Image Processing

Introduction

Mylène Christine Queiroz de Farias

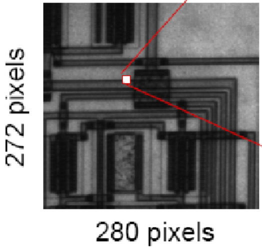
Departamento de Engenharia Elétrica
Universidade de Brasília (UnB)
Brasília, DF 70910-900

mylene@unb.br

9 de Março de 2017

Class 02: Processing Images in the Spatial (Pixel) Domain
Parte 1 - Transforms

Imagem *Digital*



128	125	107	105	110	118	116	114	110
121	122	115	108	106	107	116	116	107
110	114	112	107	105	103	106	106	100
100	96	100	99	94	94	101	101	89
85	82	81	80	76	75	80	82	72
58	58	56	54	53	52	51	49	45
41	41	41	39	39	38	36	35	33
43	43	42	43	41	41	41	43	40
60	60	59	59	60	59	59	58	56

- Pixel = “picture element”
- Represents brightness at one point

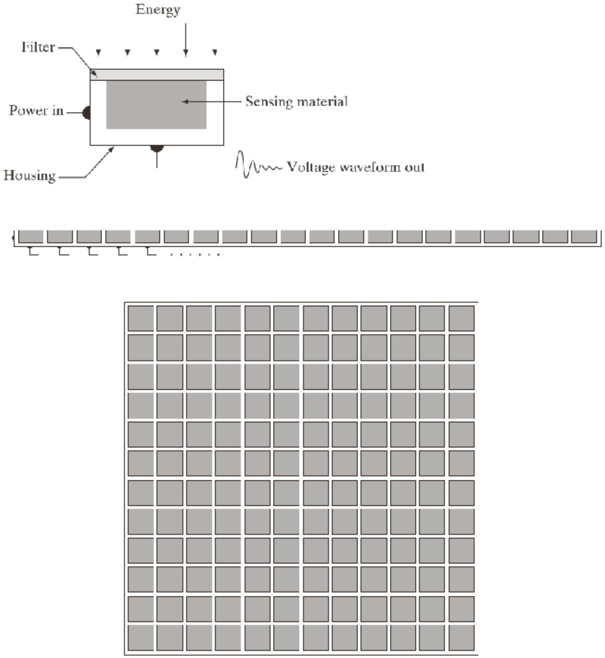


FIGURE 2.12
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

Digital Images

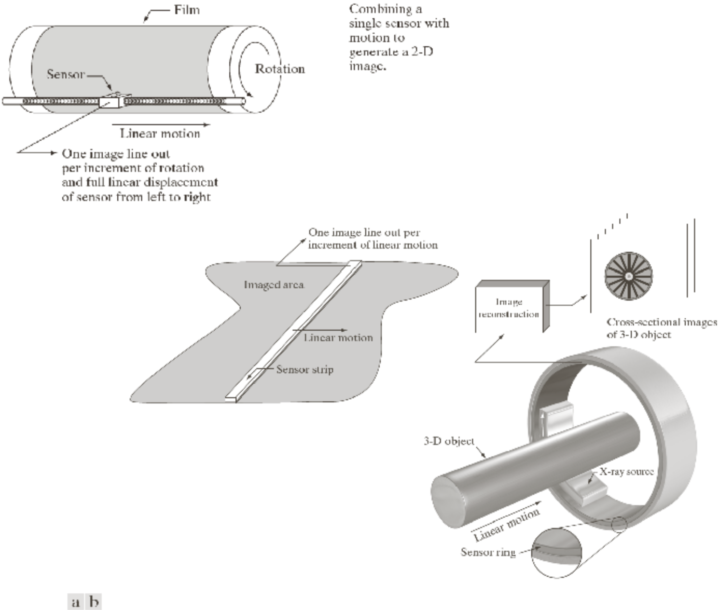


FIGURE 2.13 Combining a single sensor with motion to generate a 2-D image.

Digital Images

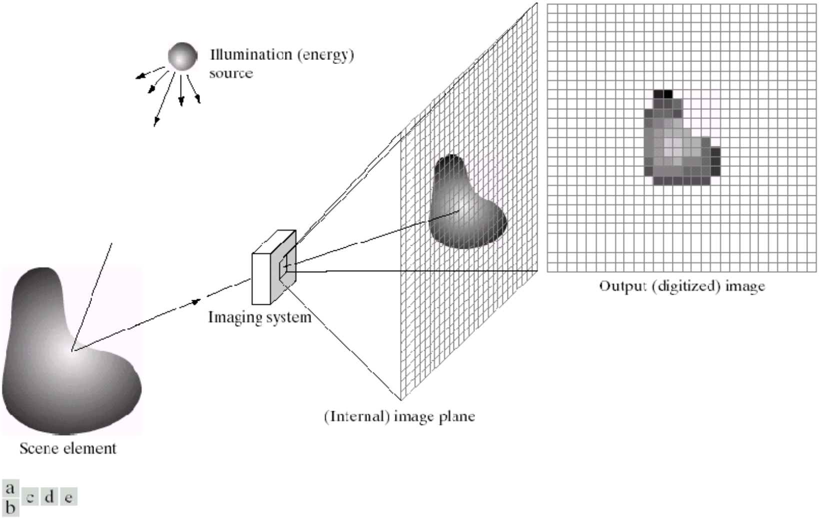
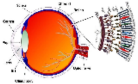


