图像采集

高 飞

计算机学院

智能可视建模与仿真实验室

Intelligent Visual Modeling & Simulation (iGame) Lab

目 录

像素间的一些基本关系

邻接关系

- **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$N_8(p)$

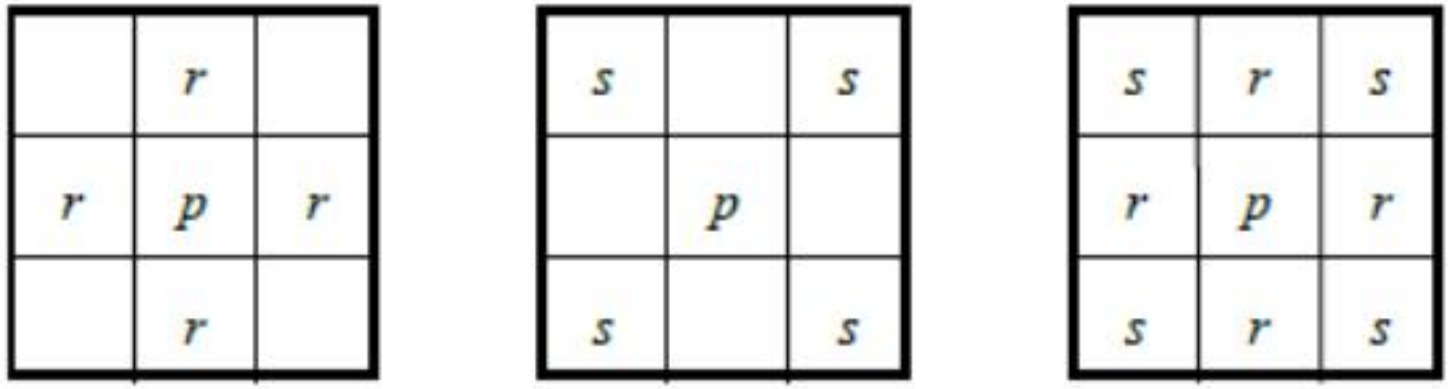


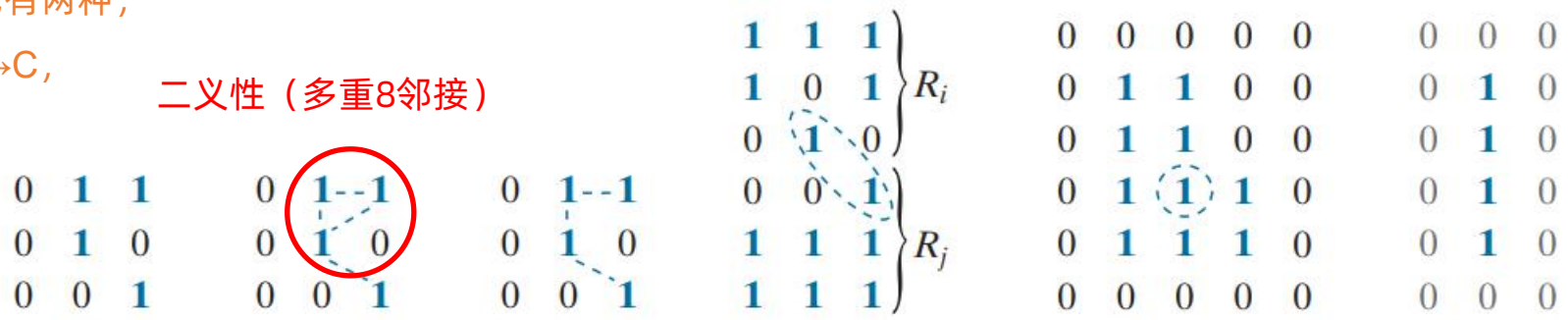
图 2.4.1　像素的邻域

邻接关系

- **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；

A到C的方式有两种，
① 是A→B→C，
② 是A→C

二义性（多重8邻接）



a b c d e f

FIGURE 2.28 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines). (c) *m*-adjacency. (d) Two regions (of 1's) that are 8-adjacent. (e) The circled point is on the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

邻接关系

- **邻接性、连通性、区域和边界**
  - m邻接（混合邻接）：如果
    - q 在$N_4(p)$中，或
    - 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

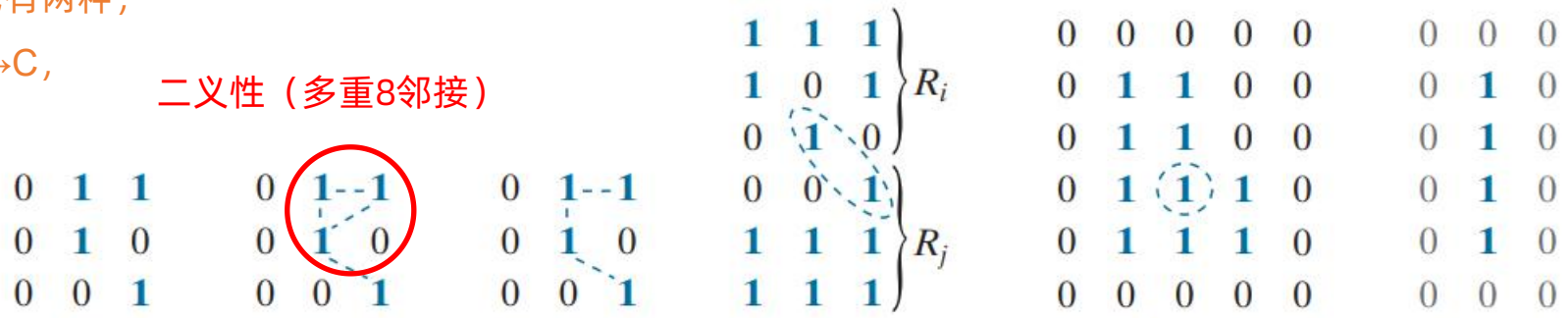
则具有V中数值的两个像素p和q是m邻接的。

A到C的方式有两种，

① 是A→B→C，

② 是A→C

二义性（多重8邻接）

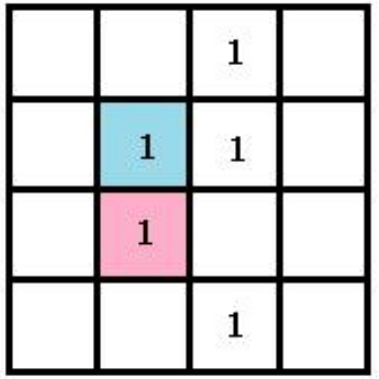


a b c d e f

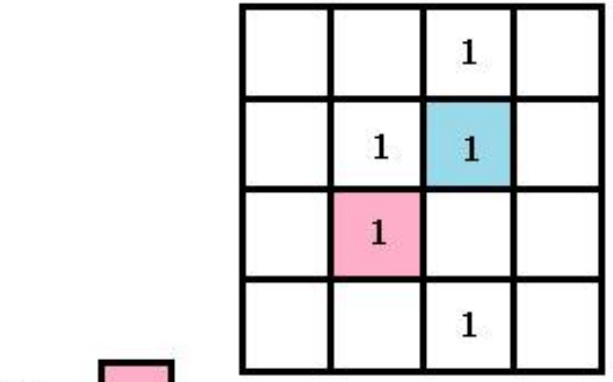
FIGURE 2.28 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines). (c) *m*-adjacency. (d) Two regions (of 1's) that are 8-adjacent. (e) The circled point is on the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

邻接关系

- 邻接性、连通性、区域和边界
  - 邻接像素
  - 4邻接
  - 8邻接
  - *m*邻接（混合邻接）



4邻接

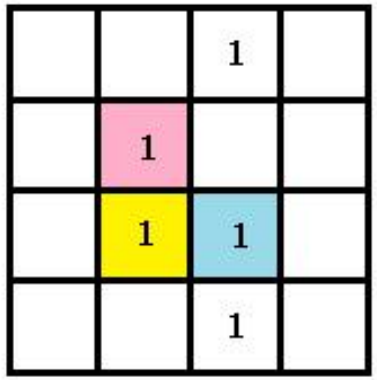


p:

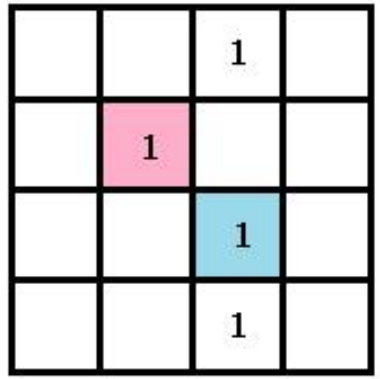
q:

