[y])\}+q\{(1-p)\times g([x],[y]+1)+p\times g([x]+1,[y]+1)\}$$

式中：$g(x，y)$为坐标$(x，y)$处的灰度值，［x］、［y］分别为不大于x，y的整数。关于这个问题的详细算法及其实现，读者可以参考有关参考文献。

6.2.2 比例缩放的实现

图6-11 图像比例缩放处理结果

6.3 图 像 平 移

6.3.1 图像平移变换

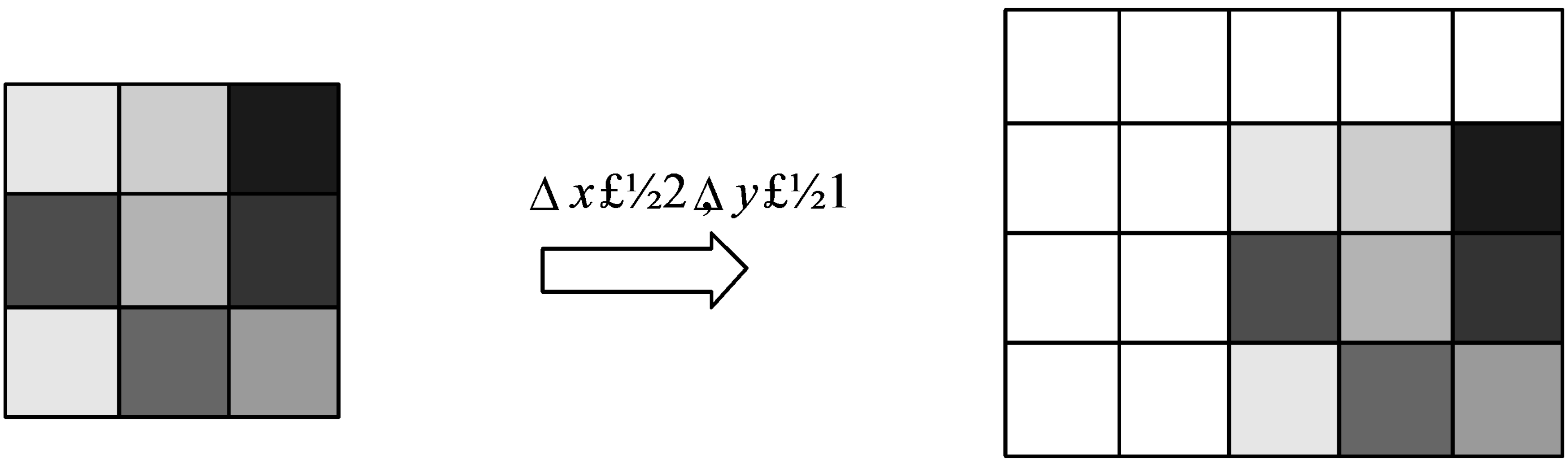

图6-12 图像平移

设点$P_0(x_0, y_0)$进行平移后，移到$P(x, y)$，其中x方向的平移量为Δx，y方向的平移量为Δy。那么，点$P(x, y)$的坐标为

$$\begin{cases} x = x_0 + \Delta x \\ y = y_0 + \Delta y \end{cases}$$

利用齐次坐标，变换前后图像上的点$P_0(x_0, y_0)$和$P(x, y)$之间的关系可以用如下的矩阵变换表示为

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{6-2}$$

对变换矩阵求逆，可以得到式（6-2）的逆变换

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta x \\ 0 & 1 & -\Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

即

$$\begin{cases} x_0 = x - \Delta x \\ y_0 = y - \Delta y \end{cases}$$

这样，平移后的图像上的每一点都可以在原图像中找到对应的点。例如，对于新图中的（0，0）像素，代入上面的方程组，可以求出对应原图中的像素($-\Delta x$，$-\Delta y$)。如果Δx或Δy大于0，则点($-\Delta x$，$-\Delta y$)不在原图像中。对于不在原图像中的点，可以直接将它的像素值统一设置为0或者255。同样，若有像素点不在原图像中，也就说明原图像中有点被移出显示区域。如果不想丢失被移出的部分图像，可以将新生成的图像宽度扩大$|\Delta x|$，高度扩大$|\Delta y|$。