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Mosaicos



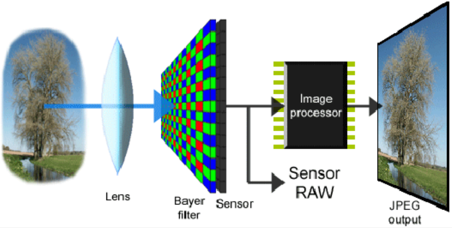
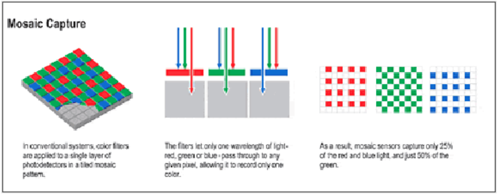
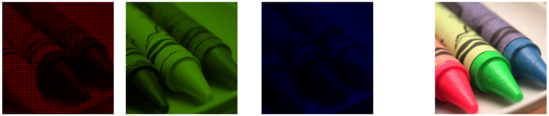
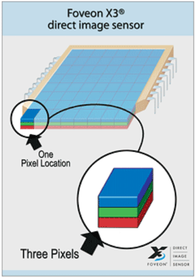
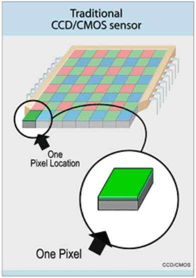


Image
processor
Lens
Bayer
filter
Sensor
Sensor
RAW
JPEG
output

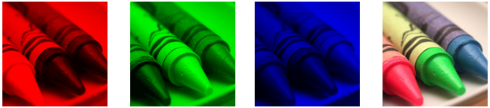

Mosaic Capture
In conventional systems, color filters are applied to a single layer of photodetectors in a tiled mosaic pattern.
The filters let only one wavelength of light - red, green or blue - pass through to any given pixel, allowing it to record only one color.
As a result, mosaic sensors capture only 25% of the red and blue light, and just 50% of the green.

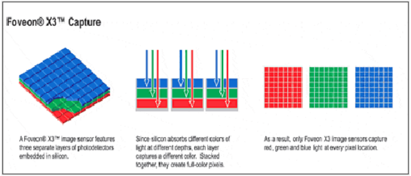
- 

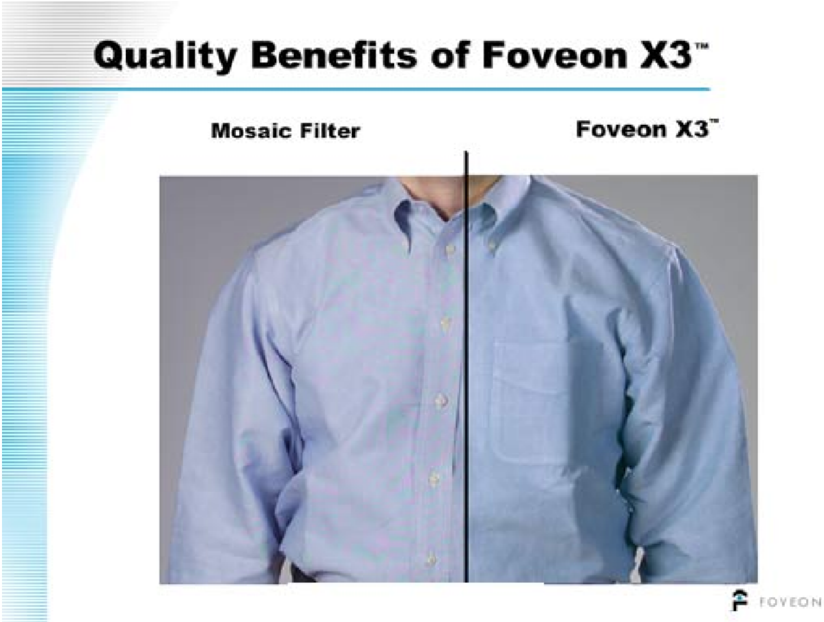

Foveon X3®
direct image sensor
One
Pixel Location
Three Pixels


Traditional
CCD/CMOS sensor
One
Pixel Location
One Pixel
CCD/CMOS

X3




Foveon® X3™ Capture
A Foveon® X3™ image sensor features three separate layers of photodetectors embedded in silicon.
Since silicon absorbs different colors of light at different depths, each layer captures a different color. Stacked together, they create full-color pixels.
As a result, only Foveon X3 image sensors capture red, green and blue light at every pixel location.


Quality Benefits of Foveon X3™
Mosaic Filter
Foveon X3™
FOVEON

Digital Images

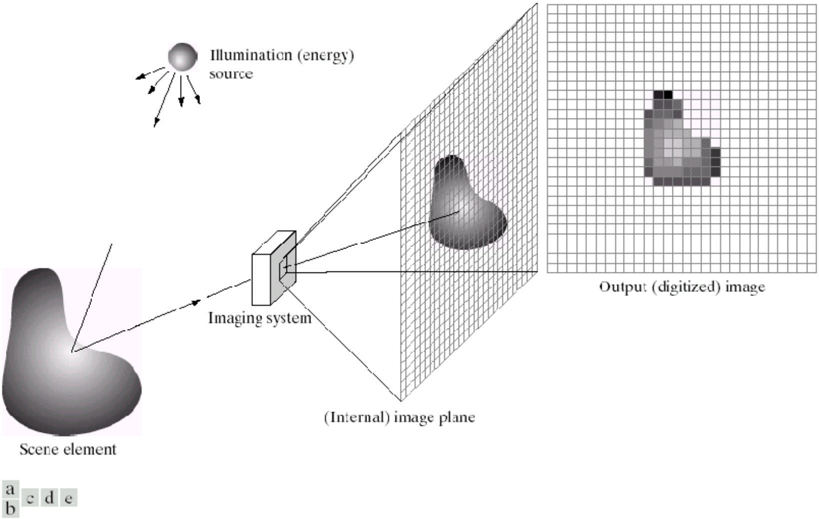


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Digital Images

- Models for Images:
  - $f(x, y)$
  - f - intensity, (x, y) spatial coordinates
  - illumination x reflected component