8邻接

不是m邻接

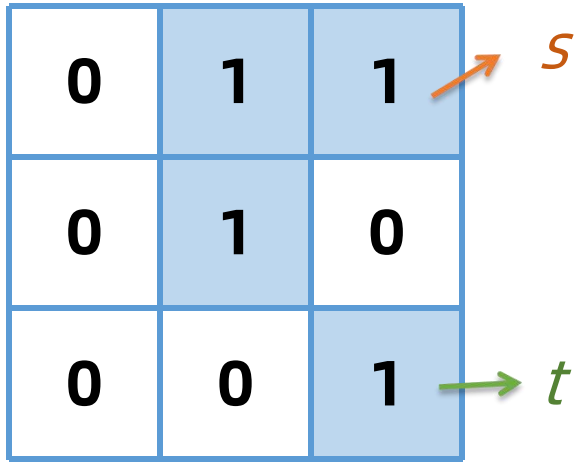


m邻接



邻接关系

- **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



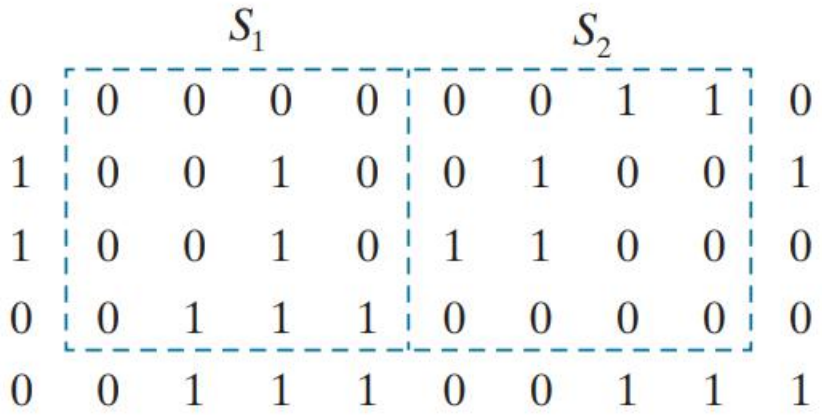
练习题

2.14 Consider the two image subsets, S_1 and S_2 in the following figure. With reference to Section 2.5, and assuming that $V = \{1\}$, determine whether these two subsets are:

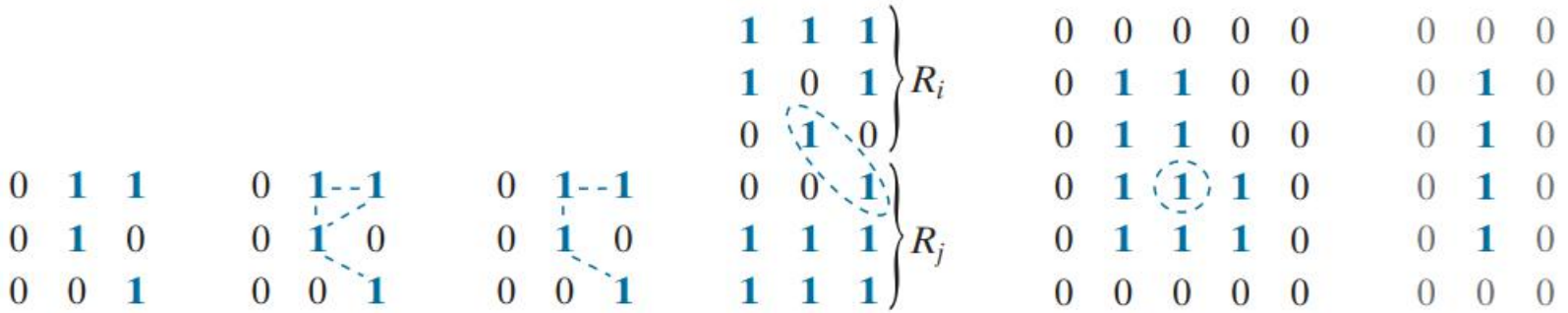
(a)* 4-adjacent.

(b) 8-adjacent.

(c) *m*-adjacent.



- **邻接性、连通性、区域和边界**
  - 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。



a b c d e f

FIGURE 2.28 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines). (c) *m*-adjacency. (d) Two regions (of 1's) that are 8-adjacent. (e) The circled point is on the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

- **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。

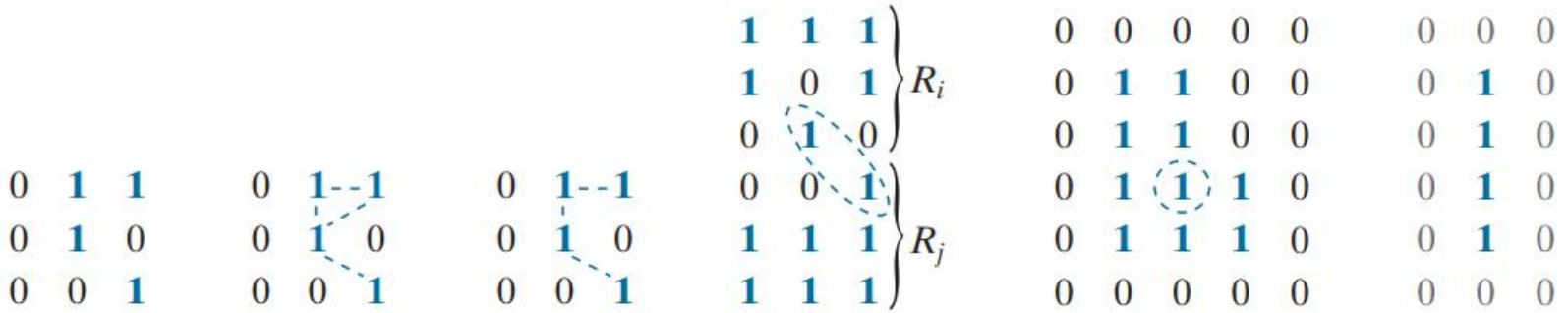
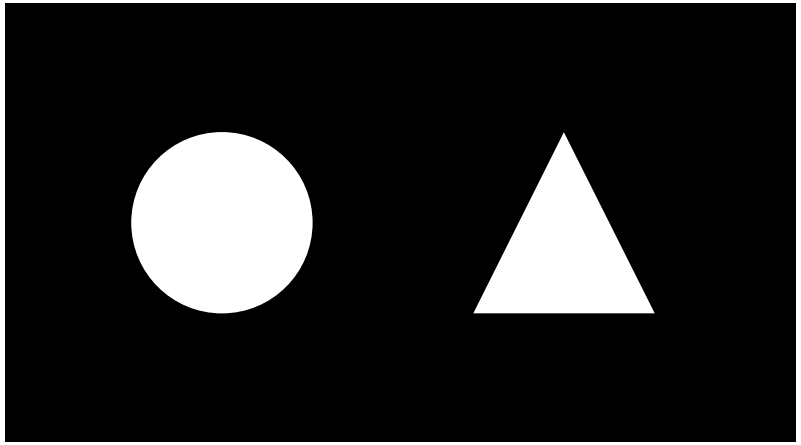


FIGURE 2.28 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines). (c) *m*-adjacency. (d) Two regions (of 1's) that are 8-adjacent. (e) The circled point is on the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

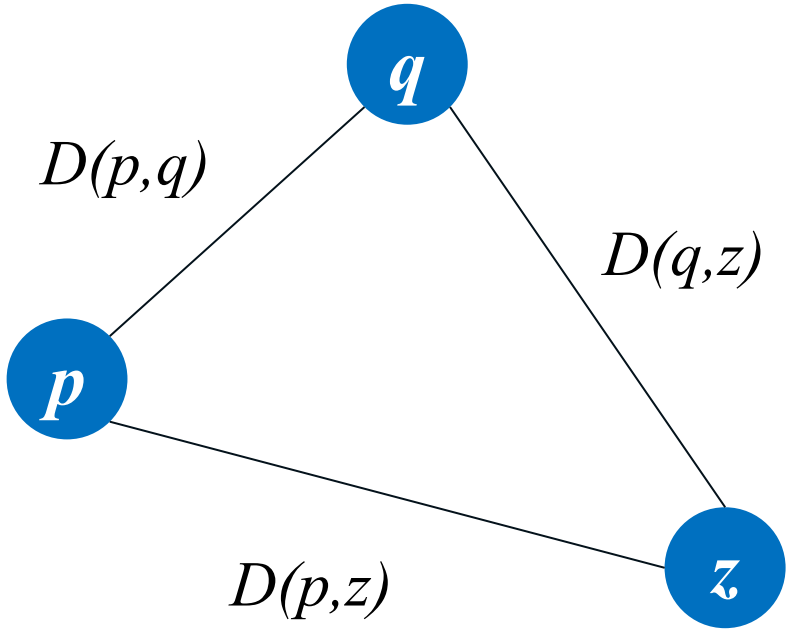
邻接关系

- **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 $p=q]$
  - $D(p,q) = D(q,p)$
  - $D(p,z) \leqq D(p,q) + D(q,z)$



