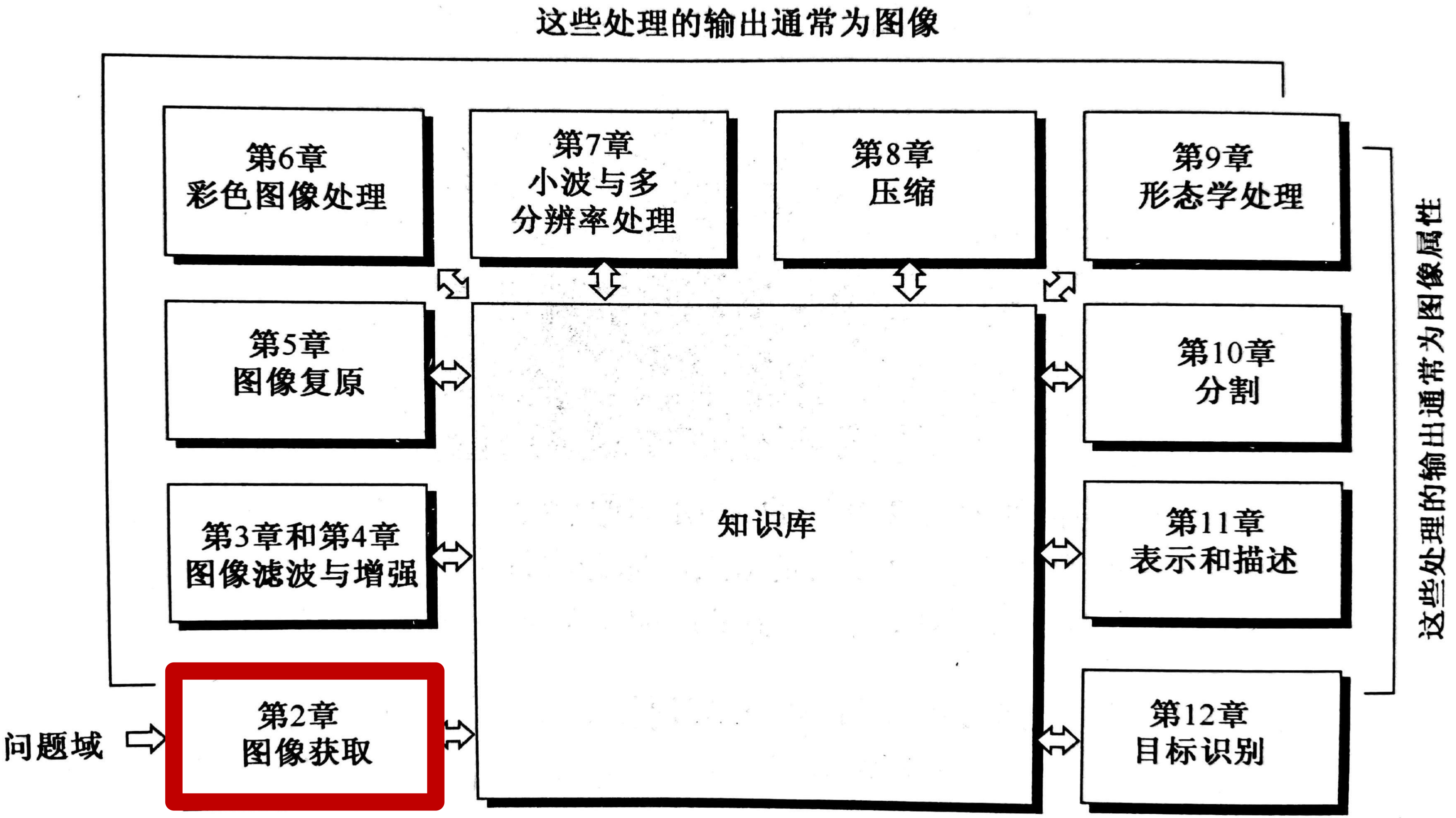
数字图像处理：图像采集 II

Fei Gao
gaofeihifly@163.com
https://fei-hdu.github.io/

数字图像处理的基本步骤及内容



目录

2.5 像素间的一些基本关系

- **相邻像素**
  - 4-邻域，记为$N_4(p)$
  - 对角邻域，记为$N_D(p)$
  - 8-邻域，记为$N_8(p)$

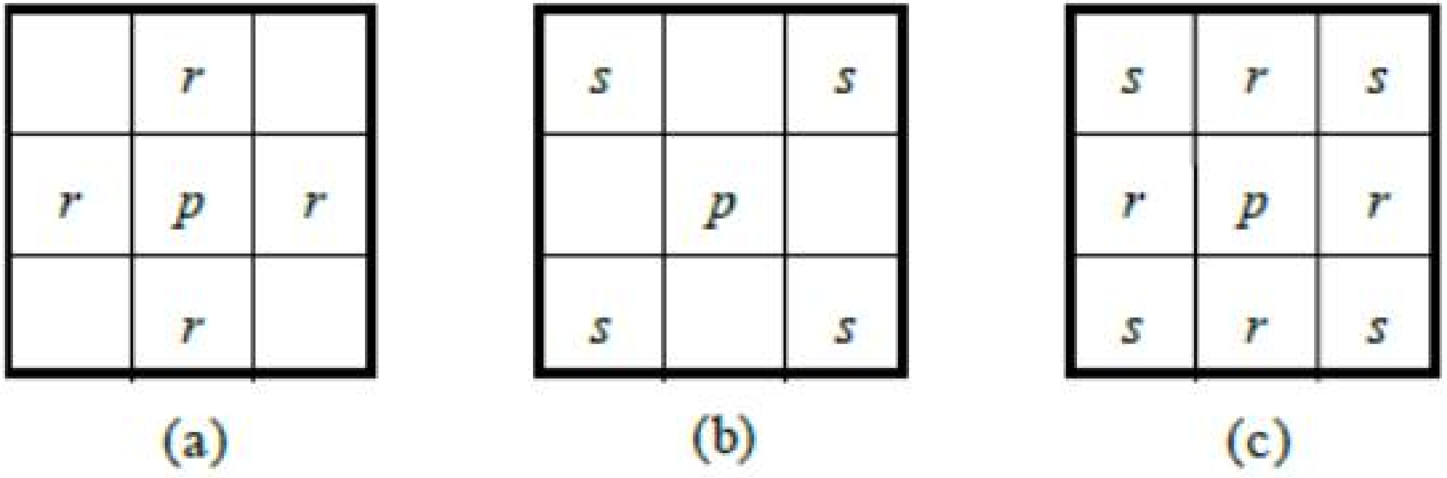


图 2.4.1 像素的邻域

2.5 像素间的一些基本关系

- 邻接性、连通性、区域和边界
  - 邻接像素：在二值图像中，把具有1值的像素归诸于邻接像素，V={1}。在灰度图像中，V可以是0-255之间灰度的子集。
  - 4邻接：如果q在集合$N_4(p)$中，则具有V中数值的两个像素p和q是4邻接的；
  - 8邻接：如果q在集合$N_8(p)$中，则具有V中数值的两个像素p和q是8邻接的；
  - m邻接（混合邻接）：如果（i）q在$N_4(p)$中，或（i）q在$N_D(p)$中，且集合 $N_4(p)\cap N_4(q)$ 中更没有来自V中数值的像素，则具有V中数值的两个像素p和q是m邻接的。

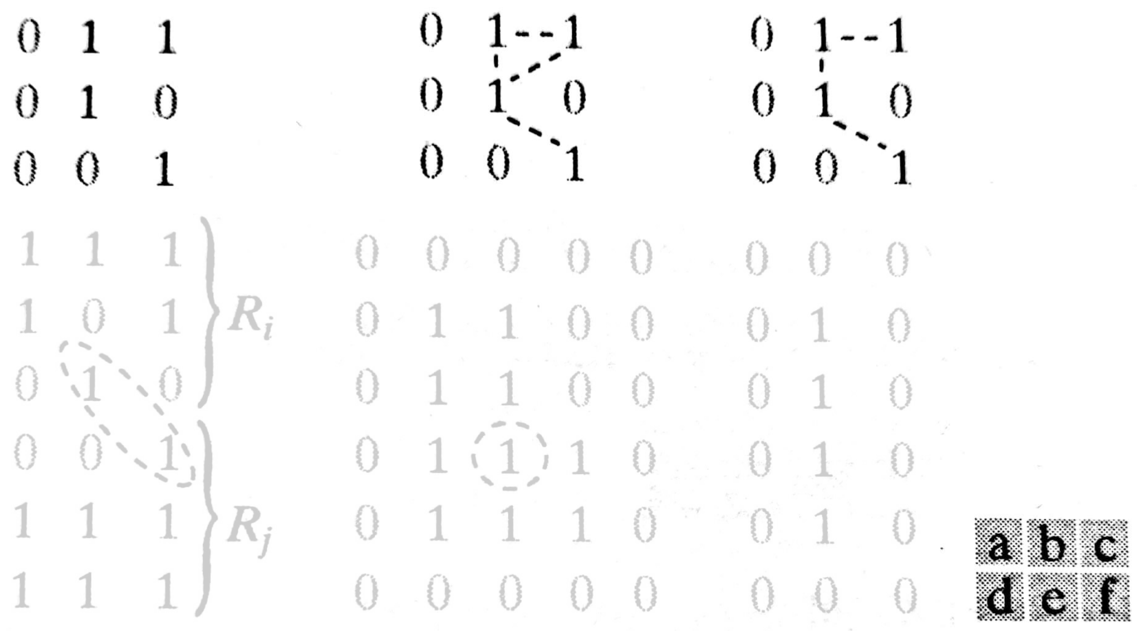
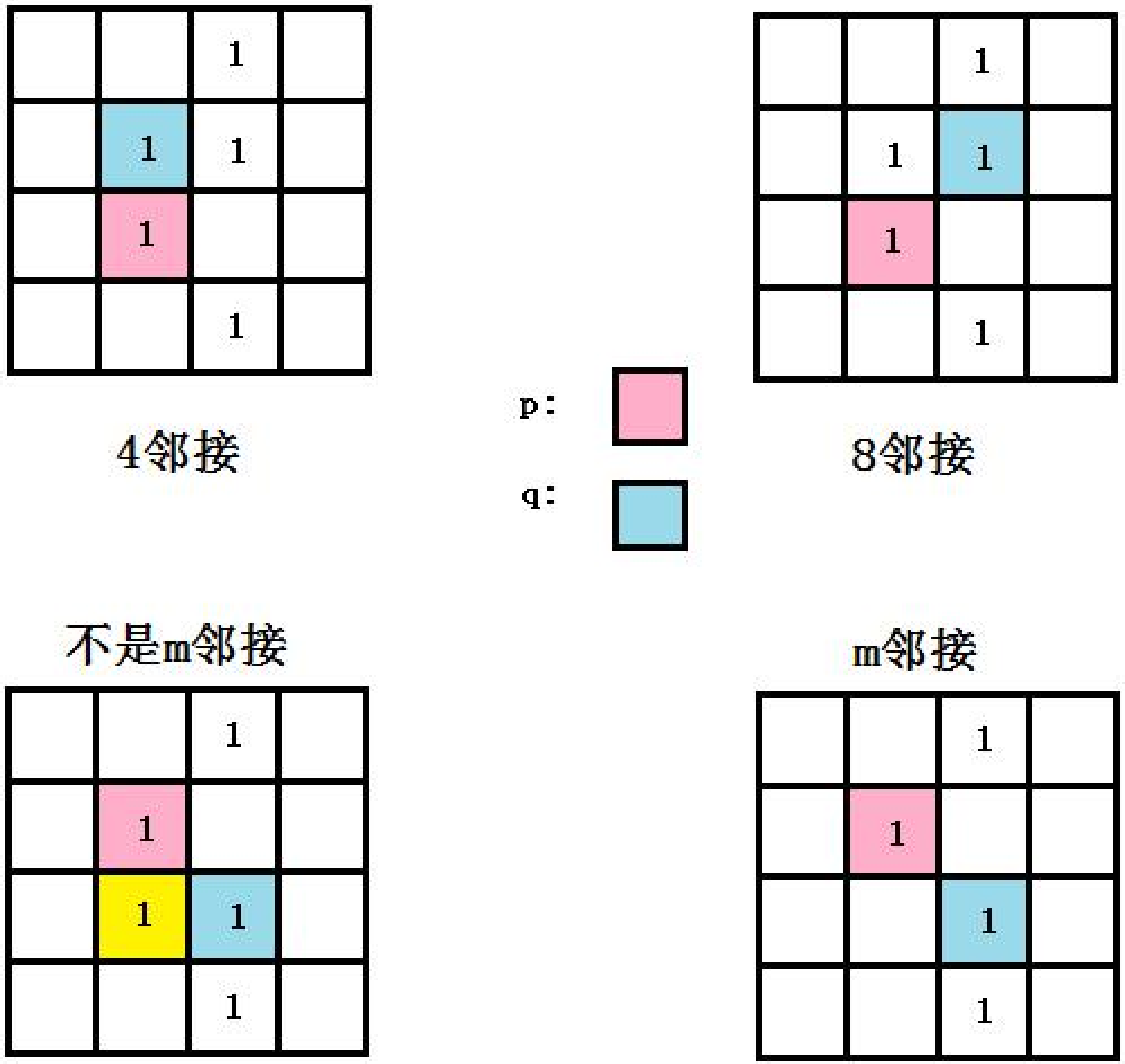