Light on the object

$$f(x, y) = i(x, y) \cdot r(x, y)$$

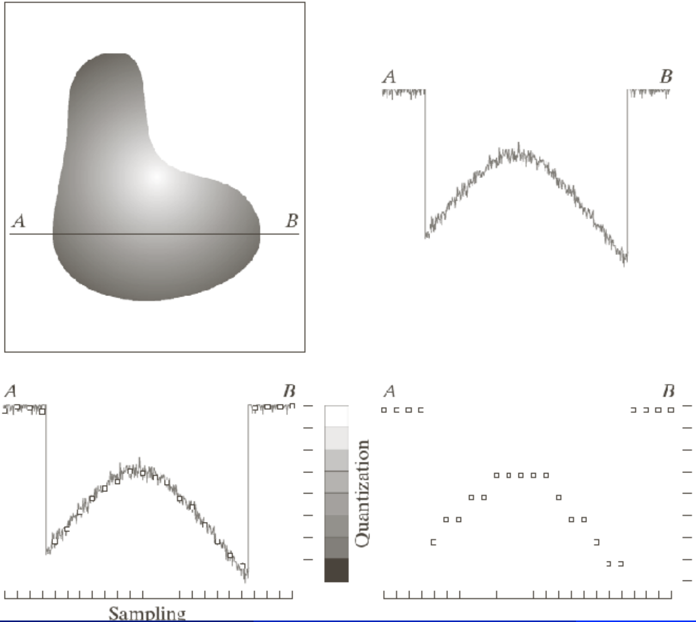
$$0 \leq i(x, y) < \infty, 0 \leq r(x, y) \leq 1)$$

$$L_{min} \leq l \leq L_{max}$$

$$L_{min} = i_{min} \cdot r_{min}$$

$$L_{max} = i_{max} \cdot r_{max}$$

Digital Images



a b
c d

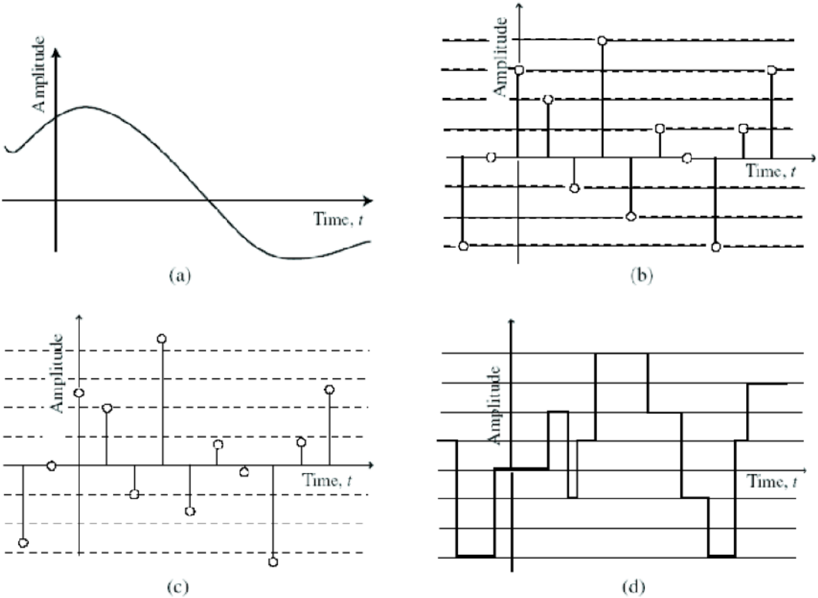
FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

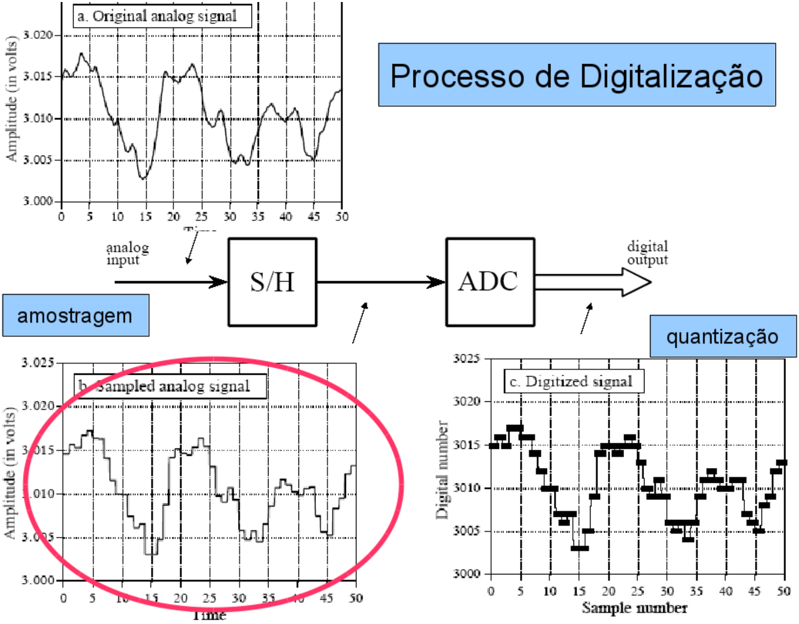
Digital and Continuous Signals

- Continuous or analogue signal
  - Amplitude values can be complex or real numbers;
  - Values are defined for all instants of time.
- Discrete Signal
  - Amplitude values are discrete - a finite number of possible values;
  - Values are defined for discrete instants in time.