- **距离度量**
  - 欧氏距离（也是范数为2的距离）

$$D_e(p,q) = \left[(x-u)^2 + (y-v)^2\right]^{\frac{1}{2}}$$

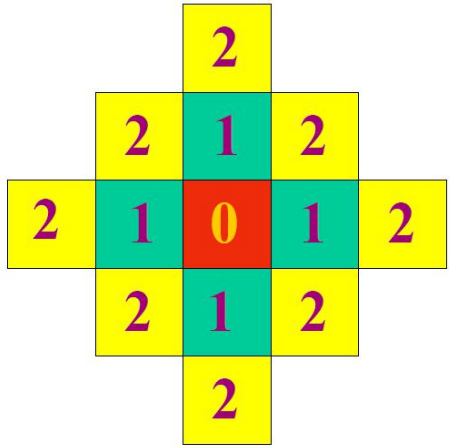
  - 城区距离（也是范数为1的距离）

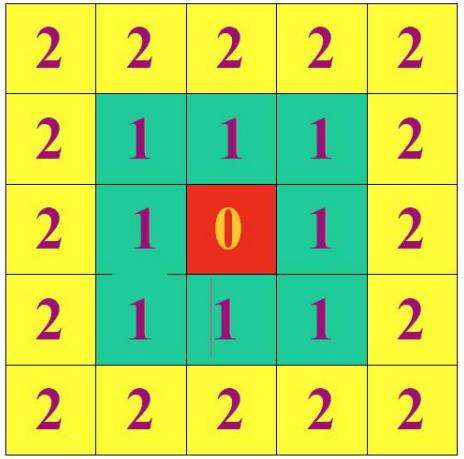
$$D_4(p,q) = |x-u| + |y-v|$$

  - 棋盘距离（也是范数为 ∞ 的距离）

$$D_8(p,q) = \max(|x-u|, |y-v|)$$

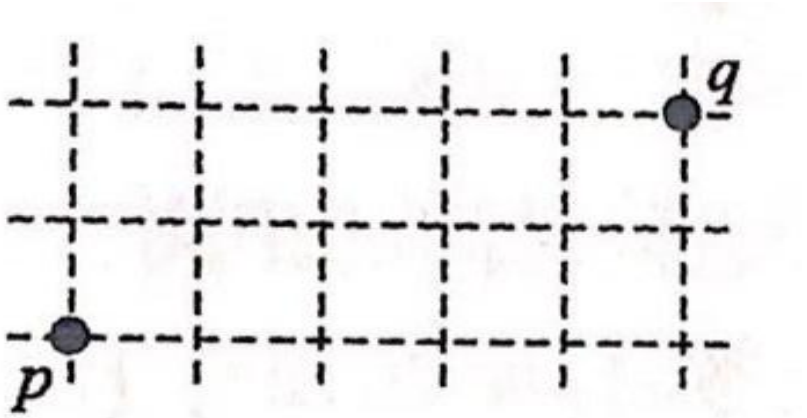
距离与邻域的关系？

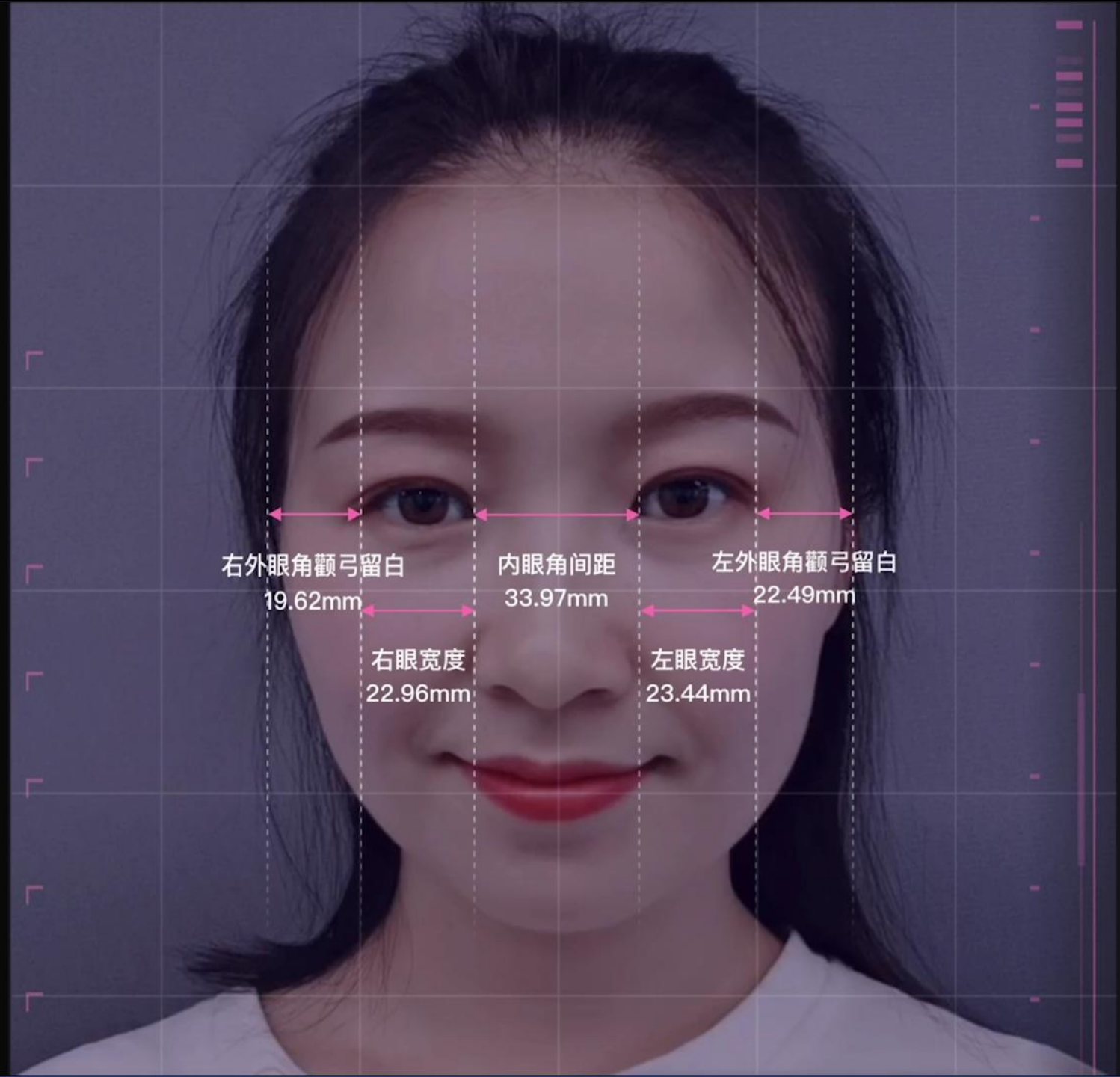




练习题

- **计算如图所示的两个像素 p 和 q 之间的**
  - DE 距离
  - D4 距离
  - D8 距离





五眼右侧偏宽

五眼左侧偏宽

内眼角间距偏宽

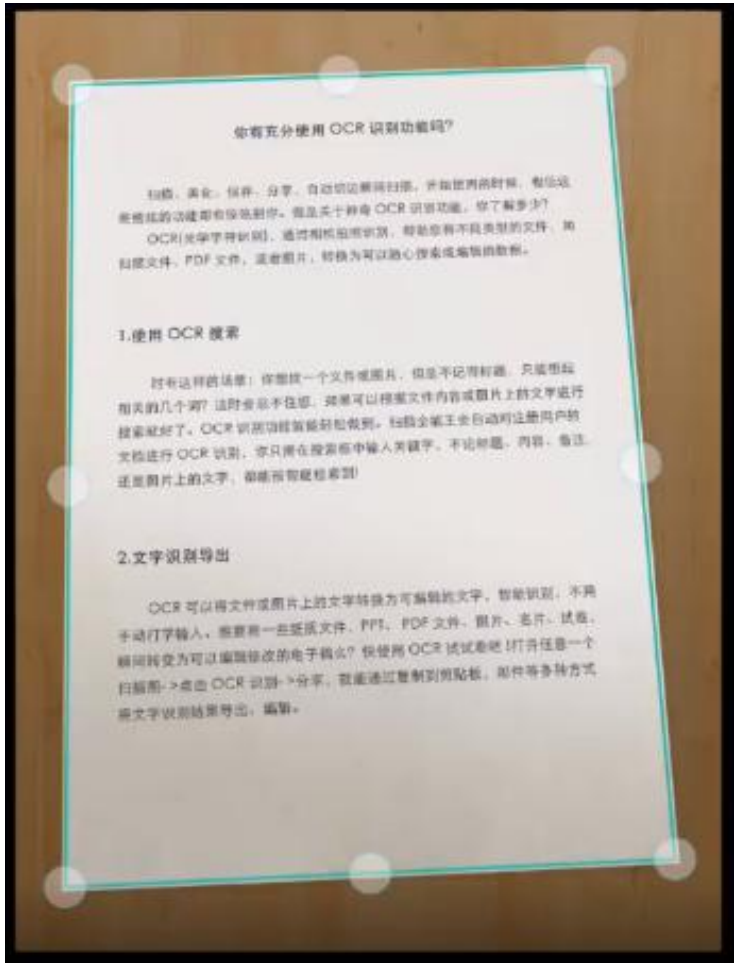
五眼比例 0.84 : 0.98 : 1.45 : 1 : 0.96

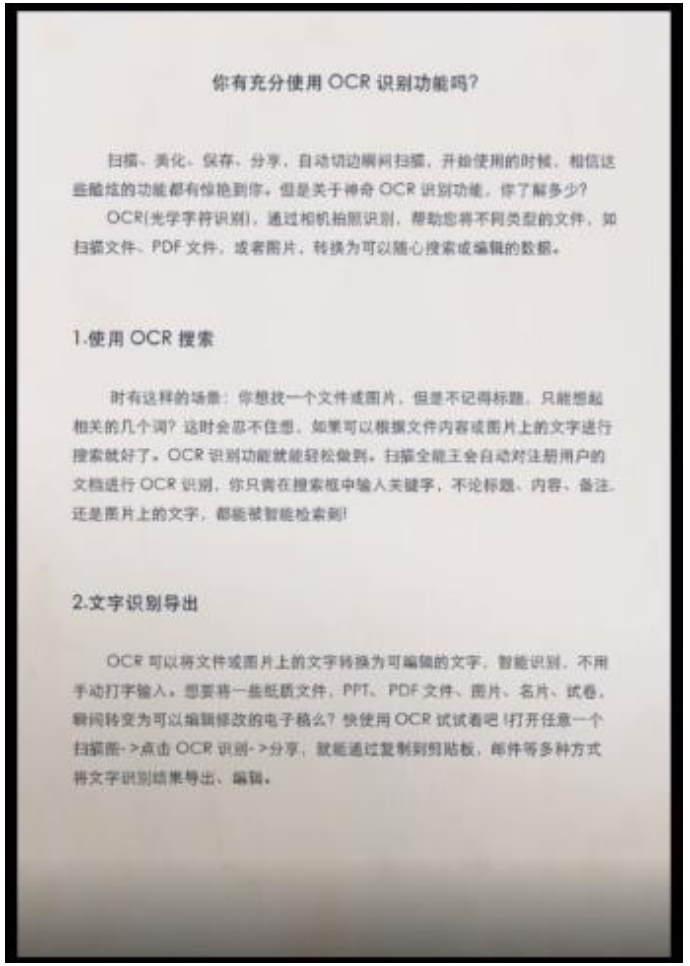
最佳比例 0.8 : 1 : 1.2 : 1 : 0.8

数字图像处理中所用数学工具

思考

- 如何实现以下功能？


你有充分使用 OCR 识别功能吗?
1.使用 OCR 搜索
2.文字识别导出


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。


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。

- **阵列与矩阵操作**
  - 考虑下面的2x2图像

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

  - elementwise product

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

  - matrix product

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

数值运算

- **数值运算**
  - 加减乘除等
  - 图像模糊、去噪等

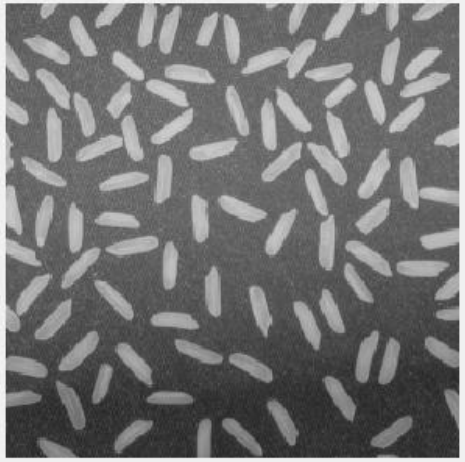
$$s(x,y) = f(x,y) + g(x,y)$$
$$d(x,y) = f(x,y) - g(x,y)$$
$$p(x,y) = f(x,y) \times g(x,y)$$
$$v(x,y) = f(x,y) \div g(x,y)$$