图 2.25　(a)像素的排列；(b) 8 邻接像素(邻接性由虚线所示，注意二义性)；(c) m 邻接；(d)采用 8 邻接时，两个值为 1 的区域是邻接的；(e)如果在区域和背景间采用 8 邻接，则加圆圈的点是仅赋 1 值的像素的边界点的一部分；(f) 1 值区域的内部边界不形成闭合通路，但其外部边界可以形成闭合通路

2.5 像素间的一些基本关系

- **邻接性、连通性、区域和边界**
  - 邻接像素：在二值图像中，把具有1值的像素归诸于邻接像素，V={1}。在灰度图像中，V可以是0-255之间灰度的子集。
  - 4邻接：如果q在集合$N_4(p)$中，则具有V中数值的两个像素p和q是4邻接的；
  - 8邻接：如果q在集合$N_8(p)$中，则具有V中数值的两个像素p和q是8邻接的；
  - *m*邻接（混合邻接）：如果（i）q在$N_4(p)$中，或（i）q在$N_D(p)$中，且集合 $N_4(p) \cap N_4(q)$ 中更没有来自V中数值的像素，则具有V中数值的两个像素p和q是m邻接的。



2.5 像素间的一些基本关系

- **邻接性、连通性、区域和边界**
  - 二义性
    - 8邻接：A到C的方式有两种，①是A→B→C，②是A→C

0	B	C
0	A	0
0	0	D

0	1	1
0	1	0
0	0	1

  - m邻接（混合邻接）可以消除8邻接的二义性
    - A和C的四邻域交集中，B为1（属于集合V），故A和C不是m邻接，则说明A无法直接连通C。
    - A到C只有一条路，即A→B→C

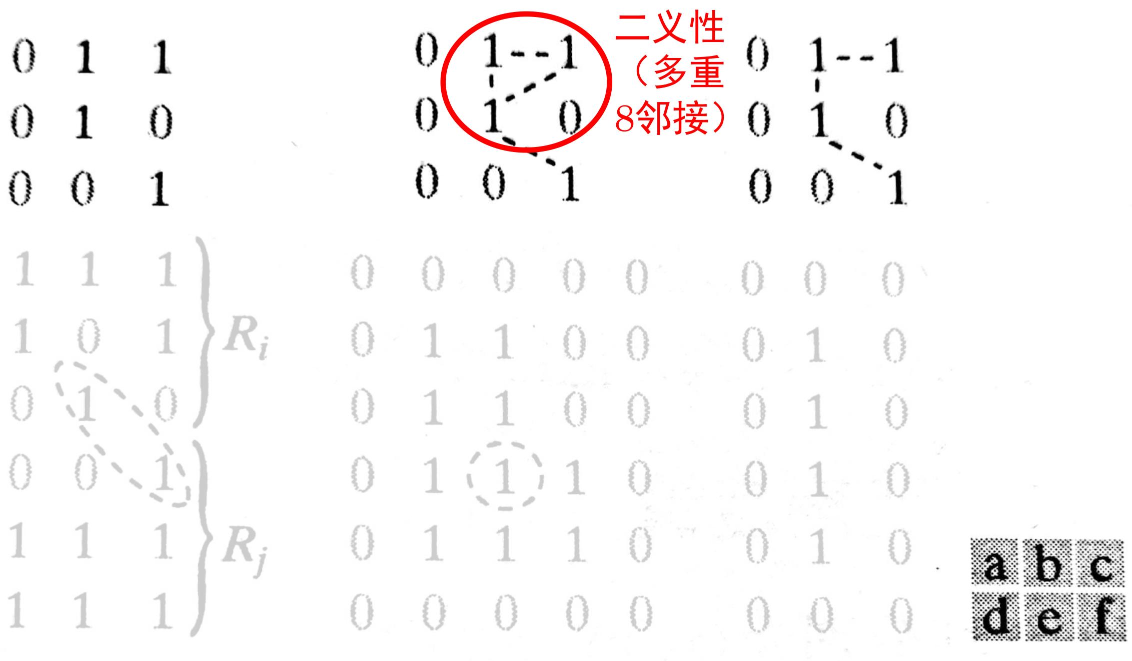


图 2.25 (a)像素的排列；(b) 8 邻接像素(邻接性由虚线所示，注意二义性)；(c) m 邻接；(d)采用 8 邻接时，两个值为 1 的区域是邻接的；(e)如果在区域和背景间采用 8 邻接，则加圆圈的点是仅赋 1 值的像素的边界点的一部分；(f) 1 值区域的内部边界不形成闭合通路，但其外部边界可以形成闭合通路

2.5 像素间的一些基本关系

• 邻接性、连通性、区域和边界

- **连通性：**
  - 连通：通路上的所有像素灰度值满足相似准则，即(xi,yi)与(xi-1,yi-1)邻接
  - 种类：4-连通，8-连通，m-连通
  - 闭合通路、8通路(b)、m通路(c)

- 例如右图：如果要从像素s到像素t：
  - 4连通：s不能到t，因为中心像素和右下角像素不满足4邻接关系。
  - 8连通：s可以到t
  - m连通：s可以到t

0	1	1	s
0	1	0	
0	0	1	t

2.5 像素间的一些基本关系

- **邻接性、连通性、区域和边界**
  - 连通分量
    - 令S是图像中的一个像素子集。
    - 如果S的全部像素之间存在一个通路，则可以说两个像素p和q在S中是连通的。
    - 对于S中任何元素p，S中连通到该像素集称为S的连通分量。
  - 连通集
    - 如果S仅有一个连通分量，即S中所有像素都互相连通，则集合S称为连通集。

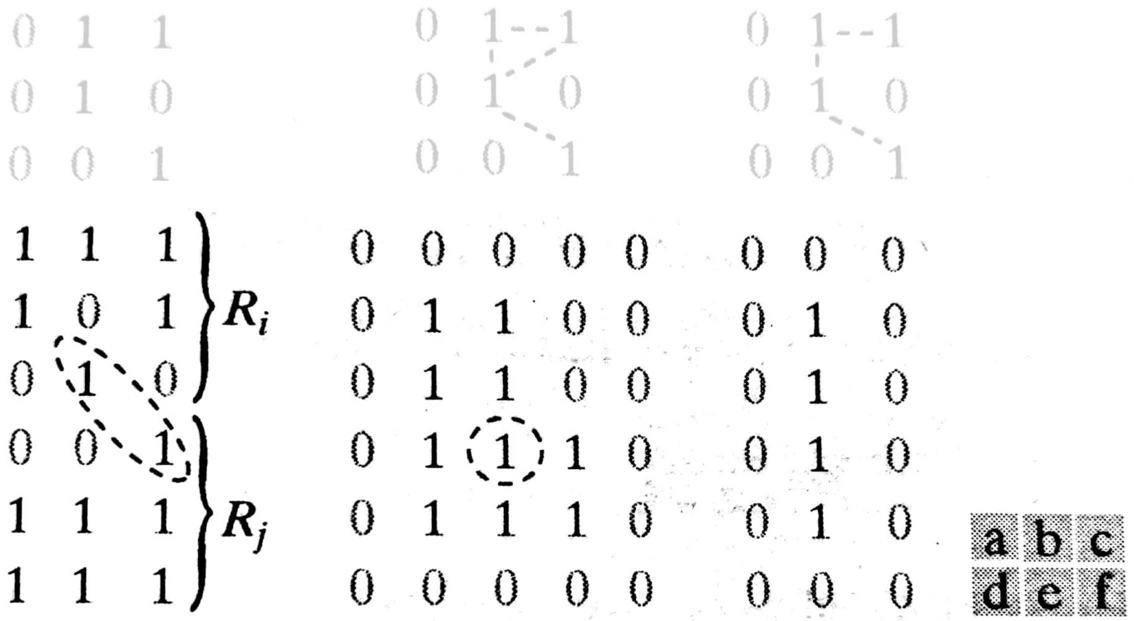
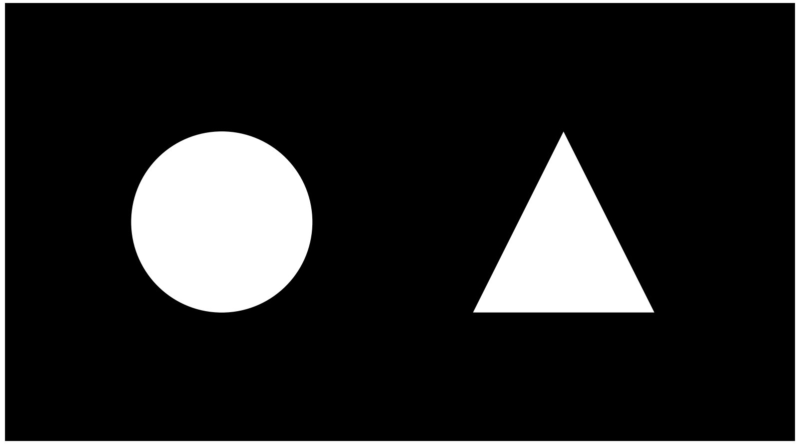
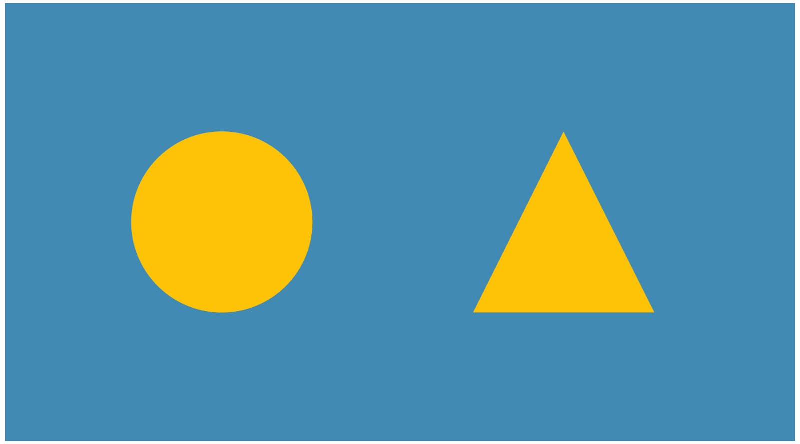


图 2.25　(a)像素的排列；(b) 8 邻接像素(邻接性由虚线所示，注意二义性)；(c) m 邻接；(d)采用 8 邻接时，两个值为 1 的区域是邻接的；(e)如果在区域和背景间采用 8 邻接，则加圆圈的点是仅赋 1 值的像素的边界点的一部分；(f) 1 值区域的内部边界不形成闭合通路，但其外部边界可以形成闭合通路

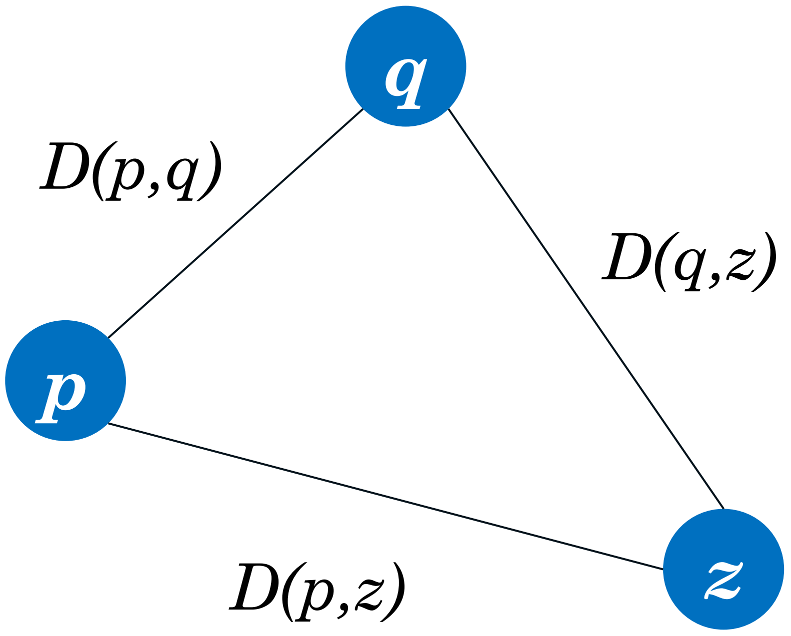
2.5 像素间的一些基本关系

- **邻接性、连通性、区域和边界**
  - 区域
    - 令R是图像中的一个像素子集。
    - 如果R是连通集，则称R为一个区域。
    - 在谈区域时，必须指定邻接的类型（4邻接或8邻接）。
  - 邻接区域、不邻接区域
  - 前景、背景
  - 边界
    - 一个R的边界（也称为边缘或轮廓）是区域中像素的集合。
    - 内边界、外边界