$$x = x[n], \ -\infty < n < \infty$$

Digital Images





Processo de Digitalização
a. Original analog signal
Amplitude (in volts)
analog input
S/H
ADC
digital output
amostragem
quantização
b. Sampled analog signal
Amplitude (in volts)
Time
c. Digitized signal
Digital number
Sample number

Sampling and Quantization:

- Sampling of the Spatial Resolution
    - determines the smallest detail in an image;
- Resolution of the number of intensity or color levels (quantization)
    - determines the smallest perceivable change in intensity.

Spatial Resolution

FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

Spatial Resolution

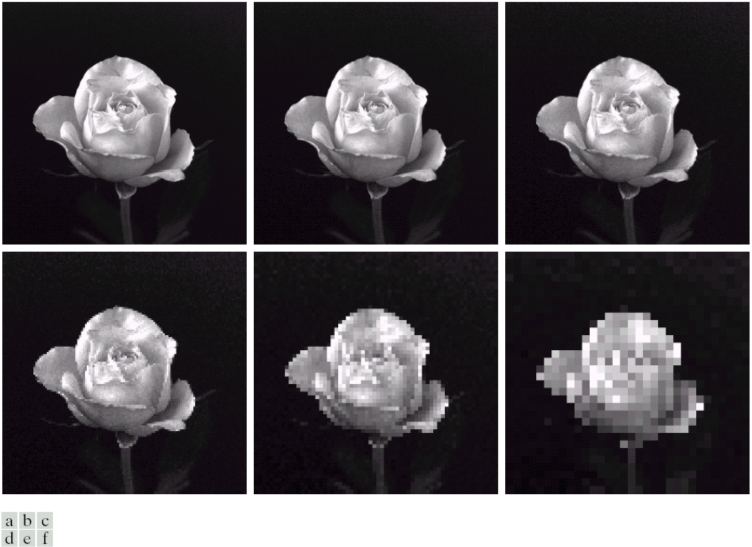
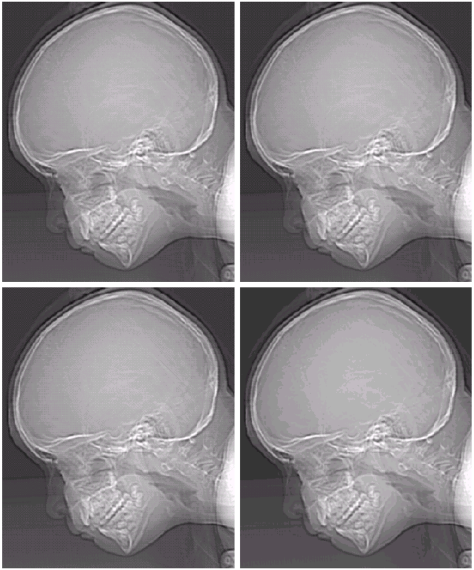


FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

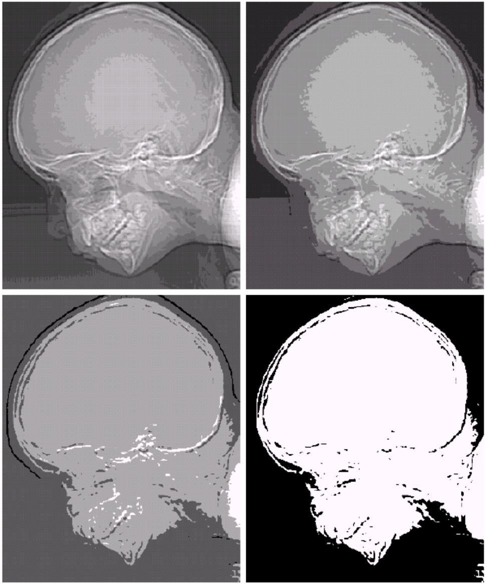


a b
c d

FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Resolução de Intensidade/Cor

$$L = 2^k$$



e f
g h

FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

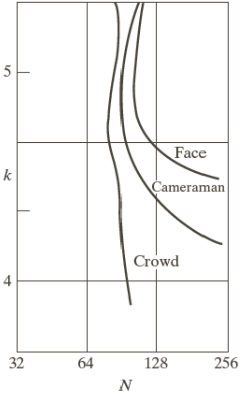
Resolução de Intensidade/Cor

Relação N x k

a b c

FIGURE 2.22 (a) Image with a low level of detail. (b) Image with a medium level of detail. (c) Image with a relatively large amount of detail. (Image (b) courtesy of the Massachusetts Institute of Technology.)

FIGURE 2.23 Typical isopreference curves for the three types of images in Fig. 2.22.



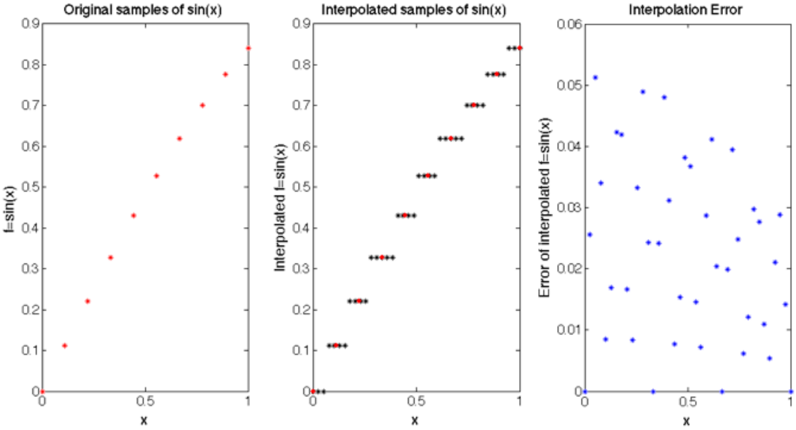
Quanto mais detalhes, menos importa o número de níveis.