A

B

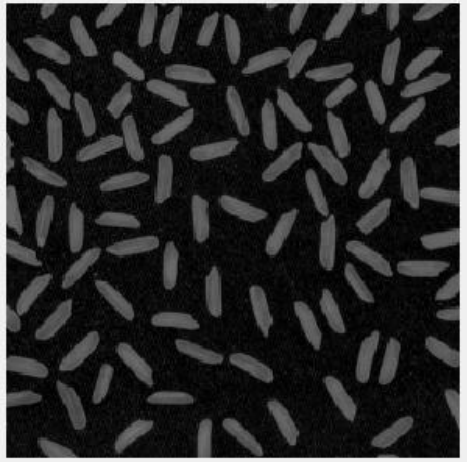
A+B

原图



背景

原图-背景



- **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。

$$s = T(z)$$

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

- **数值运算**
  - 利用图像乘法运算实现图像亮度的控制

$$s = T(z)$$

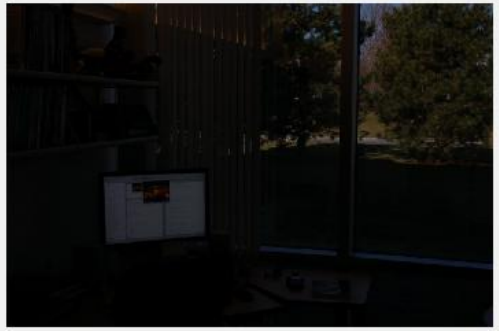
变亮

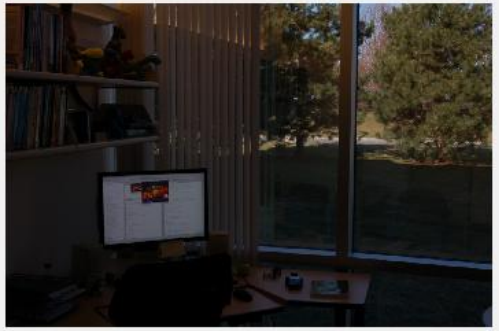
原图

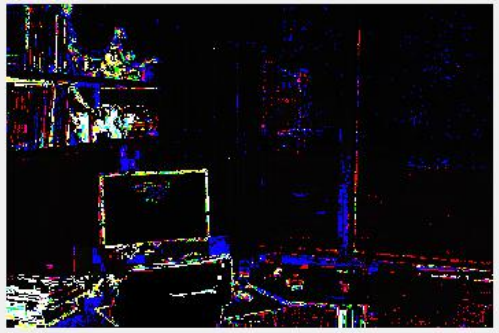
变暗

- **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);
```



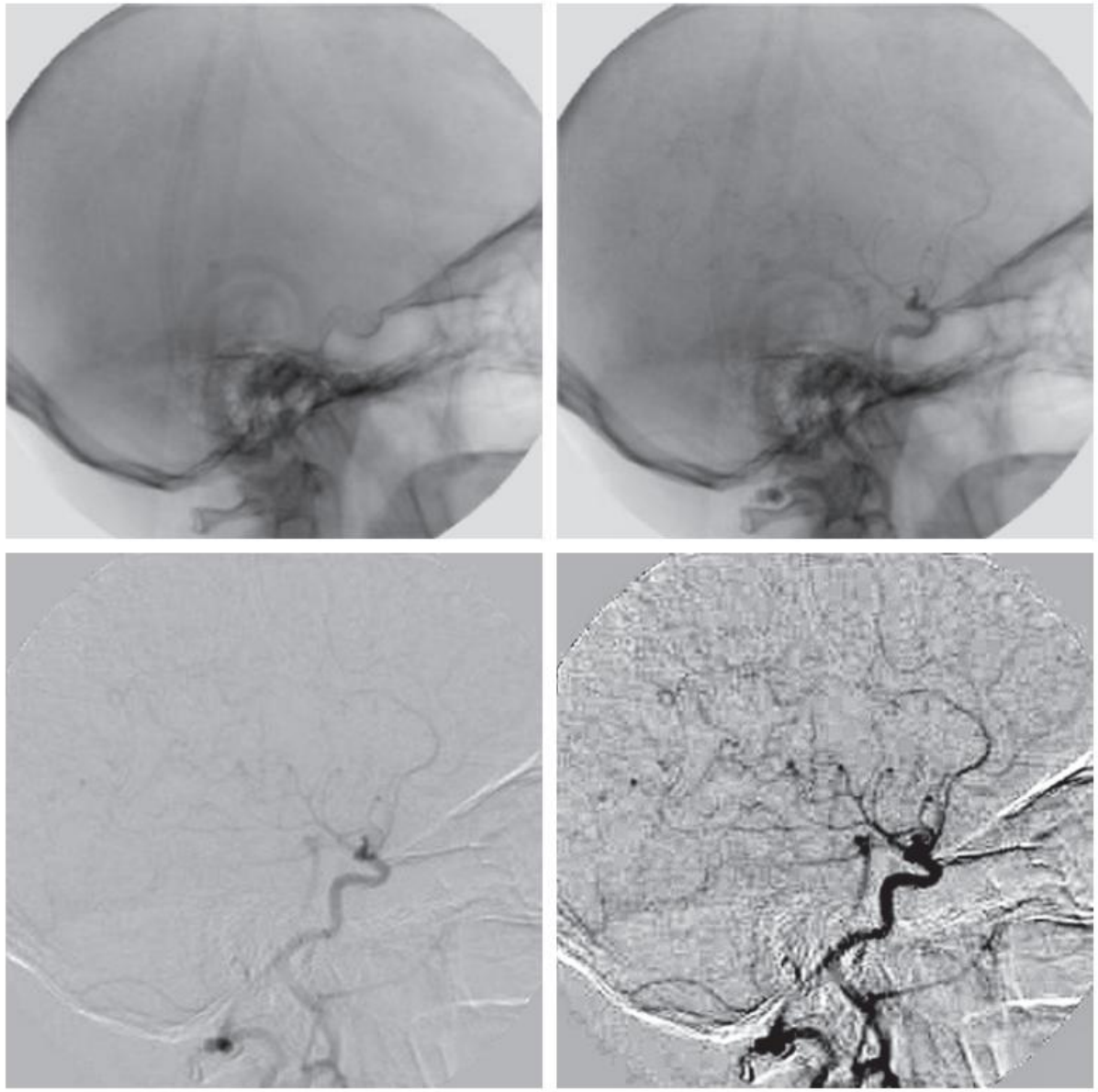




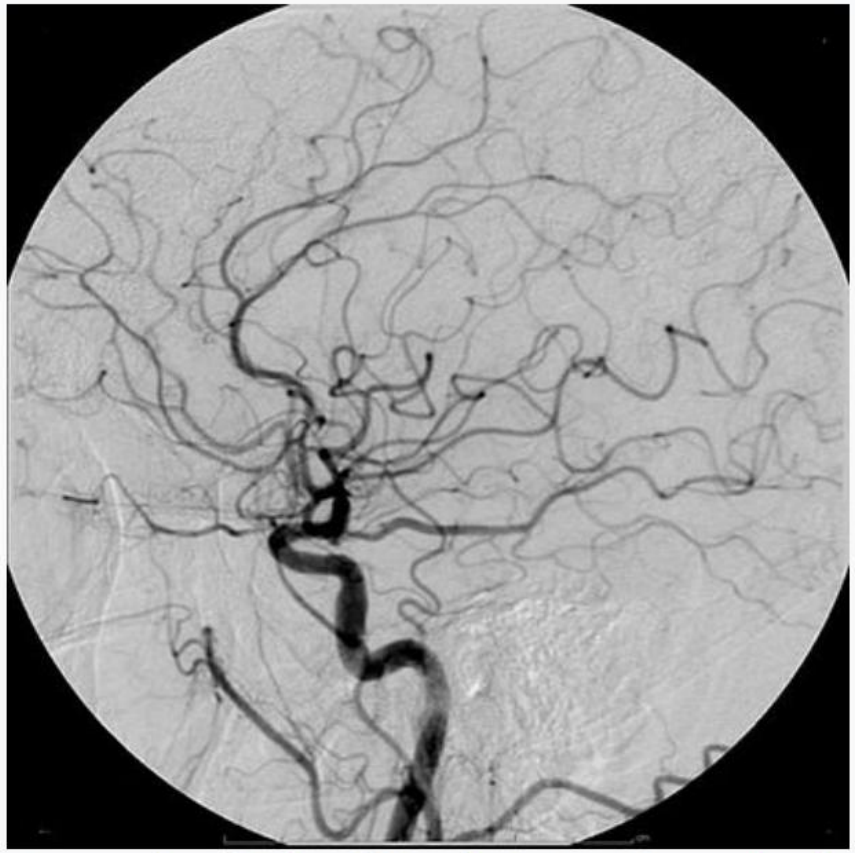
数值运算

a b
c d

FIGURE 2.32
Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of the Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)



数字减影血管造影



Example of iodine-based contrast in cerebral angiography

MeSH D015901

a b c

FIGURE 2.33 Shading correction. (a) Shaded test pattern. (b) Estimated shading pattern. (c) Product of (a) by the reciprocal of (b). (See Section 3.5 for a discussion of how (b) was estimated.)