- **距离度量**
  - 距离测度D满足三个条件：
    - $D(p,q) \geqq 0$[$D(p,q)=0$，当且仅当 $p=q$]
    - $D(p,q) = D(q,p)$
    - $D(p,z) \leqq D(p,q) + D(q,z)$



2.5 像素间的一些基本关系

• 距离度量

距离与邻域的关系？

- 欧氏距离（也是范数为2的距离）

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

- 城区距离（也是范数为1的距离）

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

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

- 棋盘距离（也是范数为 ∞ 的距离）

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

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

2.4 像素间联系

• 距离计算示例

- 在图2.4.3中，两个像素p和q之间的DE距离为5（见图2.4.3(a)），D4距离为7（见图2.4.3(b)），D8距离为4（见图2.4.3(c)）

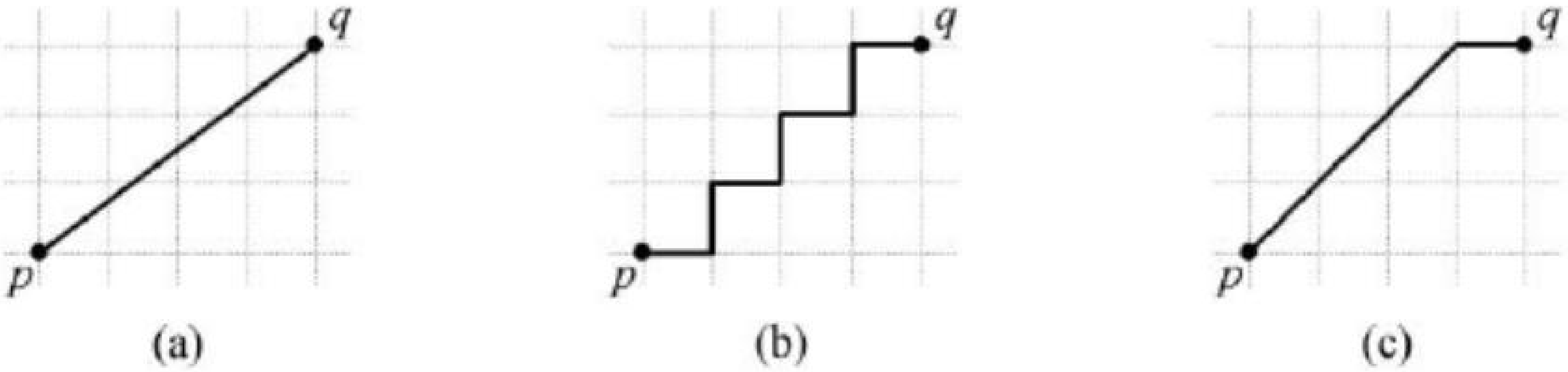
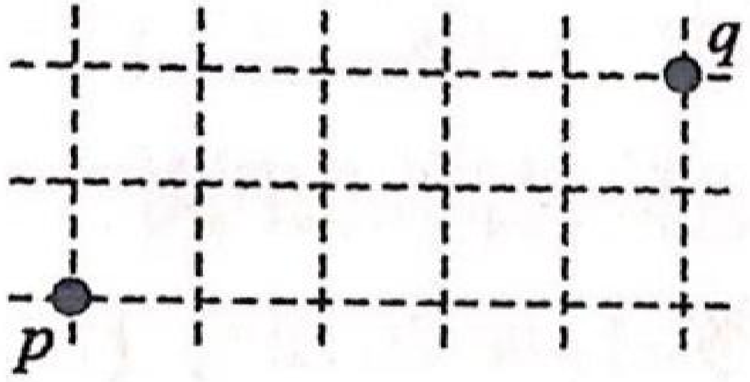


图 2.4.3　像素间距离的计算

2.5 像素间的一些基本关系

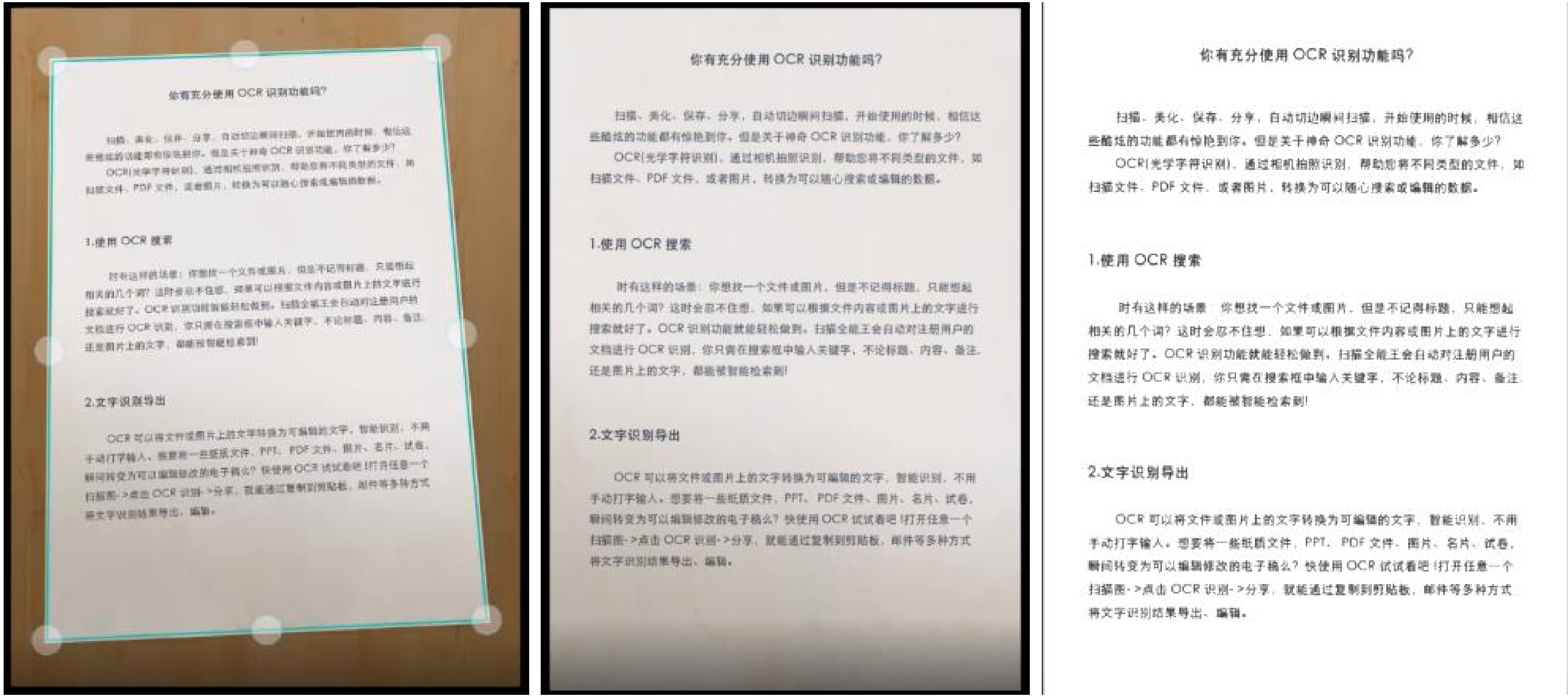
- **练习题**
  - 计算如图所示的两个像素 p 和 q 之间的
    - D_E 距离
    - D_4 距离
    - D_8 距离



目录

思考

- **如何实现以下功能？**


你有充分使用 OCR 识别功能吗?
扫描、美化、保存、分享，自动切边瞬间扫描，开始使用的时候，相信这些酷炫的功能都有惊艳到你。但是关于神奇 OCR 识别功能，你了解多少?
OCR(光学字符识别)，通过相机拍照识别，帮助您将不同类型的文件，如扫描文件、PDF 文件、或者图片，转换为可以随心搜索或编辑的数据。
1.使用 OCR 搜索
时有这样的场景：你想找一个文件或图片，但是不记得标题，只能想起相关的几个词? 这时会忍不住想，如果可以根据文件内容或图片上的文字进行搜索就好了。OCR 识别功能就能轻松做到。扫描全能王会自动对注册用户的文档进行 OCR 识别，你只需在搜索框中输入关键字，不论标题、内容、备注，还是图片上的文字，都能被智能检索到!
2.文字识别导出
OCR 可以将文件或图片上的文字转换为可编辑的文字，智能识别，不用手动打字输入。想要将一些纸质文件，PPT、PDF 文件、图片、名片、试卷，瞬间转变为可以编辑修改的电子稿么? 快使用 OCR 试试看吧!打开任意一个扫描图->点击 OCR 识别->分享，就能通过复制到剪贴板，邮件等多种方式将文字识别结果导出、编辑。

2.6 数字图像处理中所用数学工具

- **阵列与矩阵操作**
  - 考虑下面的2x2图像

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} 和 \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

这两幅图像的阵列相乘是

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

另一方面，矩阵相乘由下式给出：

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

2.6 数字图像处理中所用数学工具

- **数值运算**
  - 加减乘除等
  - 图像模糊、去噪等

A B A+B

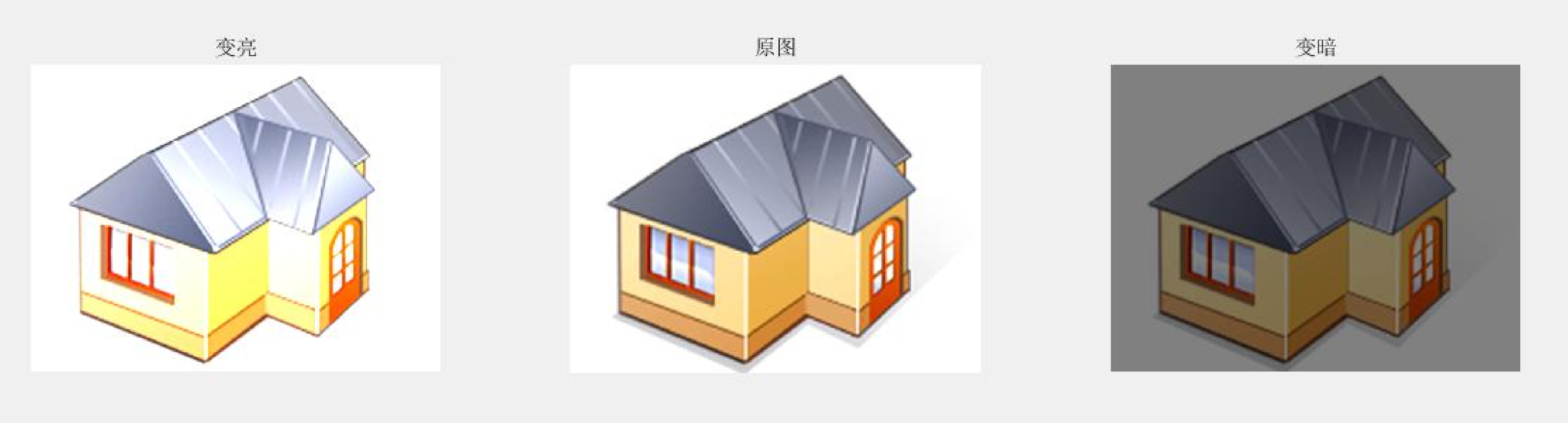
• 数值运算

- 进行基于常用对数的非线性灰度变换
  - 图像通过对数变换可扩展低值灰度，压缩高值灰度。

H = (log(J+1)) / 10;