- The interpolation or resampling of an image is commonly used em image (and video) processing;
- Interpolation is used to:
  - increase or decrease the size of an image ;
  - rotate an image;
  - create morphing or warping effects;
  - correct lens distortions;
  - register images (create one single image by merging two or more images);
  - stabilize camera trembling;
  - correct patient movements
  - normalize medical images involving several subjects;
  - etc.

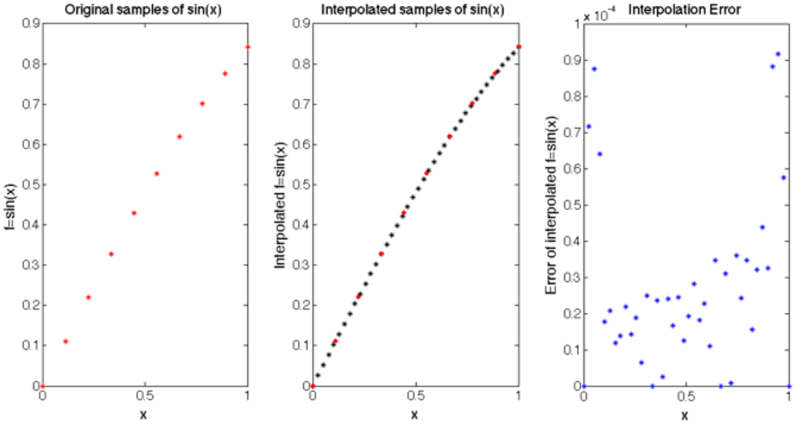
In image processing

- Interpolation is the process in which known values of a signal are used to estimare unknown values.

- Nearest neighbor

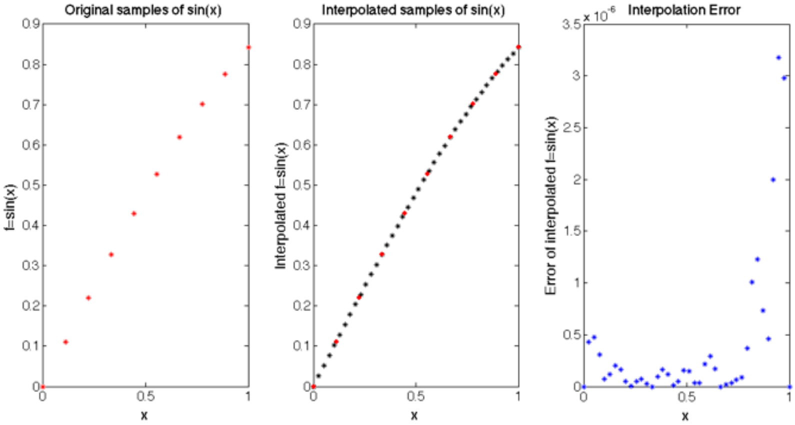


- Nearest neighbor
- Linear: $v(x, y) = a \cdot x + b \cdot y + c \cdot x \cdot y + d$



1-D Interpolation

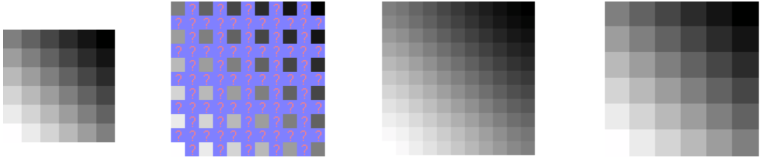
- Nearest neighbor
- Linear: $v(x) = a \cdot x + b$
- Cubic: $v(x) = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$



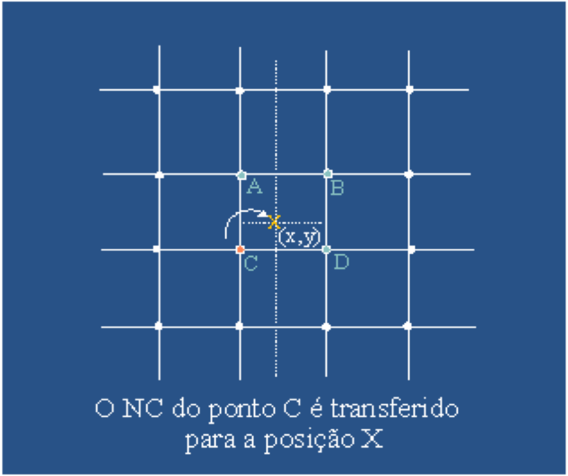
2-D Nearest Neighbor Interpolation

How to increase the size of an image using the Nearest Neighbour interpolation:

- Set to each new position a value equal to the value of the nearest neighbor pixel;
- Replicating – In the special case in which the the increase ratio is a integer number (2, 3, 4, etc.)