思 考

- 如何基于(a)和(b)得到(c)?

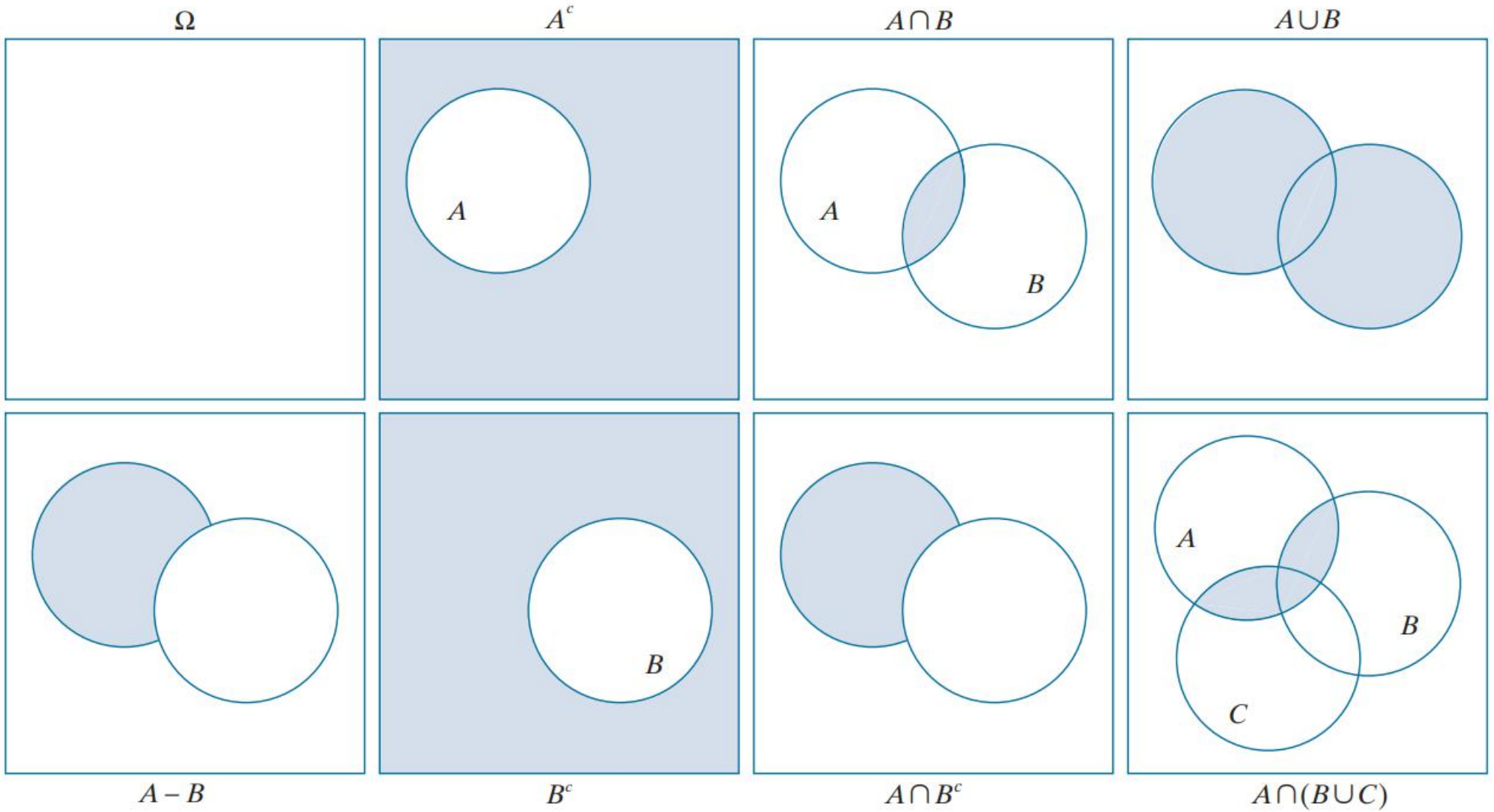
a b c

FIGURE 2.34 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

逻辑运算

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

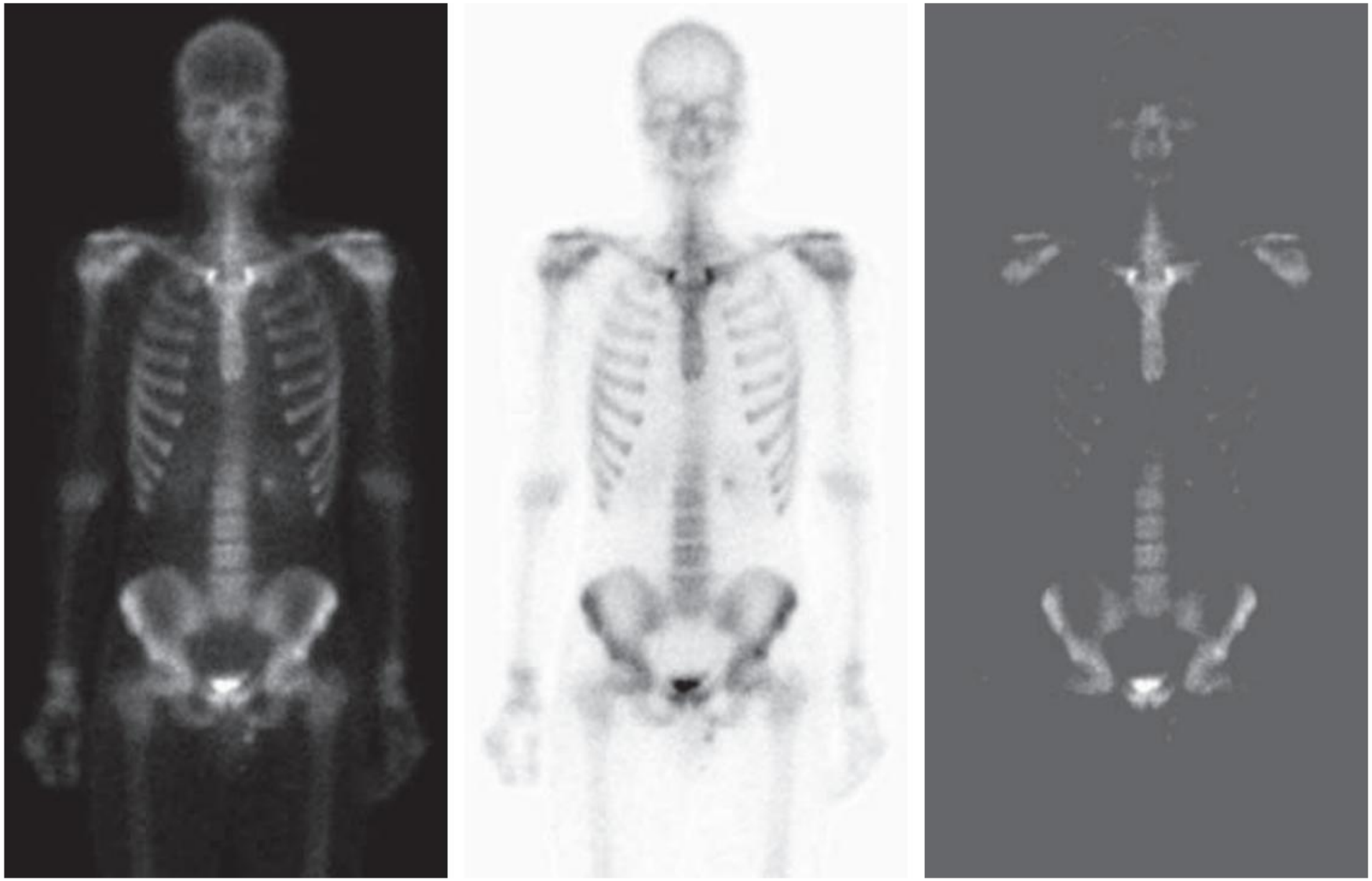
a	b	a AND b	a OR b	NOT(a)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0



逻辑运算

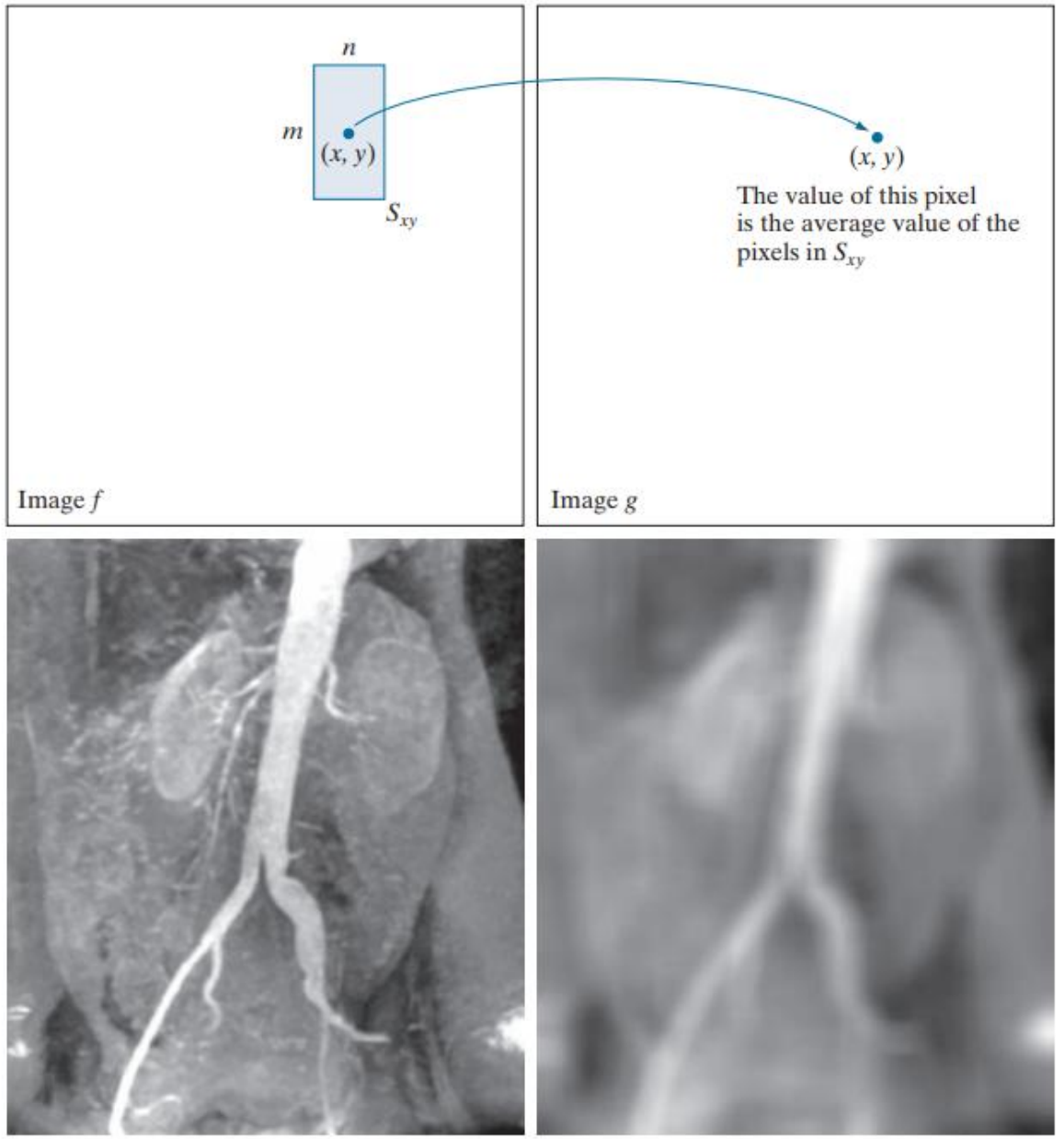
a b c

FIGURE 2.36
Set operations involving grayscale images. (a) Original image. (b) Image negative obtained using grayscale set complementation. (c) The union of image (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)



局部像素操作

$$g(x,y) = \frac{1}{mn} \sum_{(r,c) \in S_{xy}} f(r,c)$$



a b
c d

FIGURE 2.39 Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) An aortic angiogram (see Section 1.3). (d) The result of using Eq. (2-43) with $m = n = 41$. The images are of size 790×686 pixels. (Original image courtesy of Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)

