数字图像处理

Digital Image Processing

信息工程学院

School of Information Engineering

3.4 几何运算

胡辑伟 主讲

3.4 几何运算 (Geometric Operation)

1.概念

- 几何运算就是改变图像中物体对象（像素）之间的空间关系。
- 从变换性质来分，几何变换可以分为图像的位置变换（平移、镜像、旋转）、形状变换（放大、缩小）以及图像的复等合变换。

图像几何运算的一般定义为：

$$g(x,y)=f(u,v)=f(p(x,y),q(x,y))$$

式中，$u=p(x,y)$ ，$v=q(x,y)$ 唯一的描述了空间变换，即将输入 图像 $f(u,v)$ 从 $u-v$ 坐标系变换为 $x-y$ 坐标系的输出图像 $g(x,y)$

3.4.1图像的平移(Image Translation)

图像的平移：

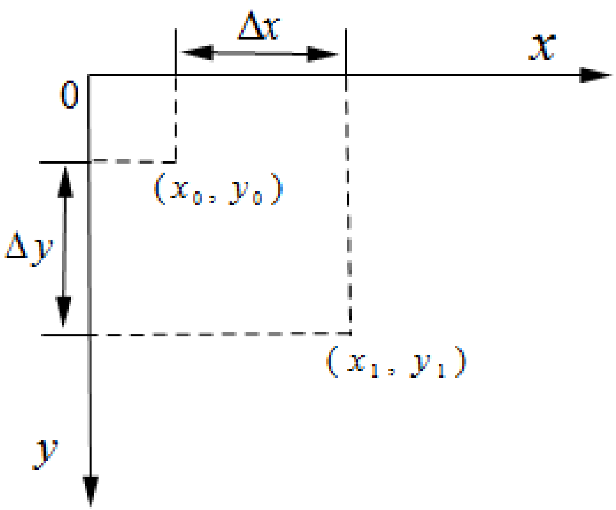


图3.8 像素点的平移

两点之间存在如下关系：

$$\begin{cases} x_1 = x_0 + \Delta x \\ y_1 = y_0 + \Delta y \end{cases}$$

3.4.1图像的平移(Image Translation)

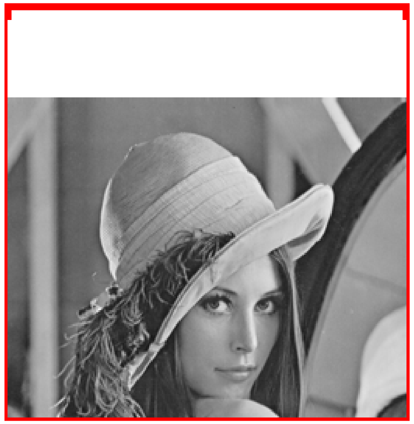
$$\begin{cases} x_1 = x_0 + \Delta x \\ y_1 = y_0 + \Delta y \end{cases}$$

以矩阵形式表示平移前后的像素关系为：

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta x \\ 0 & 1 & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

3.4.1图像的平移(Image Translation)

（ a ）原始图像



（ b ）平移后的图像

图3.9　图像的平移

3.4.2图像的镜像(Image Mirror)

图像的镜像（Mirror）：

定义：指原始图像相对于某一参照面旋转180°的图像

设原始图像的宽为 w **，高为** h**，原始图像中的点为** (x_0, y_0) **，对称变换后的点为** (x_1, y_1) **。**

- **（1）水平镜像（相对于** y**轴）**

水平镜像的变换公式 如下：

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

3.4.2图像的镜像(Image Mirror)

(a)原始图像

(b)水平镜像

图3.10　图像水平镜像变换

3.4.2图像的镜像(Image Mirror)