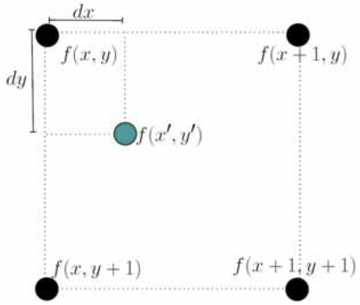


2-D Nearest Neighbor Interpolation



- Blocking effect;
- Fast processing;
- Does not create new values (maintain image statistics)

2-D Nearest Neighbor Interpolation



$$f(x', y') = \begin{cases} f(x, y) & \text{para } dx < 0.5 \text{e } dy < 0.5 \\ f(x+1, y) & \text{para } dx \geq 0.5 \text{e } dy < 0.5 \\ f(x, y+1) & \text{para } dx < 0.5 \text{e } dy \geq 0.5 \\ f(x+1, y+1) & \text{para } dx \geq 0.5 \text{e } dy \geq 0.5 \end{cases} \quad (1)$$

2-D Nearest Neighbor Interpolation

Nearest neighbor interpolation (NNI):

- Example:

...	f(i,j)	f(i,j+1)	...
...	f(i+1,j)	f(i+1,j+1)	...

2-D Nearest Neighbor Interpolation

Nearest neighbor interpolation (NNI):

- Example:

...	f(i,j)	f(i,j+1)	...
...	f(i+1,j)	f(i+1,j+1)	...

- Add lines and columns of zeros

...	f(i,j)	0	f(i,j+1)	...
...	0	0	0	...
...	f(i+1,j)	0	f(i+1,j+1)	...

Nearest neighbor interpolation (NNI):

- Example:

$$\begin{array}{cccc} \dots & f(i,j) & f(i,j+1) & \dots \\ \dots & f(i+1,j) & f(i+1,j+1) & \dots \end{array}$$

- Add lines and columns of zeros

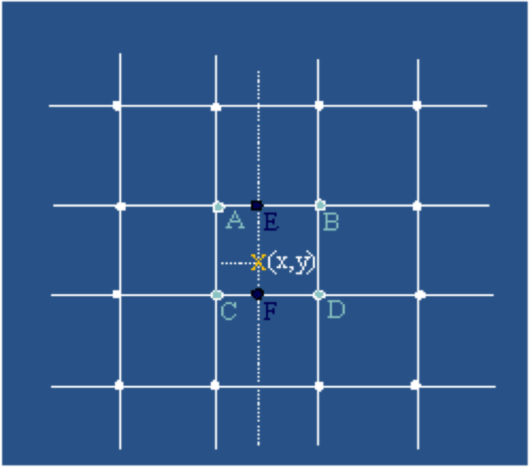
$$\begin{array}{ccccc} \dots & f(i,j) & 0 & f(i,j+1) & \dots \\ \dots & 0 & 0 & 0 & \dots \\ \dots & f(i+1,j) & 0 & f(i+1,j+1) & \dots \end{array}$$

- After the NNI

$$\begin{array}{ccccc} \dots & f(i,j) & f(i,j) & f(i,j+1) & \dots \\ \dots & f(i,j) & f(i,j) & f(i,j+1) & \dots \\ \dots & f(i+1,j) & f(i+1,j) & f(i+1,j+1) & \dots \end{array}$$

(a) Imagem original. (b) Região da imagem ampliada.

2-D Bilinear Interpolation



- The value of the position X is the weighted average of the pixels E and F;
- Blurring effect due to the averaging effect.

2-D Bilinear Interpolation

Bilinear:

$$\begin{matrix} \ldots & \text{f(i,j)} & \text{f(i,j+1)} & \ldots \\ \ldots & \text{f(i+1,j)} & \text{f(i+1,j+1)} & \ldots \end{matrix}$$

2-D Bilinear Interpolation

Bilinear:

...	f(i,j)	f(i,j+1)	...
...	f(i+1,j)	f(i+1,j+1)	...

- Adding columns and lines of zeros:

...	f(i,j)	a	f(i,j+1)	...
...	b	c	d	...
...	f(i+1,j)	e	f(i+1,j+1)	...

2-D Bilinear Interpolation

Bilinear:

$$\begin{matrix} \dots & f(i,j) & f(i,j+1) & \dots \\ \dots & f(i+1,j) & f(i+1,j+1) & \dots \end{matrix}$$

- Adding columns and lines of zeros:

$$\begin{matrix} \dots & f(i,j) & a & f(i,j+1) & \dots \\ \dots & b & c & d & \dots \\ \dots & f(i+1,j) & e & f(i+1,j+1) & \dots \end{matrix}$$

$a = (f(i,j) + f(i,j+1))/\ 2$
$e = (f(i+1,j) + f(i+1,j+1))/\ 2$
$b = (f(i,j) + f(i+1,j))/\ 2$
$d = (f(i,j+1) + f(i+1,j+1))/\ 2$
$c = (f(i,j) + f(i,j+1) + f(i+1,j) + f(i+1,j+1))/\ 4$

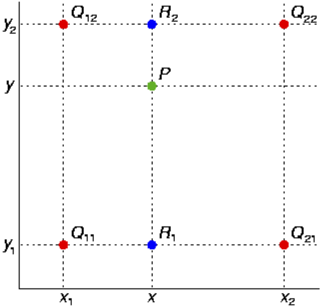
2-D Bilinear Interpolation

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

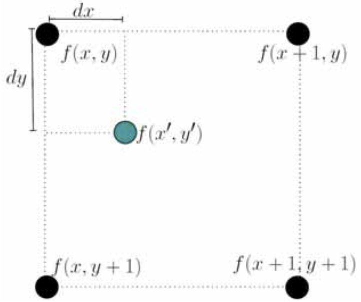
$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2).$$

$$\begin{aligned}
f(x,y) \approx\ & \frac{f(Q_{11})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y) + \\
& \frac{f(Q_{21})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y) + \\
& \frac{f(Q_{12})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1) + \\
& \frac{f(Q_{22})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1) \\
=\ & \frac{1}{(x_2-x_1)(y_2-y_1)}\Big(f(Q_{11})(x_2-x)(y_2-y) + \\
& f(Q_{21})(x-x_1)(y_2-y) + \\
& f(Q_{12})(x_2-x)(y-y_1) + \\
& f(Q_{22})(x-x_1)(y-y_1)\Big)
\end{aligned}$$