图像坐标变换

- **基本坐标变换**

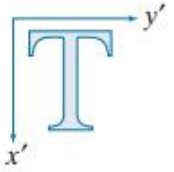
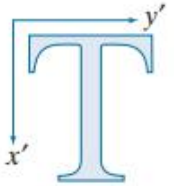
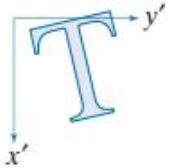
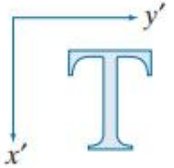
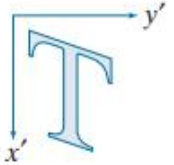
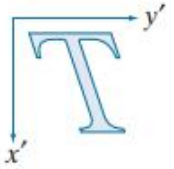
  - 坐标变换可借助矩阵写为：

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

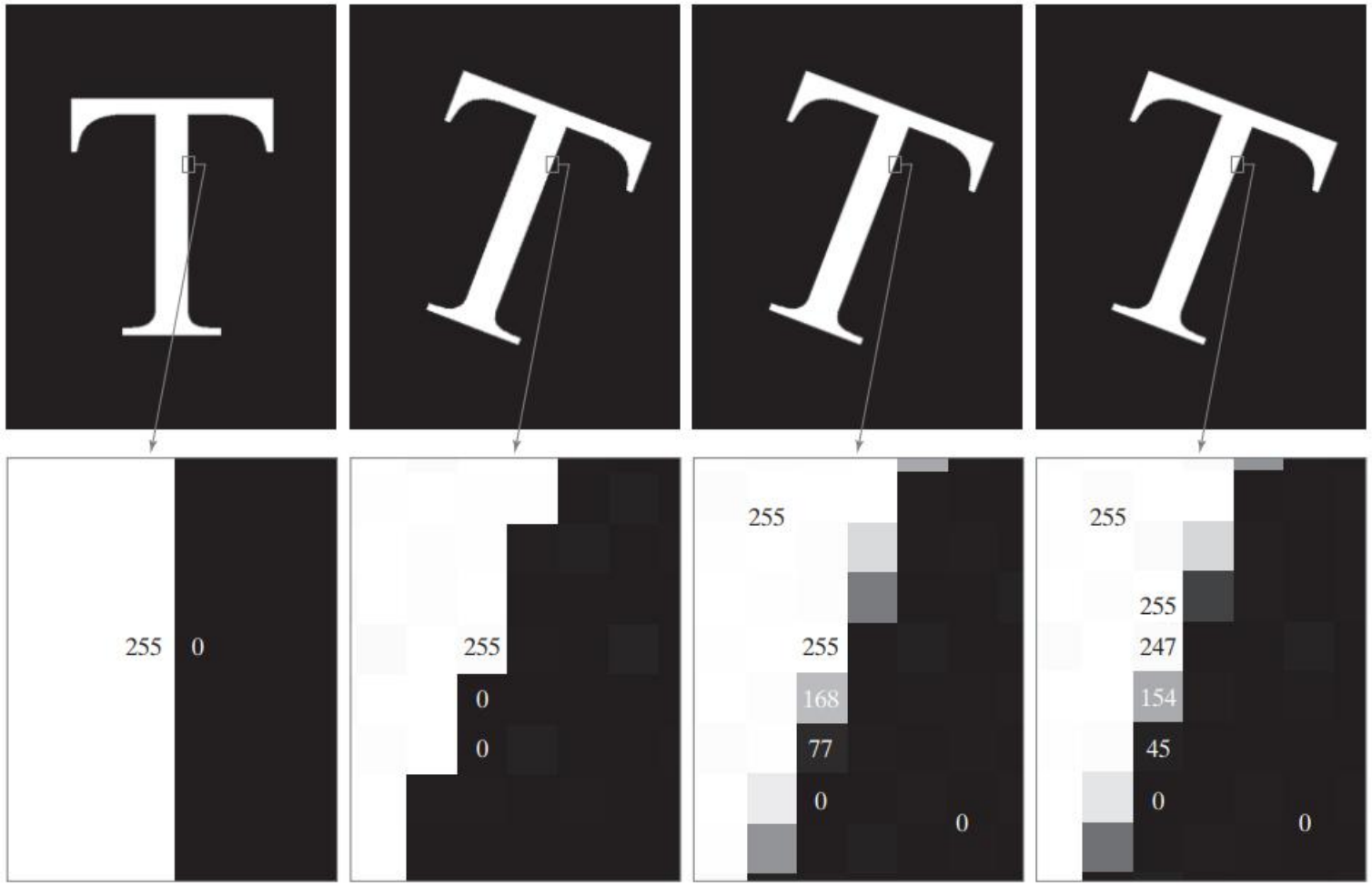
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

齐次坐标系

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

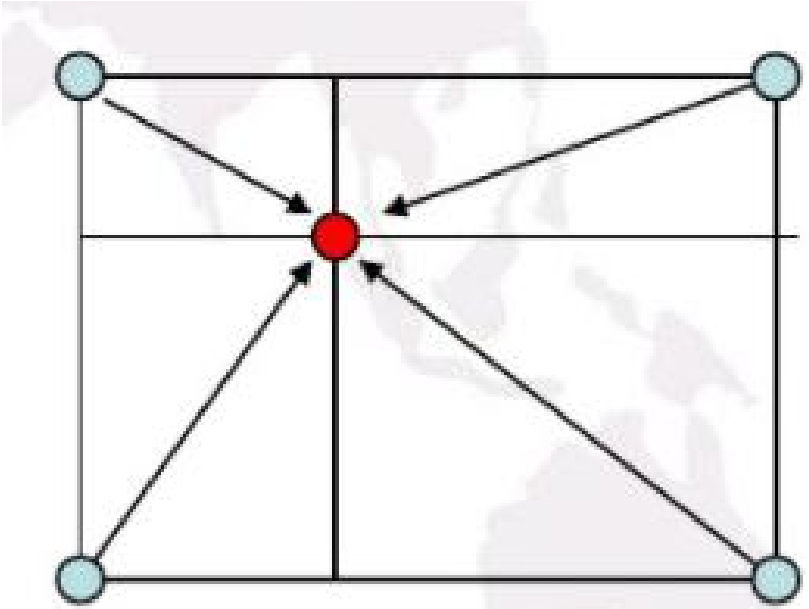
Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling/Reflection (For reflection, set one scaling factor to –1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

图像坐标变换



a b c d
e f g h

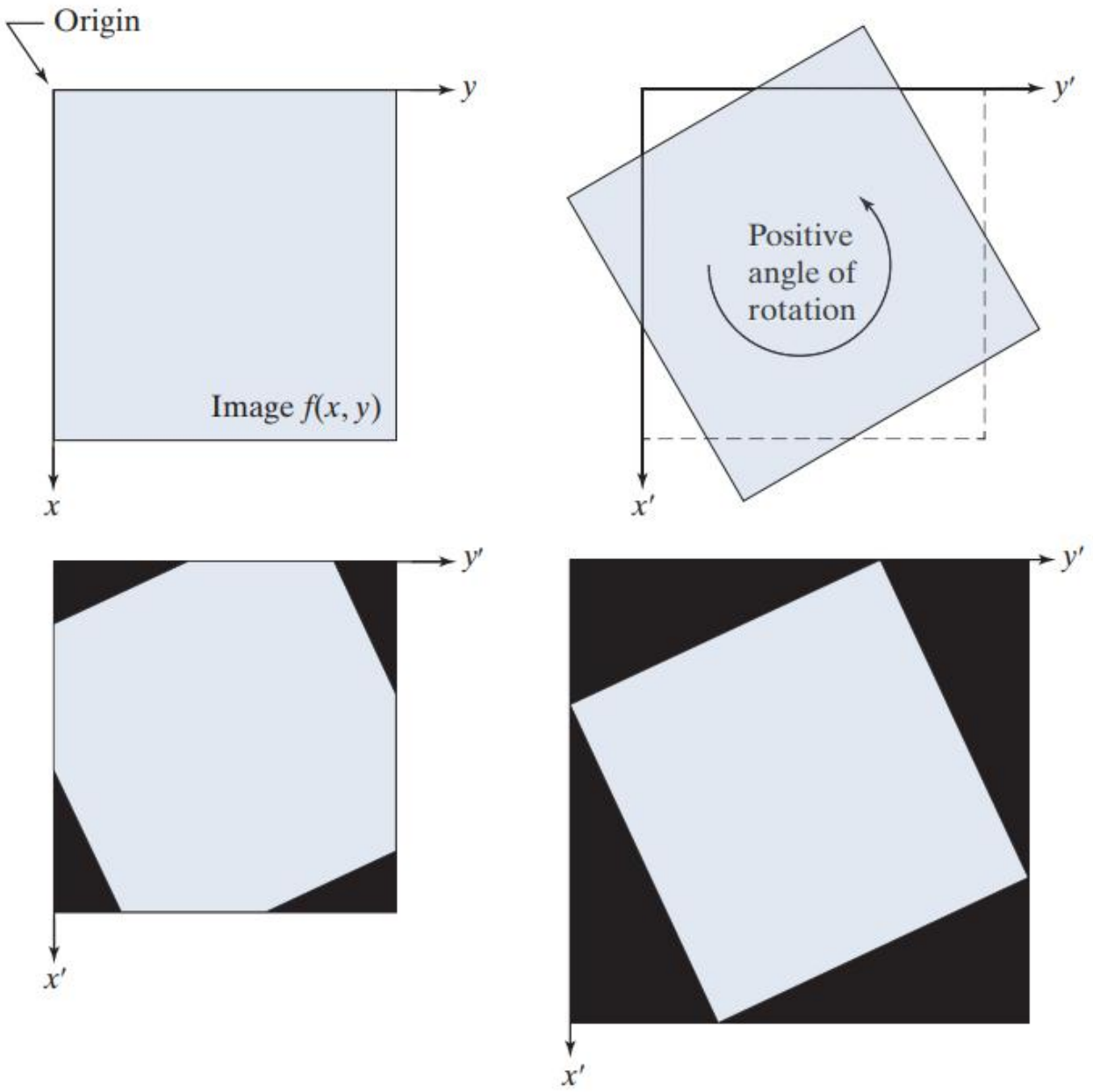
FIGURE 2.40 (a) A 541 × 421 image of the letter T. (b) Image rotated –21° using nearest-neighbor interpolation for intensity assignments. (c) Image rotated –21° using bilinear