- （2）垂直镜像（相对于x轴）

垂直镜像的变换公式为如下：

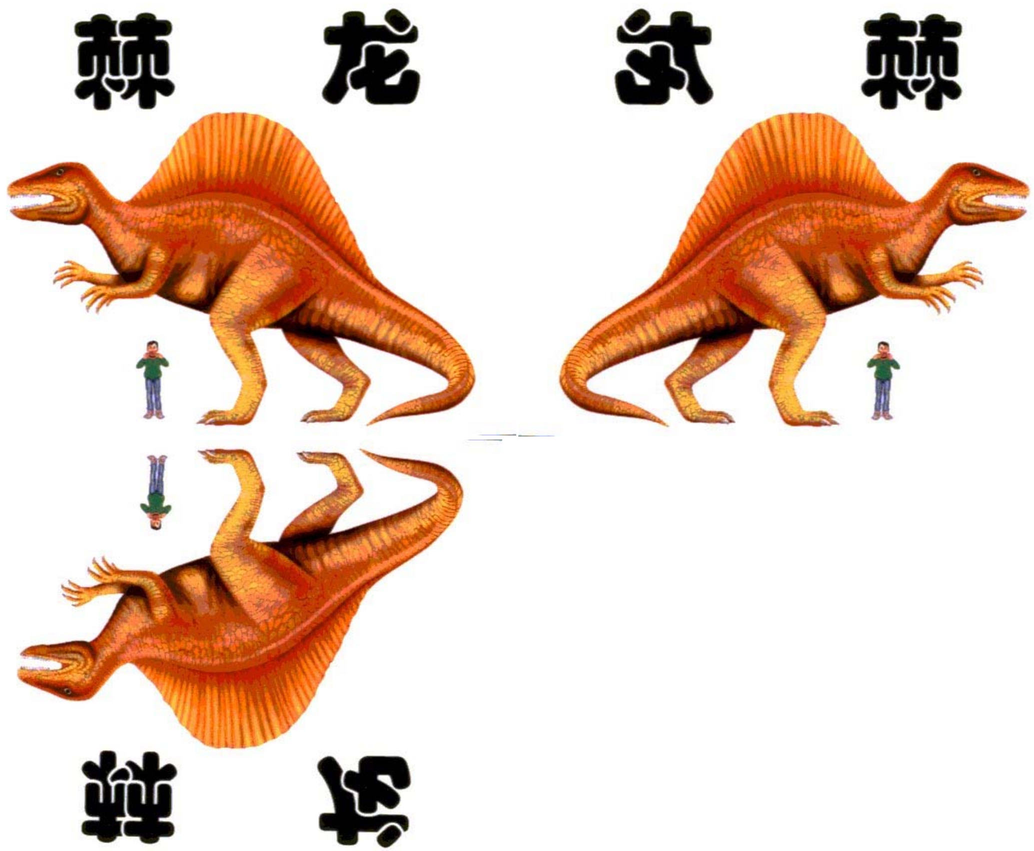
$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & h \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

3.4.2图像的镜像(Image Mirror)

(a)原始图像

(b)垂直镜像

图3.11　图像垂直镜像变换



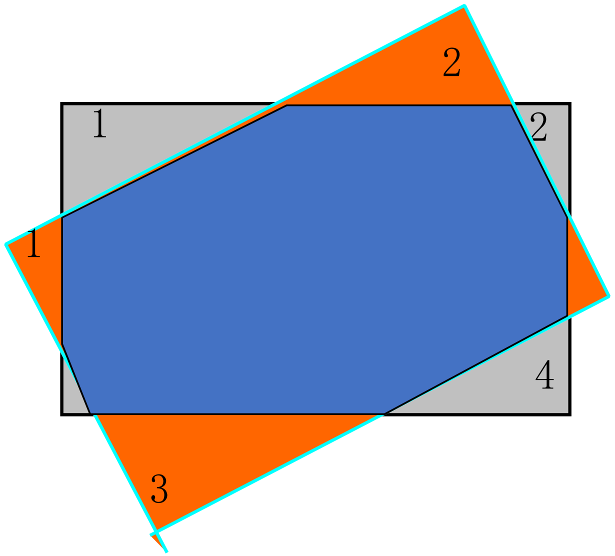
水平镜像

垂直镜像

3.4.3图像的旋转(Image Rotation)