2.6 数字图像处理中所用数学工具

- **数值运算**
  - 利用图像乘法运算实现图像亮度的控制



2.6 数字图像处理中所用数学工具

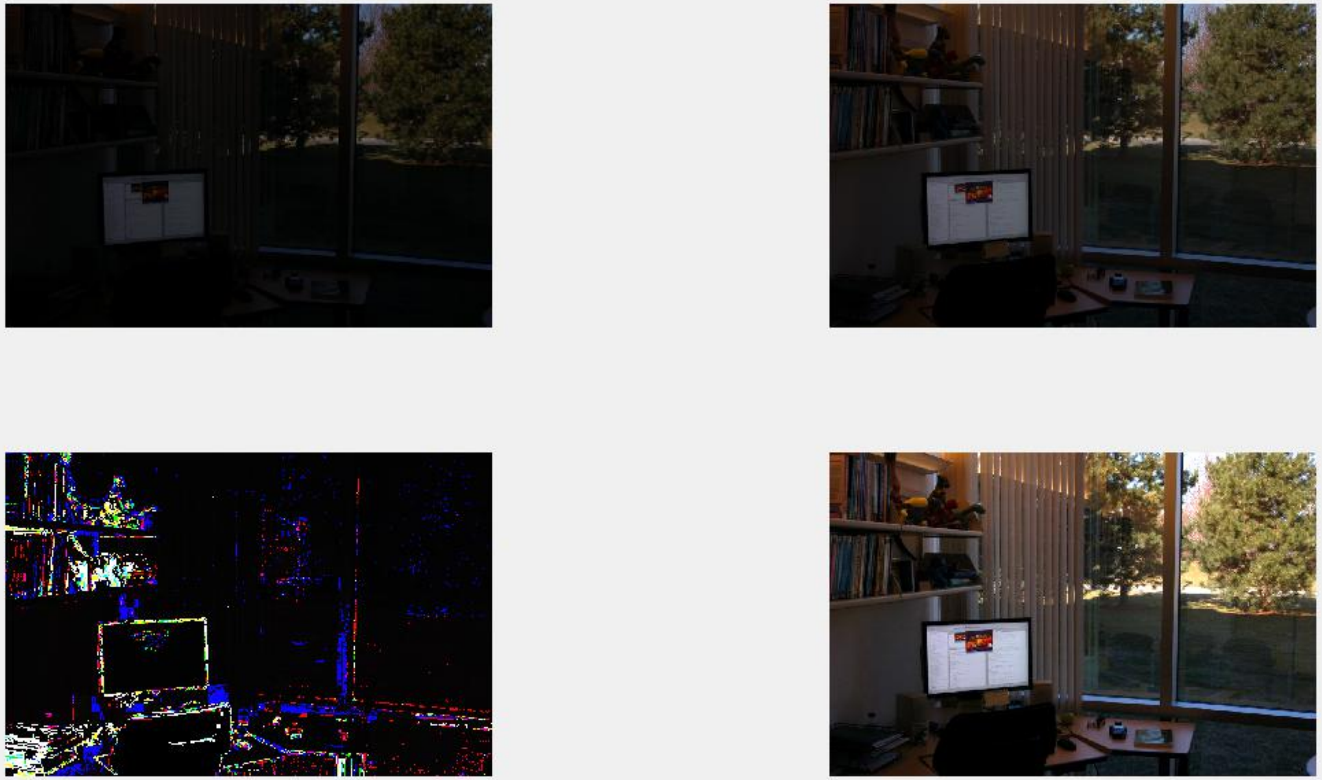
- **数值运算**
  - 图像的除法运算给出的是两幅图像相应像素值的变化比率，常用于校正成像设备的非线性影响。

```
close all; clear all; clc;

I = imread('office_1.jpg');
J = imread('office_2.jpg');

K1 = imdivide(J, I); % 两幅图像相除
K2 = imdivide(J, 0.5); % 一幅图像除以一个常数

figure;
subplot(221), imshow(I);
subplot(222), imshow(J);
subplot(223), imshow(K1);
subplot(224), imshow(K2);
```



2.6 数字图像处理中所用数学工具

- **逻辑运算**
  - 与/交
  - 或/并
  - 非/补
  - 或非
  - 与非

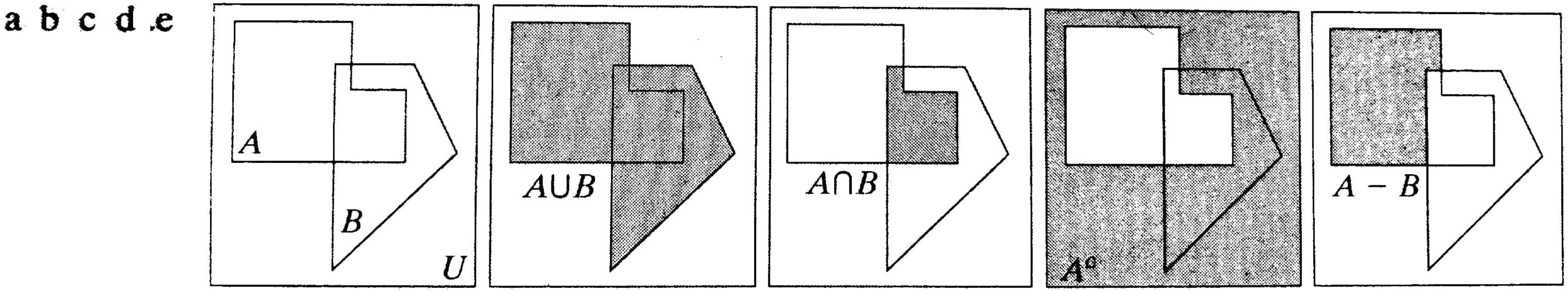
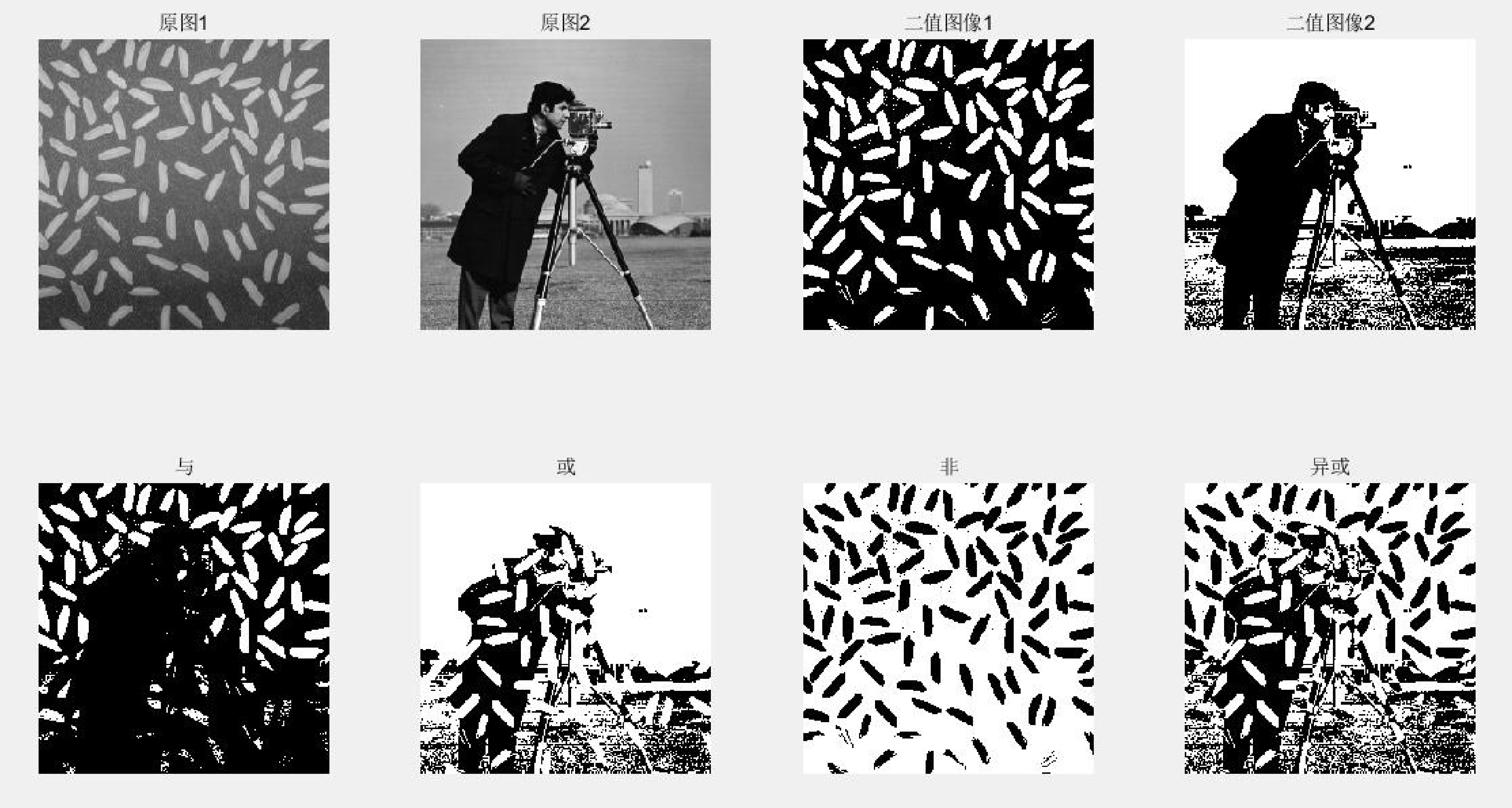


图 2.31 (a)二维空间中的两个坐标集合 A 和 B；(b)A 和 B 的并集；(c)A 和 B 的交集；(d)A 的补集；(e)A 和 B 的差。在(b)到(e)中，阴影区域表示指定集合操作的成员

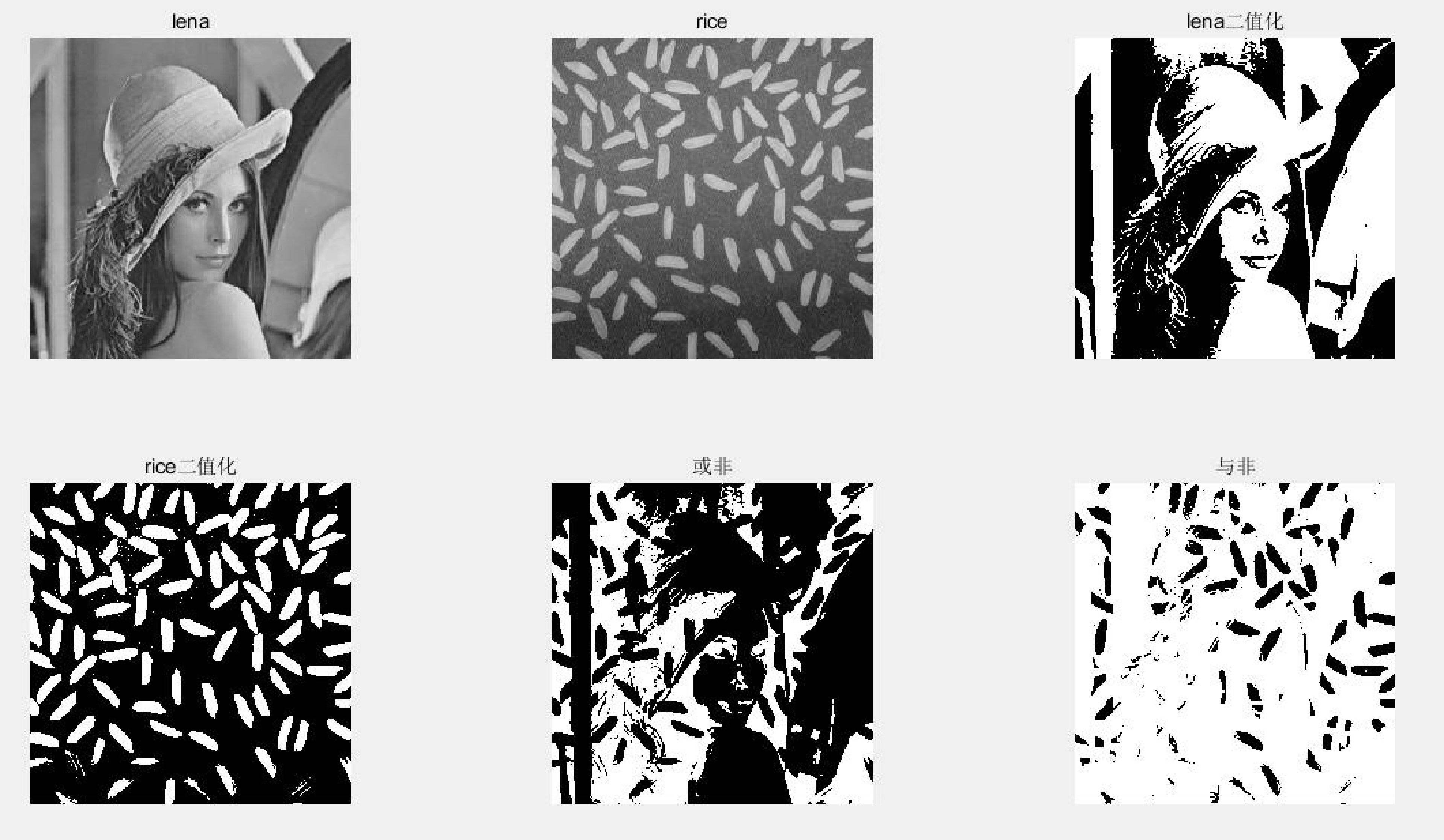
2.6 数字图像处理中所用数学工具

- **逻辑运算**
  - 与/交
  - 或/并
  - 非/补
  - 或非
  - 与非



2.6 数字图像处理中所用数学工具

- **逻辑运算**
  - 与/交
  - 或/并
  - 非/补
  - 或非
  - 与非

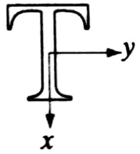
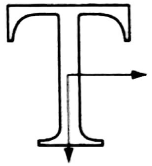
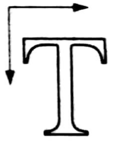
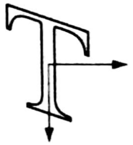
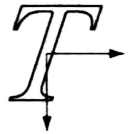