interpolation. (d) Image rotated –21° using bicubic interpolation. (e)-(h) Zoomed sections (each square is one pixel, and the numbers shown are intensity values).



图像坐标变换

a b
c d

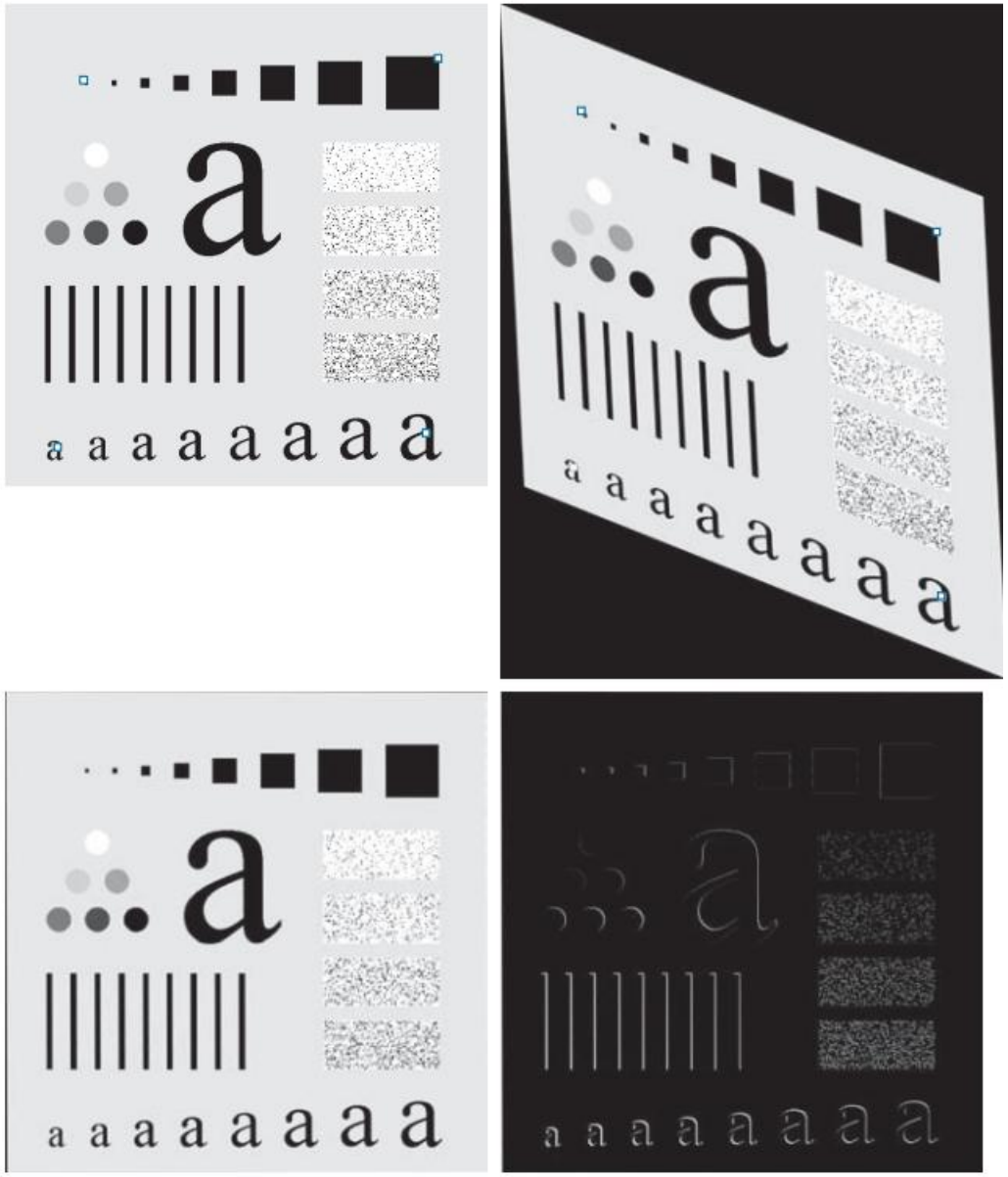
FIGURE 2.41
(a) A digital image.
(b) Rotated image (note the counterclockwise direction for a positive angle of rotation).
(c) Rotated image cropped to fit the same area as the original image.
(d) Image enlarged to accommodate the entire rotated image.



应用案例：图像配准 Image registration

a b
c d

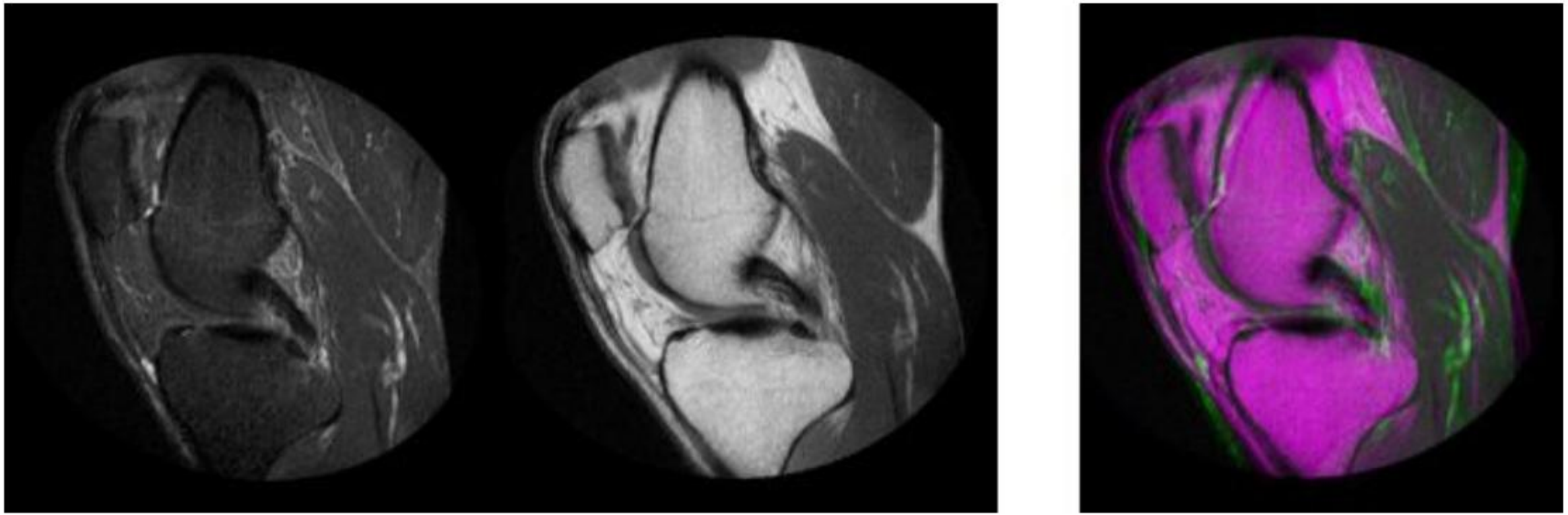
FIGURE 2.42
Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered (output) image (note the errors in the border). (d) Difference between (a) and (c), showing more registration errors.



应用案例：图像配准 Image registration

Registering aerial photos using point mapping.

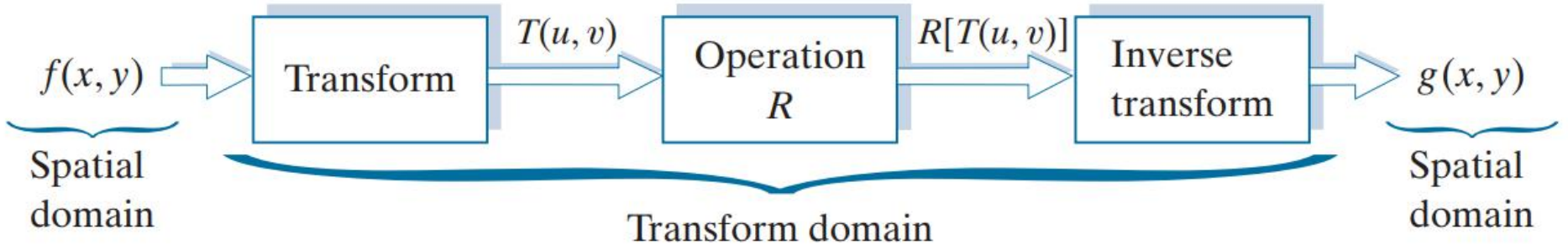
应用案例：图像配准 Image registration



Automatic registration on multimodal medical images.