图像的旋转：

一般图像的旋转是以图像的中心为原点，旋转一定的角度，即将图像上的所有像素都旋转一个相同的角度。



3.4.3图像的旋转(Image Rotation)

设原始图像的任意点$A_0(x_0, y_0)$经旋转角度β 以后到新的位置$A(x，y)$，为表示方便，采用极坐标形式表示，原始的角度为 α ，如下图所示：

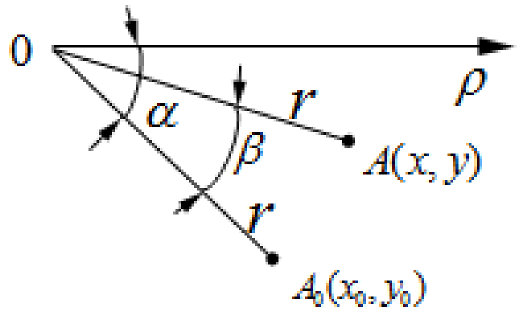


图3.12　图像的旋转

原始图像的点 $A_0(x_0, y_0)$ 的坐标如下：

$$\begin{cases} x_0 = r\cos\alpha \\ y_0 = r\sin\alpha \end{cases}$$

3.4.3图像的旋转(Image Rotation)

旋转到新位置以后点 $A(x, y)$ **的坐标如下：**

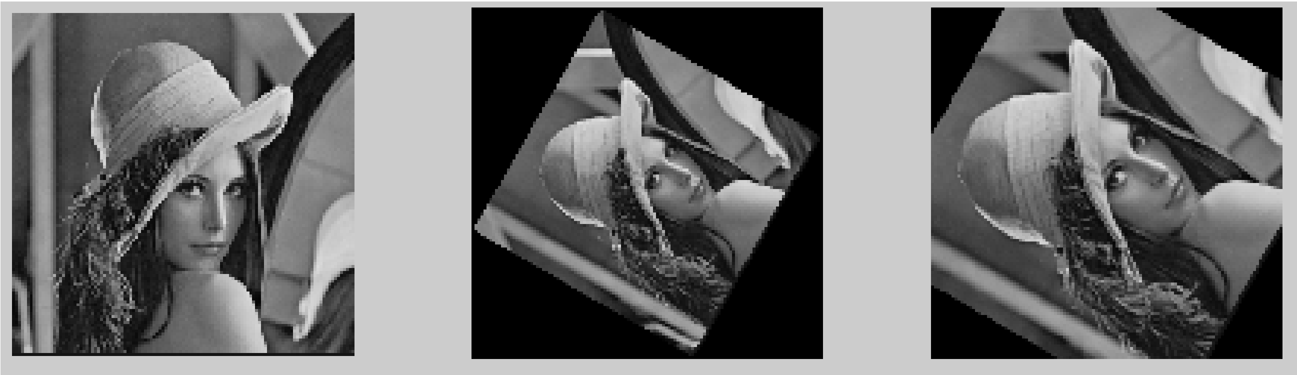
$$\begin{cases} x = r\cos(\alpha-\beta) = r\cos\alpha\cos\beta + r\sin\alpha\sin\beta \\ y = r\sin(\alpha-\beta) = r\sin\alpha\cos\beta - r\cos\alpha\sin\beta \end{cases}$$

$$\begin{cases} x = x_0\cos\beta + y_0\sin\beta \\ y = -x_0\sin\beta + y_0\cos\beta \end{cases}$$

- **图像旋转用矩阵表示如下：**

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

3.4.3图像的旋转(Image Rotation)



（a）原图　　（b）旋转图　　（c）旋转图

图3.13　图像的旋转

3.4.3图像的旋转(Image Rotation)