2.6 数字图像处理中所用数学工具

- **基本坐标变换**
  - 坐标变换可借助矩阵写为：

$$\mathbf{v}' = \mathbf{T}\mathbf{v}$$

表 2.2　基于式(2.6-23)的仿射变换

变换名称	仿射矩阵 T	坐标公式	例子
恒等变换	$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$	$x = v$ $y = w$	
尺度变换	$\begin{bmatrix}c_x&0&0\\0&c_y&0\\0&0&1\end{bmatrix}$	$x = c_x v$ $y = c_y w$	
旋转变换	$\begin{bmatrix}\cos\theta&\sin\theta&0\\-\sin\theta&\cos\theta&0\\0&0&1\end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\sin\theta + w\cos\theta$	
平移变换	$\begin{bmatrix}1&0&0\\0&1&0\\t_x&t_y&1\end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
(垂直)偏移变换	$\begin{bmatrix}1&0&0\\s_v&1&0\\0&0&1\end{bmatrix}$	$x = vs_v + w$ $y = w$	
(水平)偏移变换	$\begin{bmatrix}1&s_h&0\\0&1&0\\0&0&1\end{bmatrix}$	$x = v$ $y = s_h v + w$	

2.5 图像坐标变换

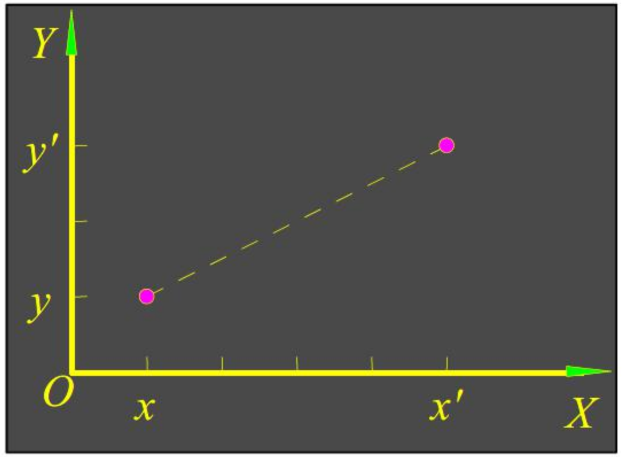
- ## 基本坐标变换

  - 平移变换矩阵　　　　平移变换的逆矩阵

$$
\boldsymbol{T}=\begin{bmatrix}1 & 0 & t_x\\ 0 & 1 & t_y\\ 0 & 0 & 1\end{bmatrix}
\qquad
\boldsymbol{T}^{-1}=\begin{bmatrix}1 & 0 & -t_x\\ 0 & 1 & -t_y\\ 0 & 0 & 1\end{bmatrix}
$$

$$
\boldsymbol{T}=\begin{bmatrix}1 & 0 & T_x\\ 0 & 1 & T_y\\ 0 & 0 & 1\end{bmatrix}
$$

$$
\begin{bmatrix}1 & 0 & 4\\ 0 & 1 & 2\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\ 1\\ 1\end{bmatrix}=\begin{bmatrix}5\\ 3\\ 1\end{bmatrix}
$$



- **基本坐标变换**
  - 平移变换矩阵

2.5 图像坐标变换

- ## 基本坐标变换

  - 尺度变换矩阵

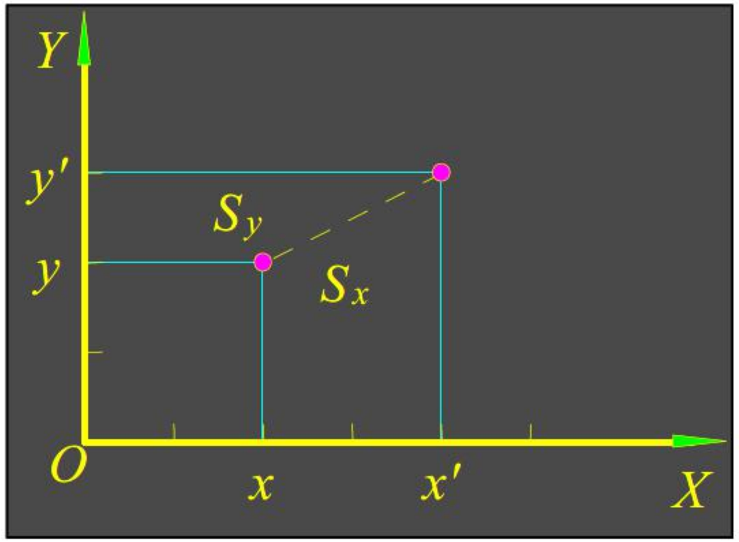
$$S=\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

尺度变换的逆矩阵

$$S^{-1}=\begin{bmatrix} 1/s_x & 0 & t_x \\ 0 & 1/s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$S=\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix}=\begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}$$



- **尺度变换（放缩变换）**

$$\boldsymbol{S}=\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.5 图像坐标变换

- **图像镜像**
  - 镜像变换又分为水平镜像和竖直镜像。
    - 水平镜像即将图像左半部分和右半部分以图像竖直中轴线为中心轴进行对换；
    - 竖直镜像则是将图像上半部分和下半部分以图像水平中轴线为中心轴进行对换.

2.5 图像坐标变换

- **图像镜像**

- 水平镜像的变换关系为：

$$[x_1 \quad y_1 \quad 1]=[x_0 \quad y_0 \quad 1]\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ Width & 0 & 1 \end{pmatrix}=[Width-x_0 \quad y_0 \quad 1]$$

对矩阵求逆得到：$[x_0 \quad y_0 \quad 1]=[x_1 \quad y_1 \quad 1]\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ Width & 0 & 1 \end{pmatrix}=[Width-x_1 \quad y_1 \quad 1]$

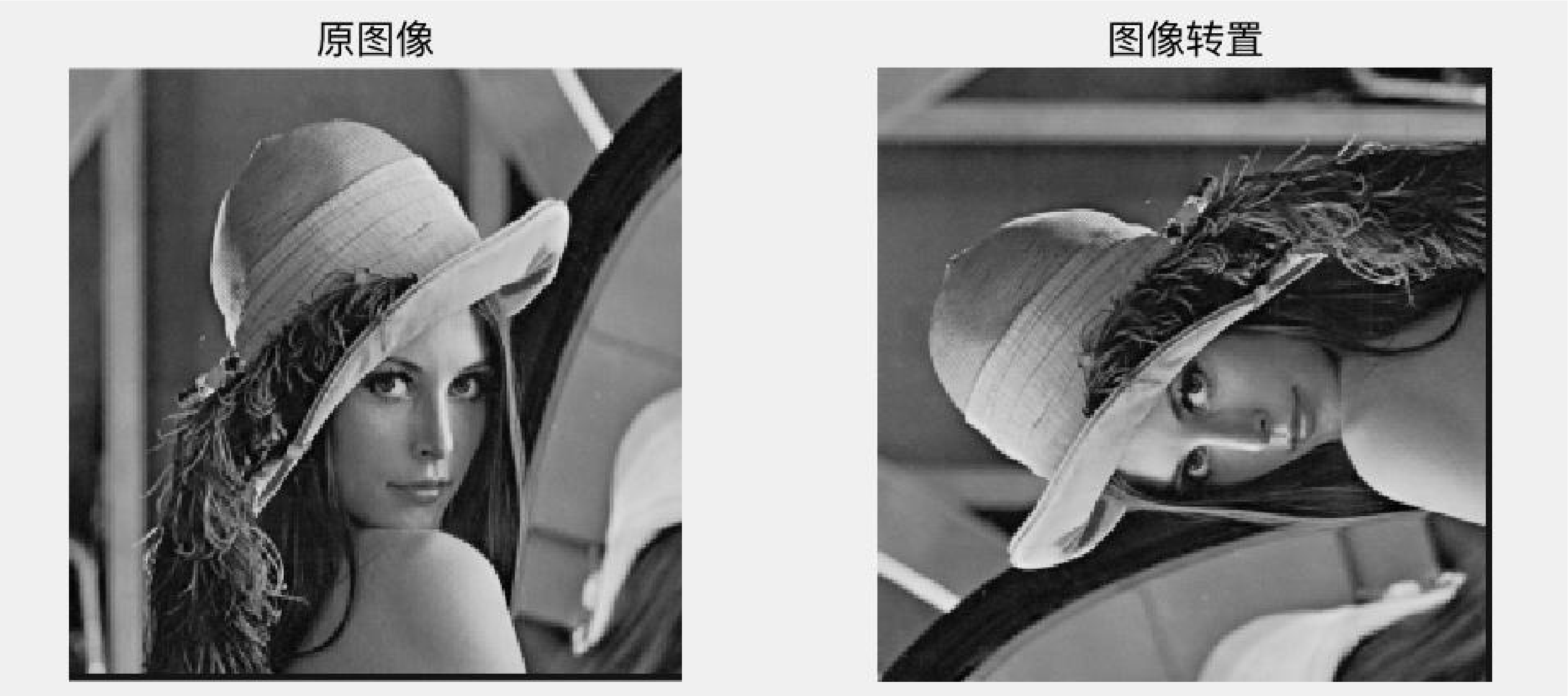
- 竖直镜像变换关系可形式化地描述为：

$$[x_1 \quad y_1 \quad 1]=[x_0 \quad y_0 \quad 1]\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & Height & 1 \end{pmatrix}=[x_0 \quad Height-y_0 \quad 1]$$

逆运算为：$[x_0 \quad y_0 \quad 1]=[x_1 \quad y_1 \quad 1]\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & Height & 1 \end{pmatrix}=[x_1 \quad Height-y_1 \quad 1]$

2.5 图像坐标变换

- **转置变换**
  - 图像转置是指将图像像素的x坐标和y坐标互换，图像的大小会随之改变：高度和宽度将互换



2.5 图像坐标变换

- **转置变换**
  - 图像转置是指将图像像素的x坐标和y坐标互换，图像的大小会随之改变：高度和宽度将互换

$$[x_1 \quad y_1 \quad 1]=[x_0 \quad y_0 \quad 1]\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}=[y_0 \quad x_0 \quad 1]$$

显然，转置矩阵 $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 的逆矩阵仍为其自身。

故转置变换的逆变换具有相同的形式。

2.5 图像坐标变换

- **基本坐标变换**
  - 旋转变换矩阵

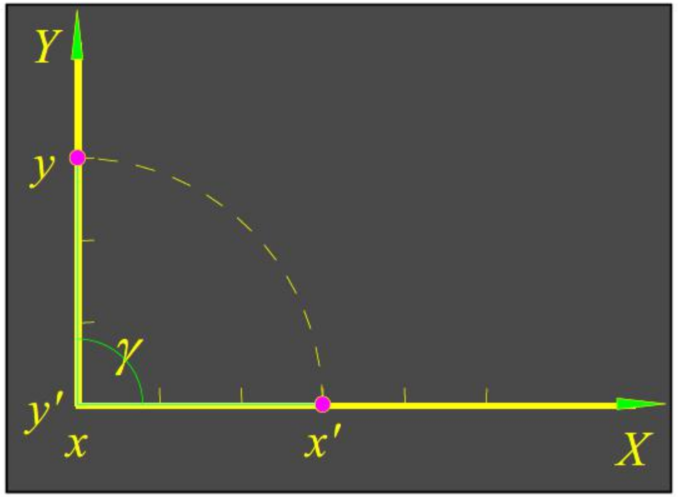
$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