2-D Bilinear Interpolation

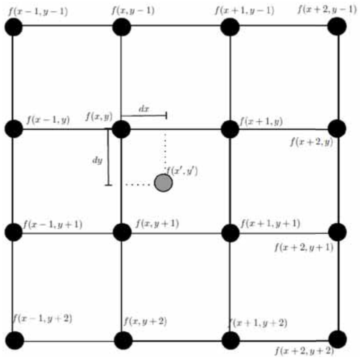
$$f(x', y') = (1 - dx) \cdot (1 - dy) \cdot f(x, y) + dx \cdot (1 - dy) \cdot f(x + 1, y) + (1 - dx) \cdot dy \cdot f(x, y + 1) + dx \cdot dy \cdot f(x, y)$$



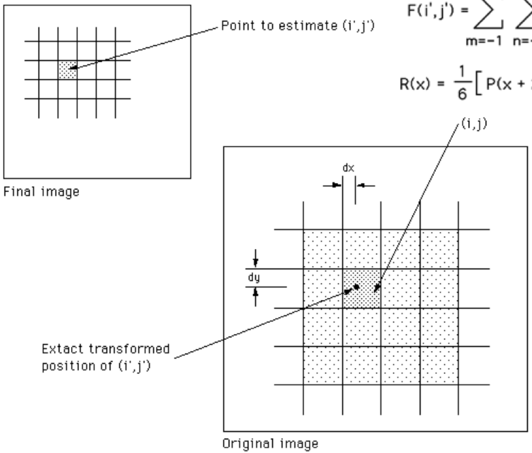
(a) Imagem original (b) Região da imagem ampliada

2-D Bicubic Interpolation

- 16 neighbors: $p(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x_i y_i$
- Generally, the bicubic interpolation does a better job in preserving the details than the bilinear interpolation;
- The bicubic interpolation is the standard used by commercial applications, like Adobe Photoshop and Corel Photopaint.



2-D Bicubic Interpolation



$$F(i',j') = \sum_{m=-1}^{2} \sum_{n=-1}^{2} F(i+m, j+n)\, R(m-dx)\, R(dy-n)$$

$$R(x) = \frac{1}{6}\left[P(x+2)^3 - 4\,P(x+1)^3 + 6\,P(x)^3 - 4\,P(x-1)^3 \right]$$

$$P(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$$

(a) Imagem original

(b) Região da imagem ampliada

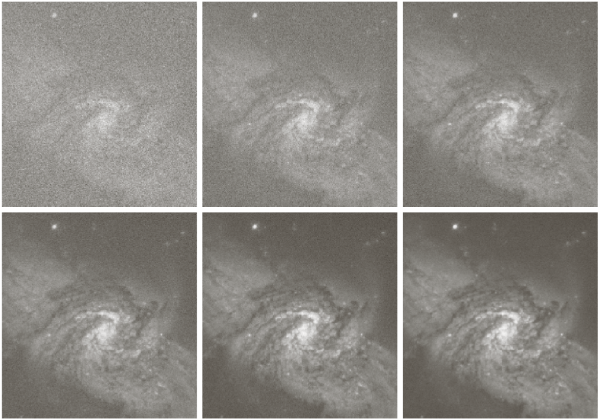
a b c
d e f

FIGURE 2.24 (a) Image reduced to 72 dpi and zoomed back to its original size (3692 × 2812 pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

Operations

- By elements or matrices
- Linear and non-linear
  - homogeneitys and additivity;
- Arithmetic operations:
  - sum, subtraction, division, multiplication
- Logic operations
- etc.

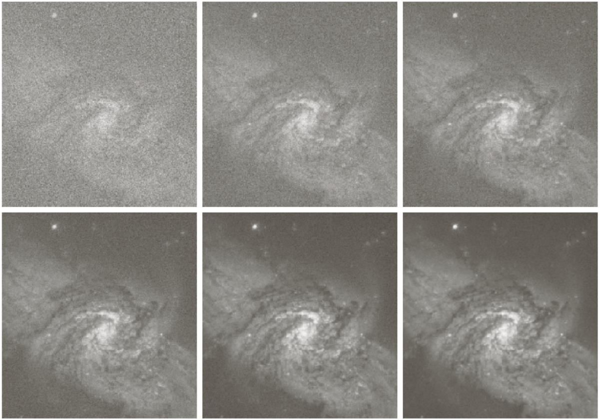
Sum (average)



a b c
d e f

FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Sum (average)