图像变换 IMAGE TRANSFORMS

FIGURE 2.44
General approach for working in the linear transform domain.



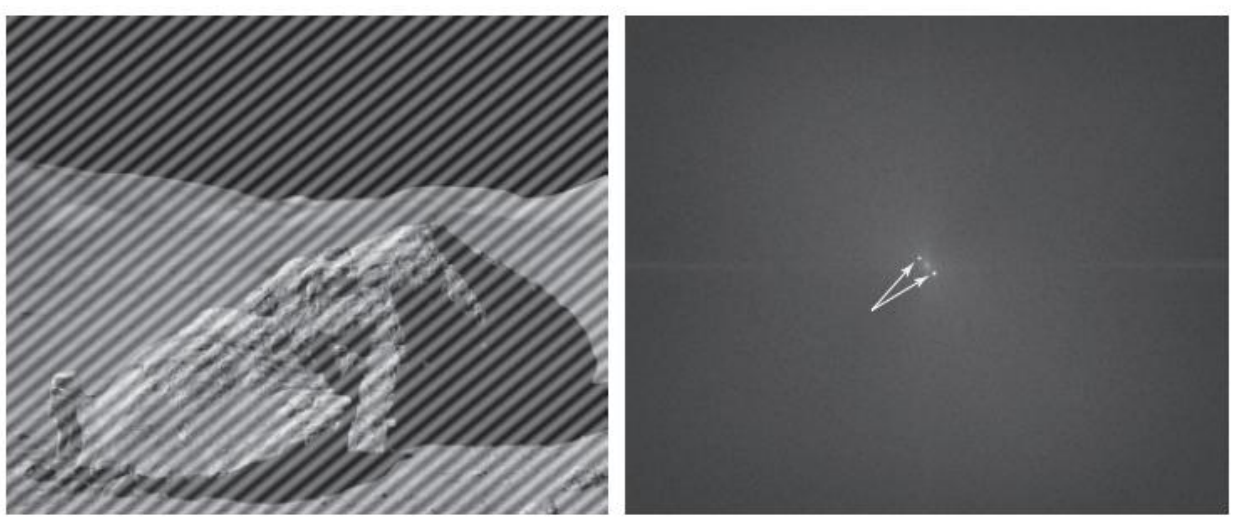
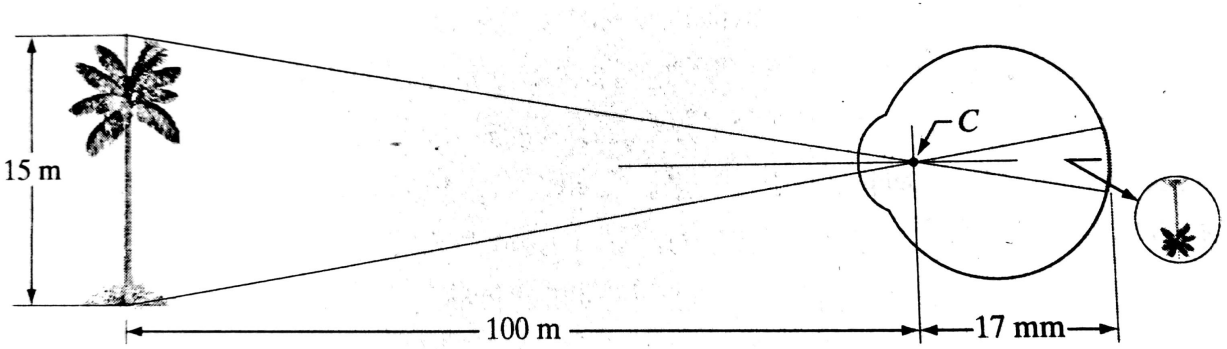
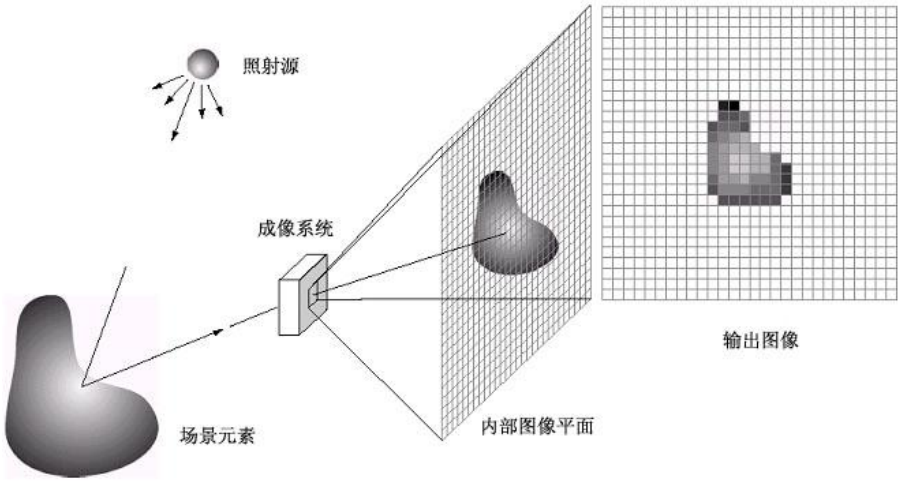


FIGURE 2.45
(a) Image corrupted by sinusoidal interference.
(b) Magnitude of the Fourier transform showing the bursts of energy caused by the interference (the bursts were enlarged for display purposes).
(c) Mask used to eliminate the energy bursts.
(d) Result of computing the inverse of the modified Fourier transform.
(Original image courtesy of NASA.)

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





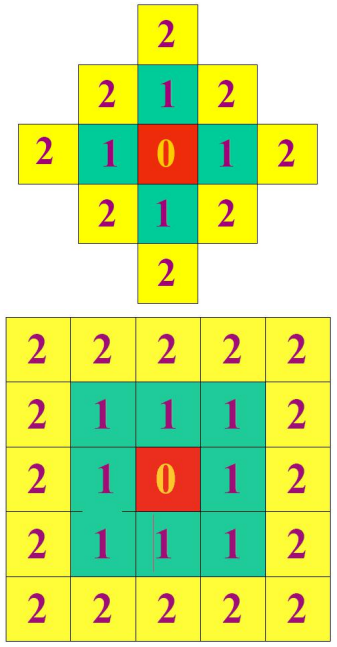


表 2.2　基于式(2.6-23)的仿射变换

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	$\begin{bmatrix}c_x&0&0\\0&c_y&0\\0&0&1\end{bmatrix}$	$x=c_xv$ $y=c_yw$	
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	$\begin{bmatrix}1&s_h&0\\0&1&0\\0&0&1\end{bmatrix}$	$x=v$ $y=s_hv+w$	

数字图像处理