旋转变换的逆矩阵

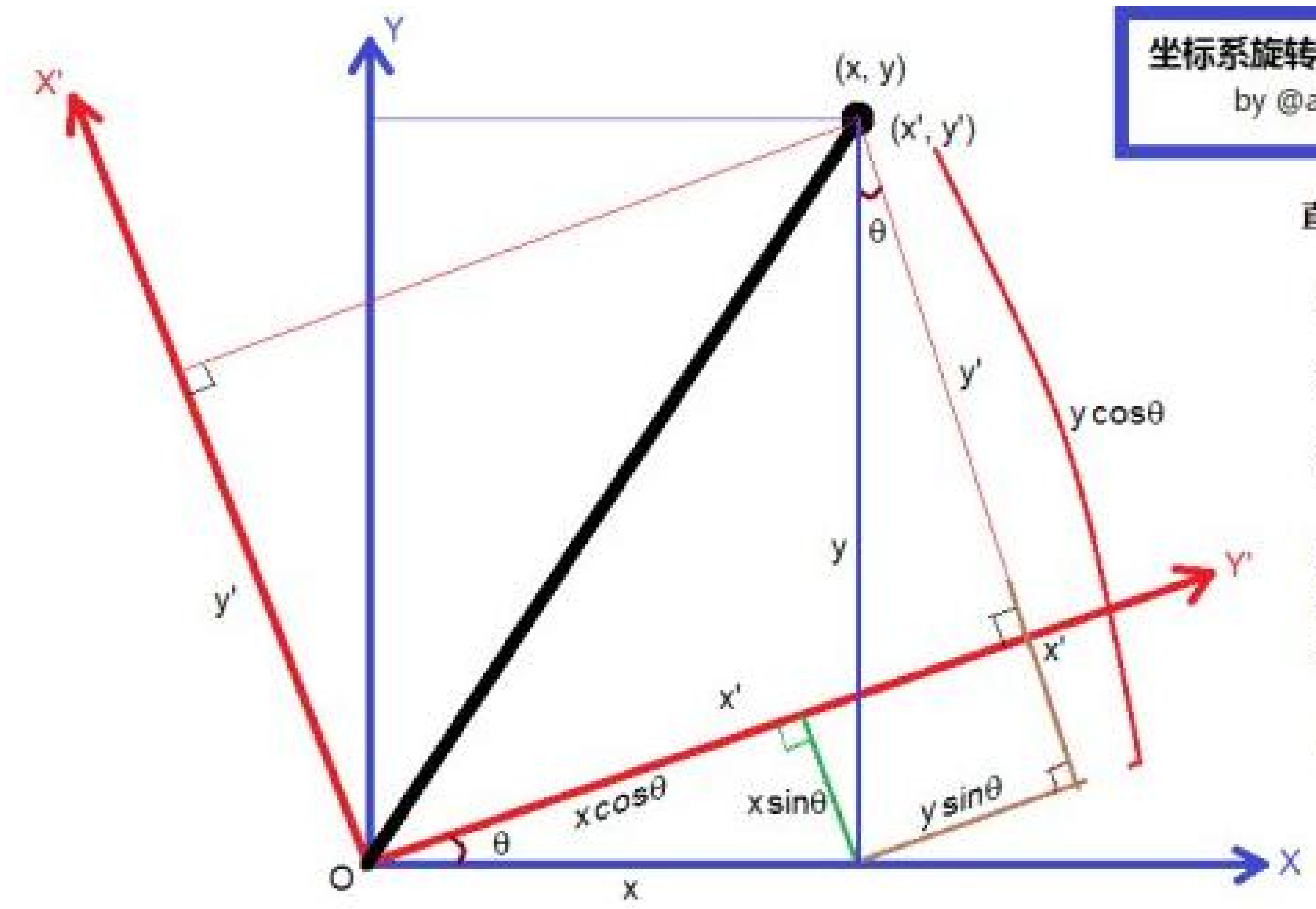
$$R^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\gamma = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 \\ -\sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 1 \end{bmatrix}$$



2.5 图像坐标变换



坐标系旋转变换公式图解

by @abada张宏兵

直角坐标系旋转θ角后，新旧坐标变换公式：

由图显然可知：

$$x' = x\cos\theta + y\sin\theta$$

$$y' = y\cos\theta - x\sin\theta$$

这里的关键是用已知角做出直角三角形，并坚持用已知边（这里是x和y）做斜边，可方便计算，一目了然。

（参见笔者"两角和的正弦和余弦公式的最简单图解"）

2.5 图像坐标变换

- **基本坐标变换：旋转变换**

```
A = imread('lena.bmp');
B = imrotate(A,30,'nearest','crop');
```

$$\gamma = -45^\circ$$

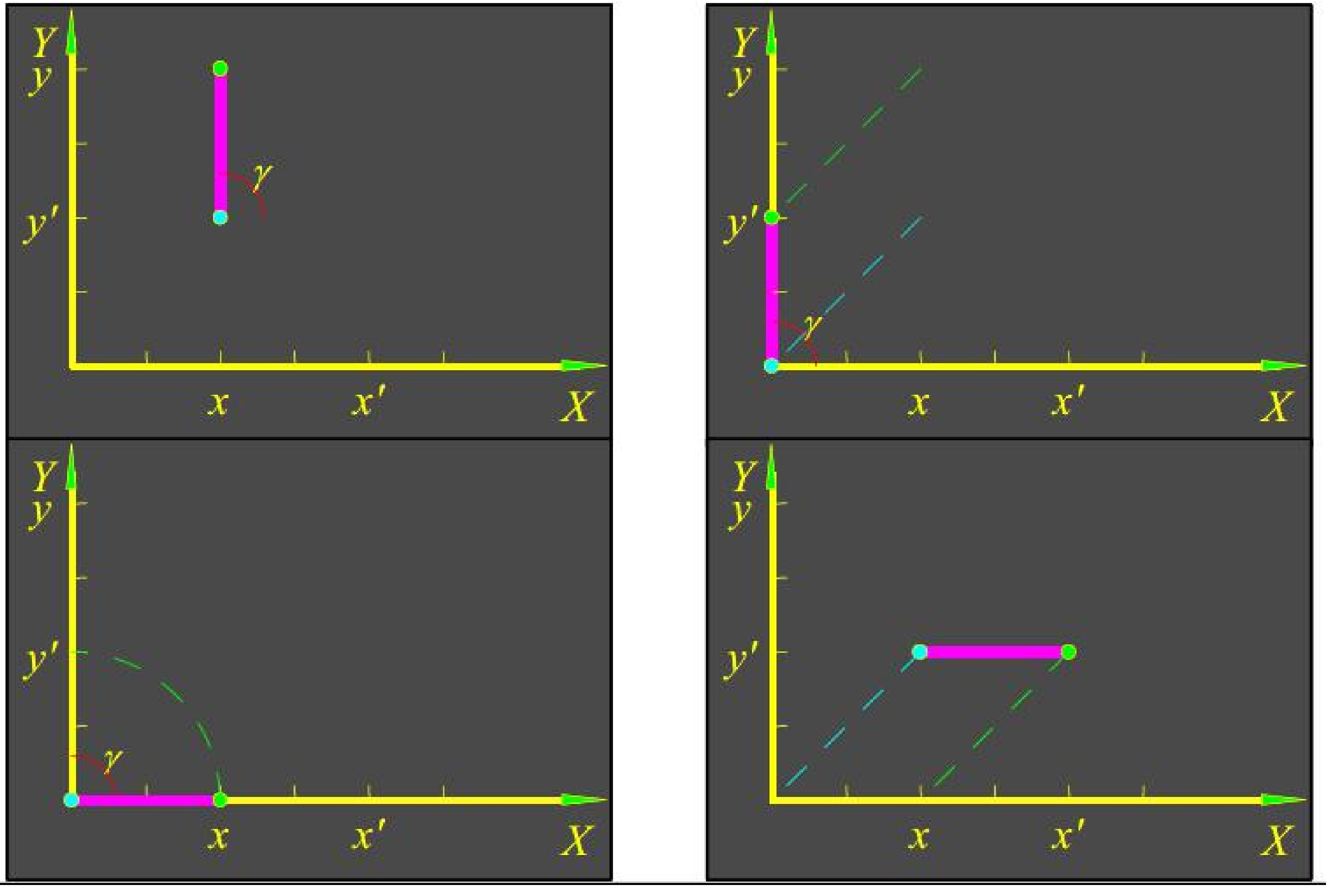
$$\boldsymbol{R} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.5 图像坐标变换

- **基本坐标变换：旋转变换**

旋转轴不在原点

绕非原点旋转
=
平移至原点
+
绕原点旋转
+
平移回去



2.5 图像坐标变换

- **基本坐标变换：变换级联**

对一个坐标为$\boldsymbol{v}$的点的平移、放缩、绕Z轴旋转变换（级联起来）可表示为：

$$\boldsymbol{v}' = \boldsymbol{R}_{\gamma}\left[\boldsymbol{S}(\boldsymbol{T}\boldsymbol{v})\right] = \boldsymbol{A}\boldsymbol{v}$$

等价于用单个变换矩阵$\boldsymbol{A}$对点$\boldsymbol{v}$进行变换

这些矩阵的运算次序一般不可互换

- **变换级联**

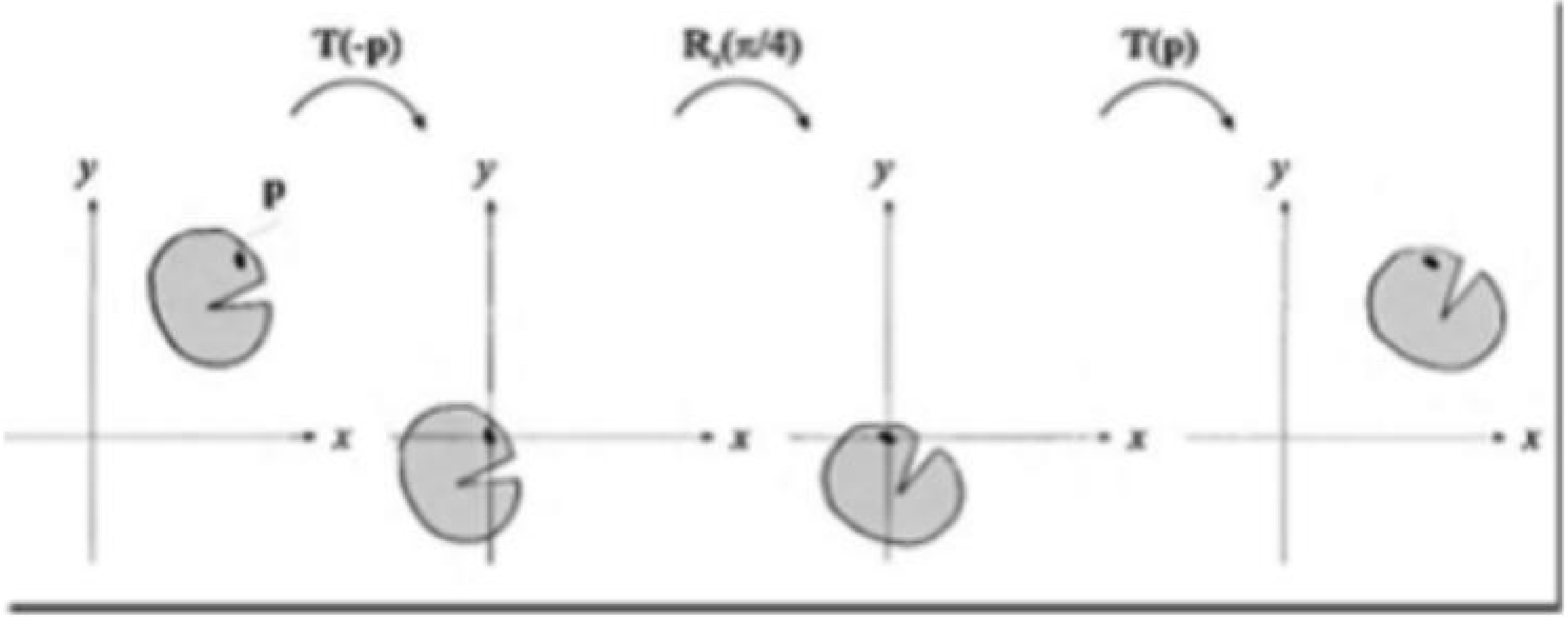


图4.2 绕指定点P旋转的例子。

2.5 图像坐标变换

- **基本坐标变换：变换级联**

$$\boldsymbol{T}=\begin{bmatrix}1 & 0 & 2\\ 0 & 1 & 4\\ 0 & 0 & 1\end{bmatrix}$$

$$\boldsymbol{S}=\begin{bmatrix}3 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$\boldsymbol{v}=\begin{bmatrix}1 & 2 & 1\end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{v}'_{ST}=\begin{bmatrix}9 & 12 & 1\end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{v}'_{TS}=\begin{bmatrix}5 & 8 & 1\end{bmatrix}^{\mathrm{T}}$$

先平移后放缩 $\neq$ 先放缩后平移

2.5 图像坐标变换

- **仿射变换**
  - 一个平面上的仿射变换有6个自由度

$$A=\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

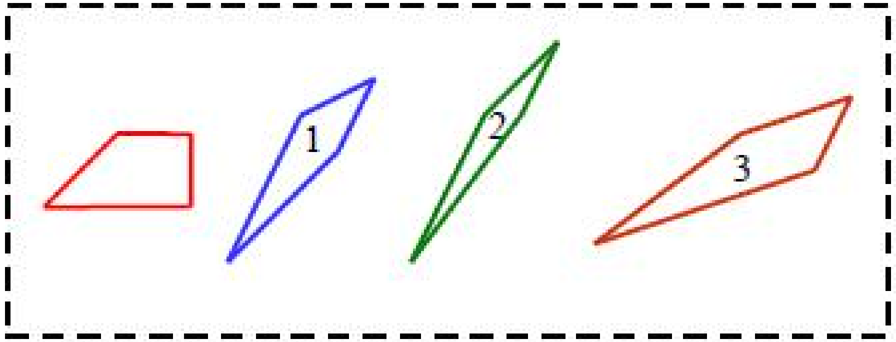


图 2.5.1　对一个多边形图形分别进行三次仿射变换得到的结果

2.5 图像坐标变换

- **仿射变换**
  - 仿射变换的一种特例是**欧氏变换**

$$E=\begin{bmatrix}\cos\theta & \sin\theta & t_x \\ -\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1\end{bmatrix}$$