图6-13 平移前的图像

图6-14 平移后的图像

图6-15 平移扩大后的图像

6.4 图 像 镜 像

6.4.1 图像镜像变换

图像的镜像(Mirror)变换分为两种：一种是水平镜像，另外一种是垂直镜像。图像的水平镜像操作是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；图像的垂直镜像操作是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，如图6-16所示。

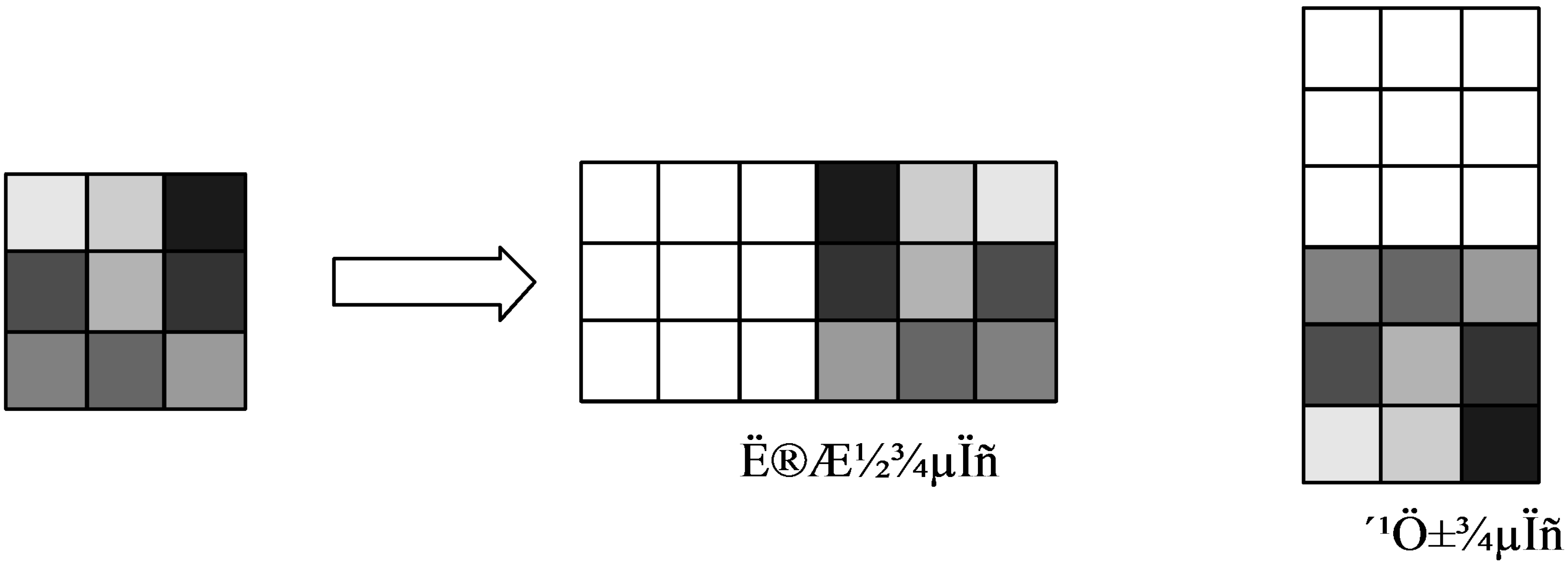

图6-16 图像的镜像

图像的镜像变换也可以用矩阵变换表示。设点$P_0(x_0, y_0)$进行镜像后的对应点为$P(x, y)$，图像高度为*fHeight*，宽度为*fWidth*，原图像中$P_0(x_0, y_0)$经过水平镜像后坐标将变为（$fWidth-x_0$，y_0），其矩阵表达式为

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & fWidth \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{6-3}$$

逆运算矩阵表达式为

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & fWidth \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

即

$$\begin{cases} x_0 = fWidth - x \\ y_0 = y \end{cases}$$

同样， $P_0(x_0, y_0)$经过垂直镜像后坐标将变为(x_0，$fHeight-y_0$)， 其矩阵表表达式为

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & fHeight \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} \tag{6-4}$$