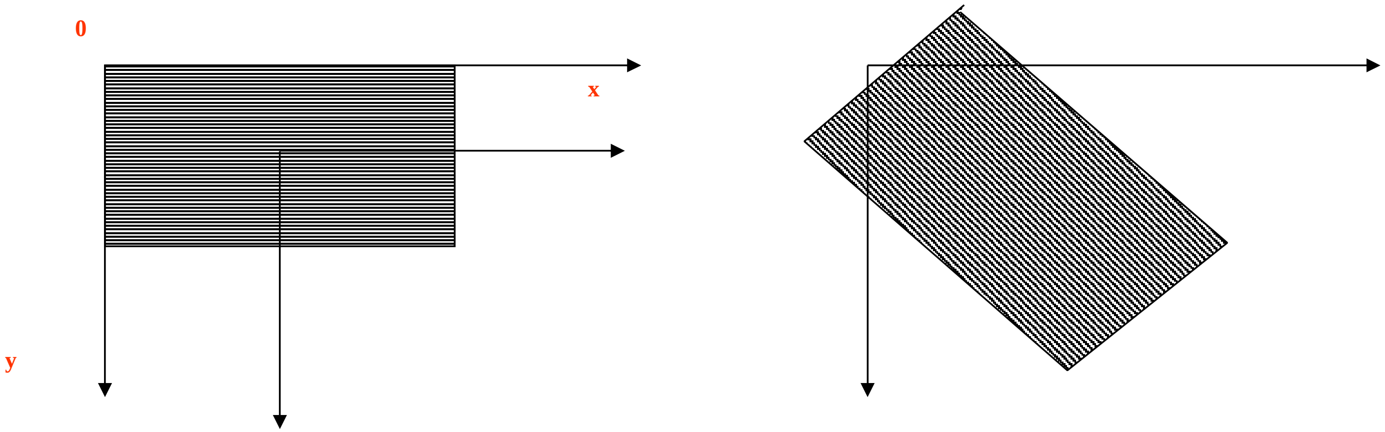
图像旋转之后，由于数字图像的坐标值必须是整数，因此，可能引起图像部分像素点的局部改变，因此，这时图像的大小也会发生一定的改变。

若图像旋转角 β =45时，则变换关系如下：

$$\begin{cases} x = 0.707x_0 + 0.707y_0 \\ y = -0.707x_0 + 0.707y_0 \end{cases}$$

图像绕任意点旋转

上述的旋转是绕坐标轴原点（0，0）进行的，如果是绕某一个指定点（a，b）旋转，则先要将坐标系平移到该点，再进行旋转，然后将旋转后的图像平移回原坐标系。例如，我们这里以图像的中心为旋转中心：



利用公式进行图像旋转正变换时需要注意如下两点：

1、为了避免图像信息的丢失，图像旋转后必须进行平移变换。

2、图像旋转之后，会出现许多空洞点，我们需要对这些空洞点必须进行填充处理，否则图像旋转后的效果不好，一般也称这种操作为插值处理，可采用行或列插值方法。最简单的插值方法是，图像旋转前某一点（x，y）的像素点颜色，除了填充在旋转后坐标（x′，y′）上外，还要填充（x′+1，y′）和（x′，y′+1）。

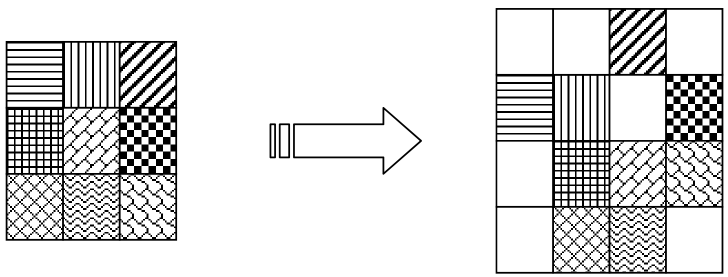


图7-9：图像的旋转

3.4.3图像的旋转(Image Rotation)

图像旋转角 β = 45°时，则变换关系如下：

$$\begin{cases} x = 0.707x_0 + 0.707y_0 \\ y = -0.707x_0 + 0.707y_0 \end{cases}$$

以原始图像的点（1，1）为例，旋转以后，均为小数，经舍入后为（1，0），产生了位置误差。因此，图像旋转之后，可能会出现一些空白点，需要对这些空白点进行灰度级的插值处理，否则影响旋转后的图像质量。

旋转前的图像

图旋转15°并进行插值处理的图像


谢 谢
THANK YOU