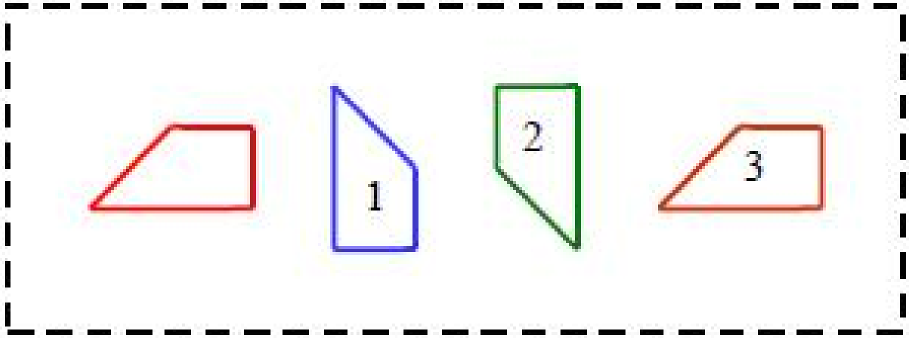


图 2.5.2 对一个多边形图形分别进行三次欧氏变换得到的结果

- **仿射变换**
  - **相似变换**也是仿射变换的一种特例

$$X = \begin{bmatrix} s\cos\theta & s\sin\theta & t_x \\ -s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

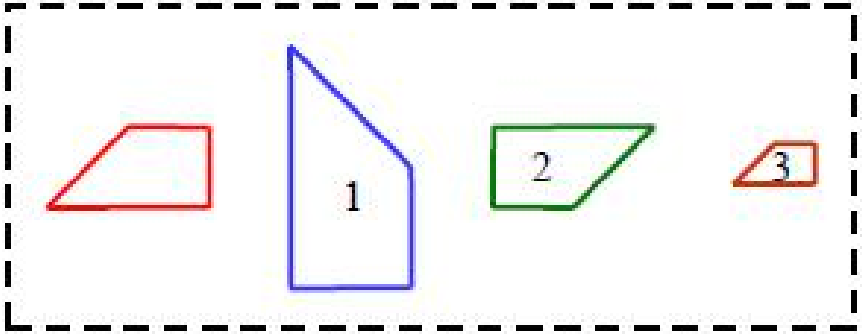
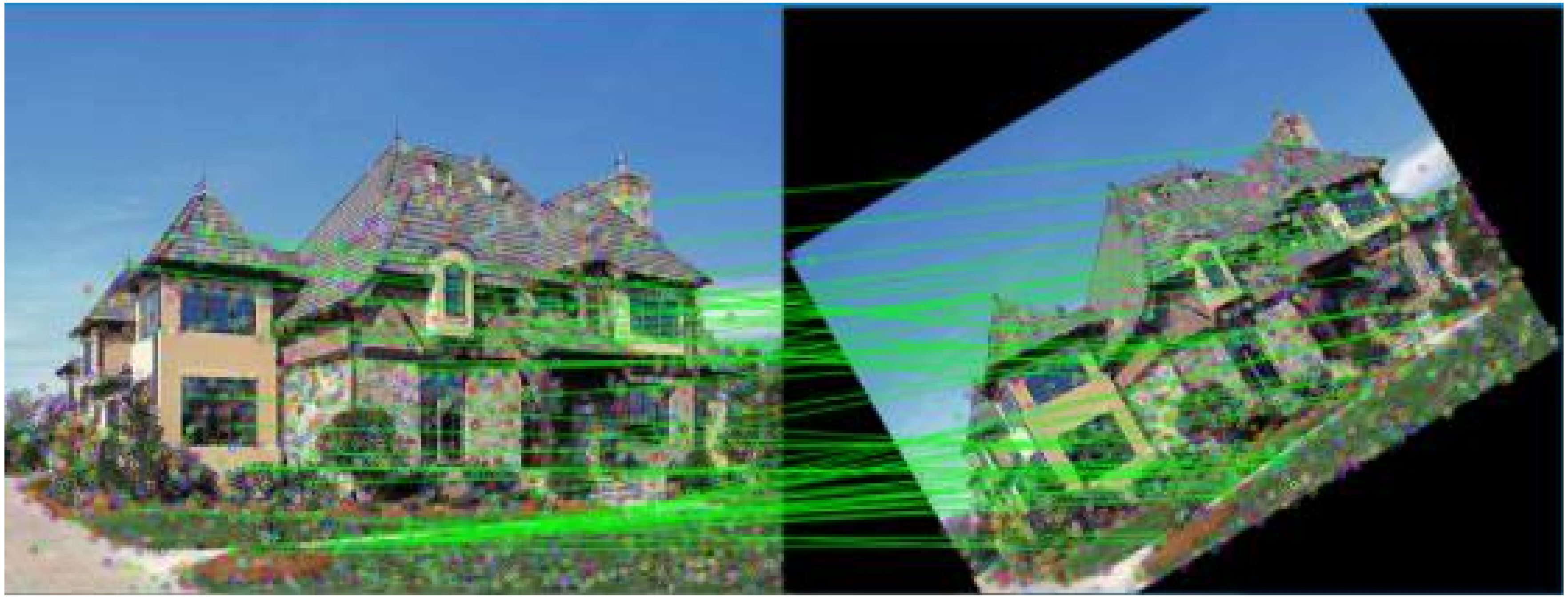


图 2.5.3 对一个多边形图形分别进行三次相似变换得到的结果

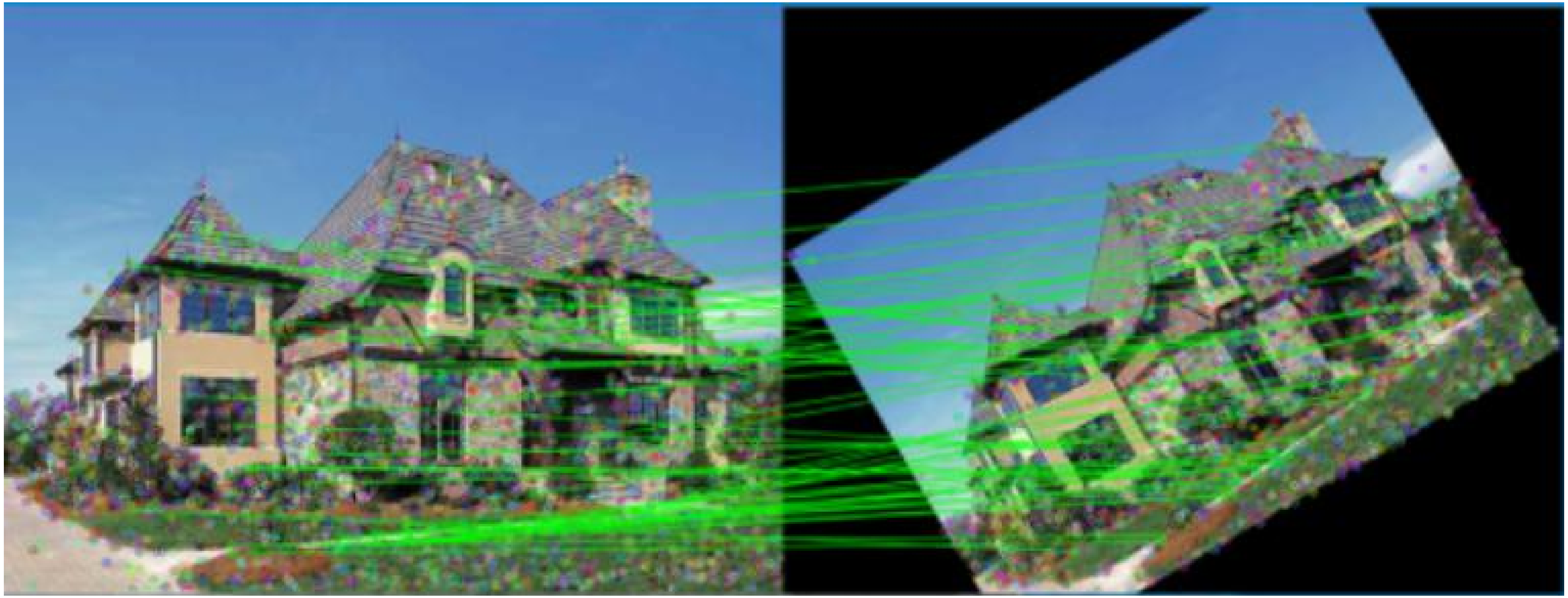
几何失真校正

- **两个步骤：**
  - 计算空间变换函数
  - 插值填充



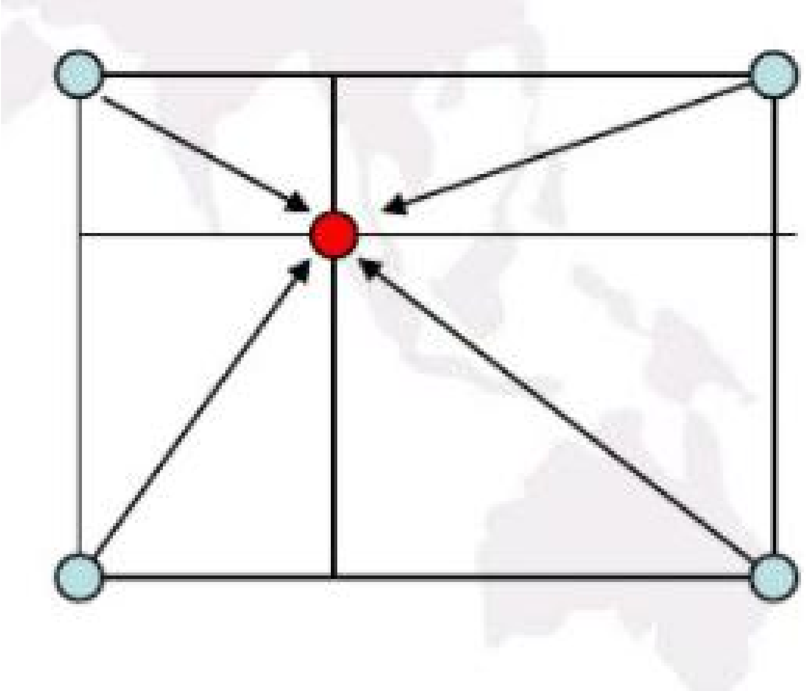
计算空间变换

- 问题：变换矩阵未知，其参数通过若干已知的匹配对应点后求解得到



插值填充

- **问题：为什么需要插值填充？**
  - 变换后的坐标 (x',y')可能不是整数，如何取整？
    - 最近邻插值：使用离(x’,y’)最近的像素的值，不够精确
    - 双线性插值：使用(x’,y’)的4邻域的像素值插值计算



2.5 图像坐标变换

- ## 几何失真校正
  - **空间变换**：对图像平面上的像素进行重新排列以恢复像素原空间关系

$$x' = s(x, y) \qquad s(x, y) = k_1 x + k_2 y + k_3$$

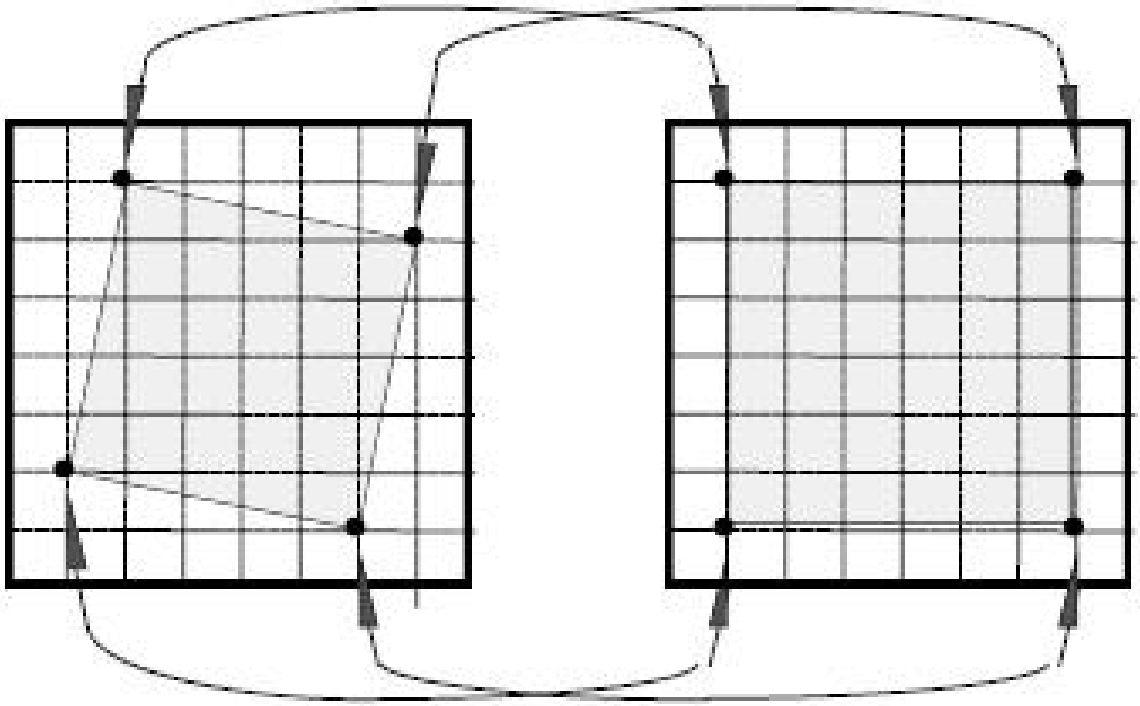
$$y' = t(x, y) \qquad t(x, y) = k_4 x + k_5 y + k_6$$

$$s(x, y) = k_1 + k_2 x + k_3 y + k_4 x^2 + k_5 xy + k_6 y^2$$

$$t(x, y) = k_7 + k_8 x + k_9 y + k_{10} x^2 + k_{11} xy + k_{12} y^2$$

2.5 图像坐标变换

- ## 几何失真校正
  - 一个在失真图上的四边形区域和在校正图上与其对应的四边形区域的顶点可作为对应点



$$x' = k_1 x + k_2 y + k_3 xy + k_4$$

$$y' = k_5 x + k_6 y + k_7 xy + k_8$$

图 2.5.4 失真图和校正图的对应点

- **几何失真校正**
  - **灰度插值（最近邻插值/零阶插值）**：对空间变换后的像素赋予相应的灰度值以恢复原位置的灰度值

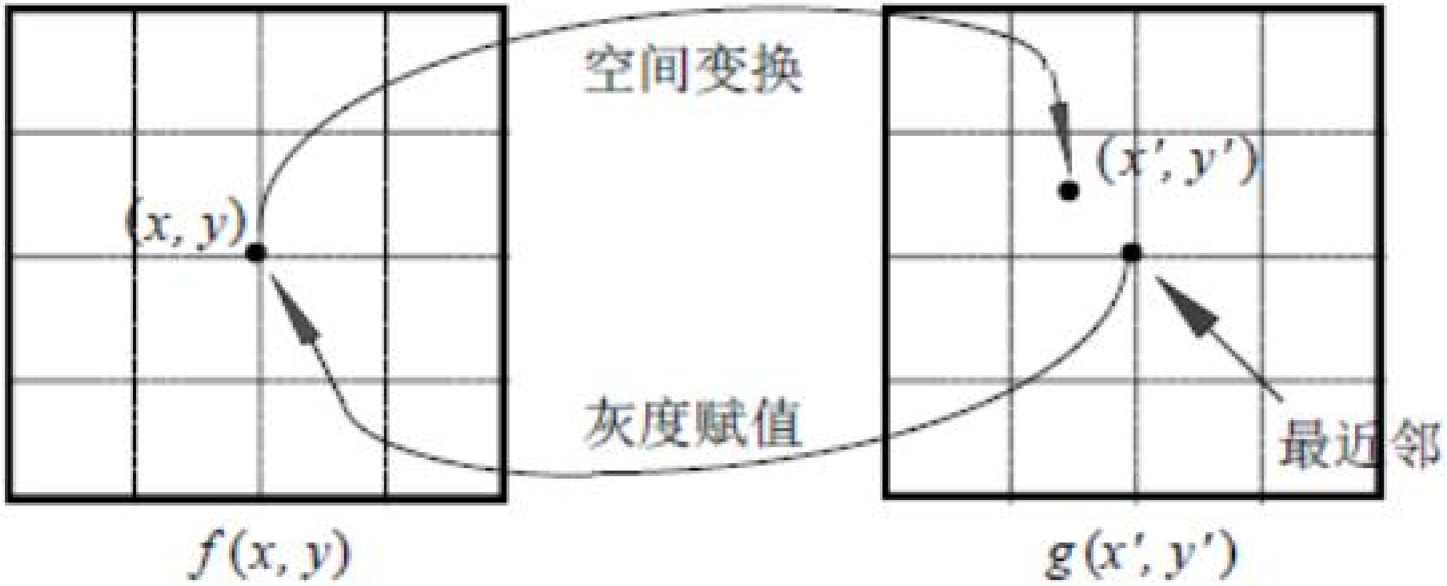
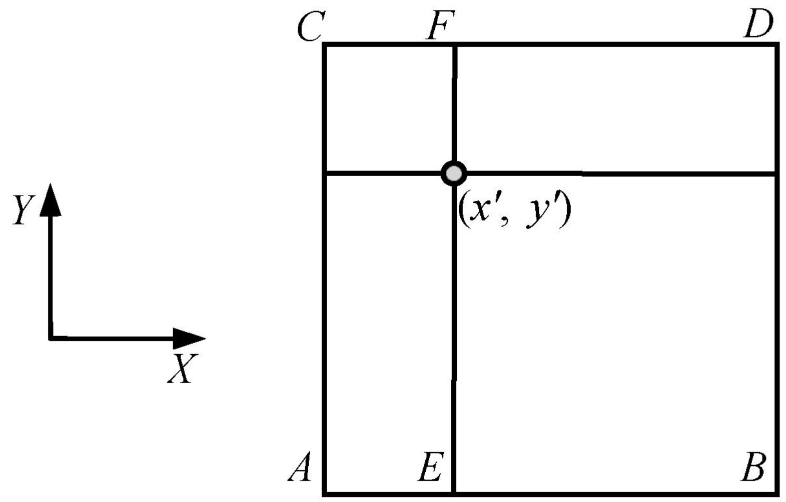


图 2.5.5 灰度插值示意图

• 几何失真校正

▪ 双线性插值：

- 利用 (x', y')点的4个最近邻像素的灰度值来计算 (x', y')点处的灰度值



$$g(x',y')=(y'-j)[g(F)-g(E)]+g(E)$$

总 结

- 2.1 视觉感知要素
- 2.2 光和电子波谱
- 2.3 图像感知和获取
- 2.4 图像取样和量化
- 2.5 像素间的一些基本关系
- 2.6 数字图像处理中所用数学工具

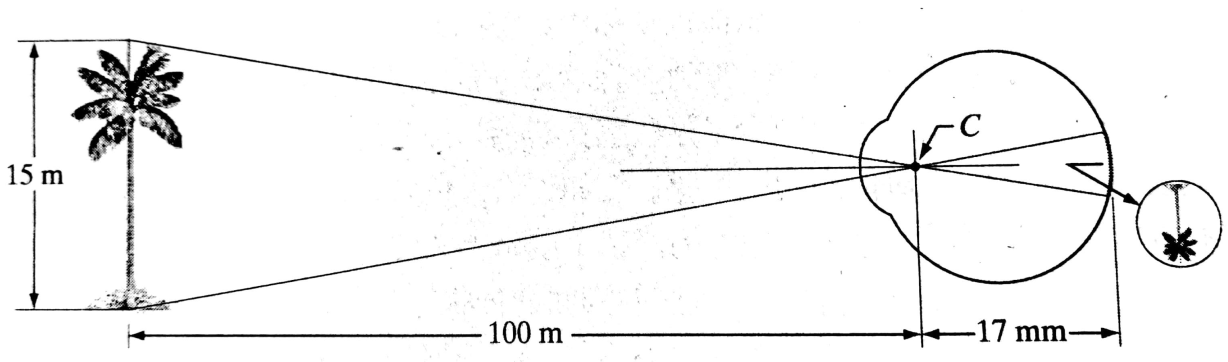
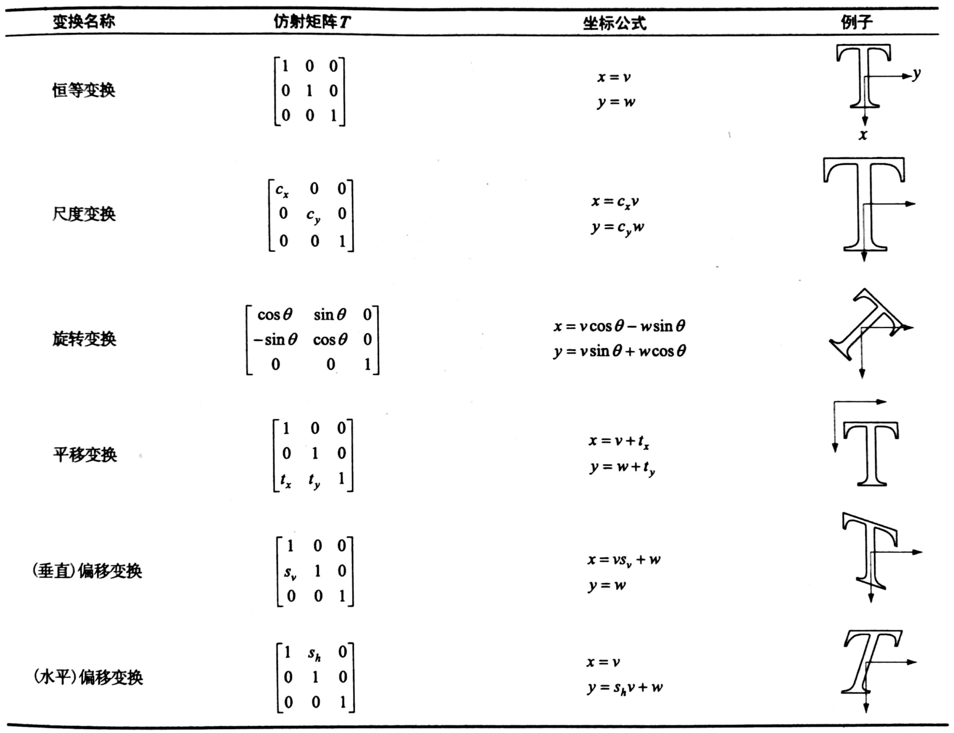
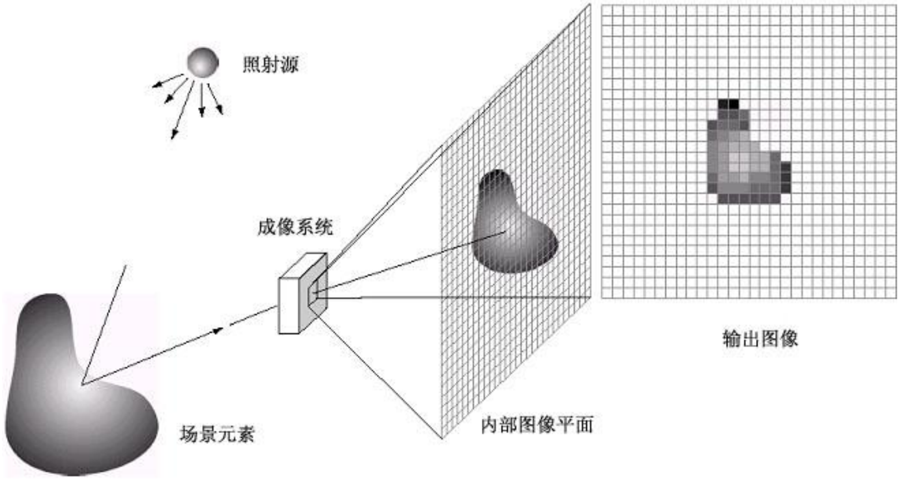
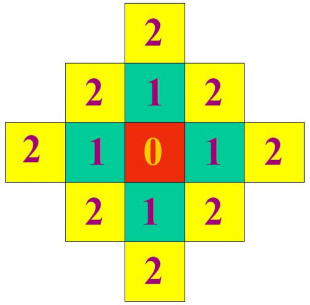


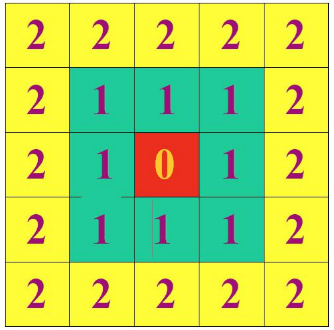
表 2.2 基于式(2.6-23)的仿射变换



变换名称	仿射矩阵 T	坐标公式	例子
恒等变换	$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$	$x=v$ $y=w$	
尺度变换	$\begin{bmatrix}c_x&0&0\\0&c_y&0\\0&0&1\end{bmatrix}$	$x=c_x v$ $y=c_y w$	
旋转变换	$\begin{bmatrix}\cos\theta&\sin\theta&0\\-\sin\theta&\cos\theta&0\\0&0&1\end{bmatrix}$	$x=v\cos\theta-w\sin\theta$ $y=v\sin\theta+w\cos\theta$	
平移变换	$\begin{bmatrix}1&0&0\\0&1&0\\t_x&t_y&1\end{bmatrix}$	$x=v+t_x$ $y=w+t_y$	
(垂直)偏移变换	$\begin{bmatrix}1&0&0\\s_v&1&0\\0&0&1\end{bmatrix}$	$x=vs_v+w$ $y=w$	
(水平)偏移变换	$\begin{bmatrix}1&s_h&0\\0&1&0\\0&0&1\end{bmatrix}$	$x=v$ $y=s_h v+w$	







Thank You !

Q & A