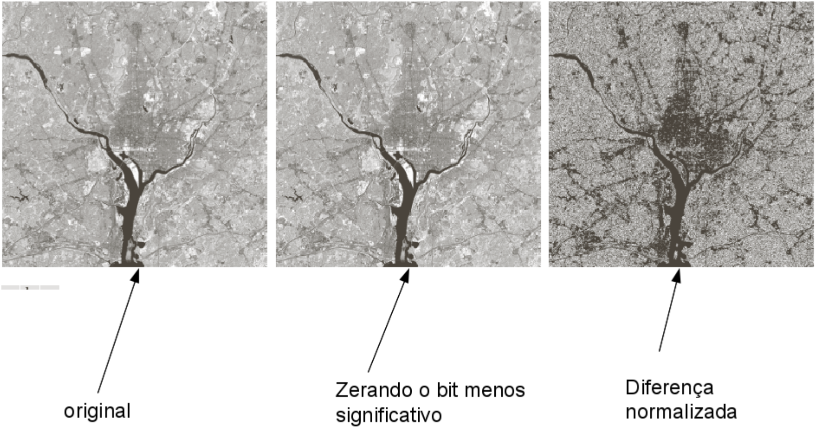


a b c
d e f

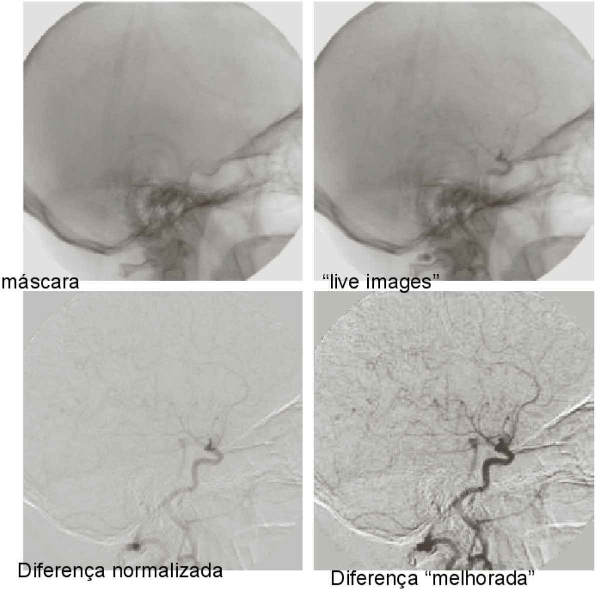
FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

- This procedure works if the noise has a zero average and is uncorrelated to the signal.

Difference


original
Zerando o bit menos significativo
Diferença normalizada

Difference


máscara
"live images"
Diferença normalizada
Diferença "melhorada"

Correction

Original
g(x,y) = f(x,y) h(x,y)

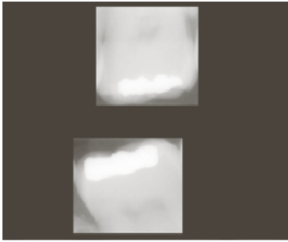
h(x,y)

g'(x,y)=g(x,y)/h(x,y)

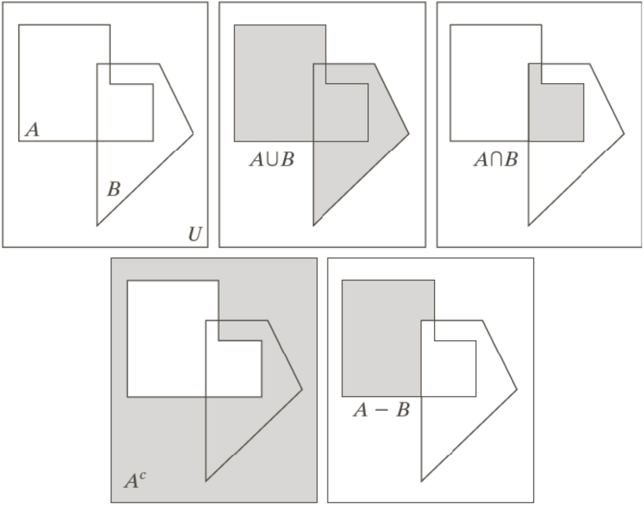
Masking – ROI

original

Máscaras englobando as
Regiões de interesse



Multiplicação do original
pela imagem com másca



a b c
d e

FIGURE 2.31
(a) Two sets of coordinates, A and B, in 2-D space. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B. In (b)–(e) the shaded areas represent the member of the set operation indicated.

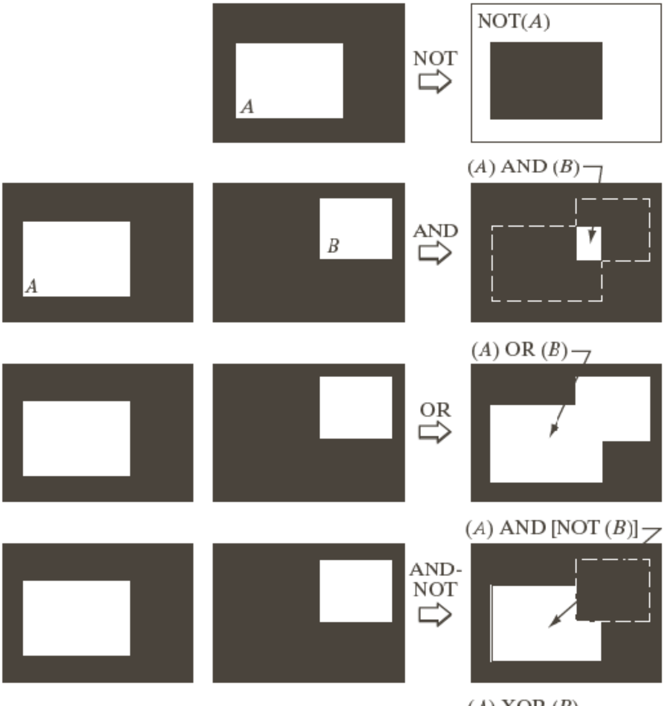


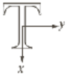
FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Original
f(x,y)

Imagem negativa
255-f(x,y)

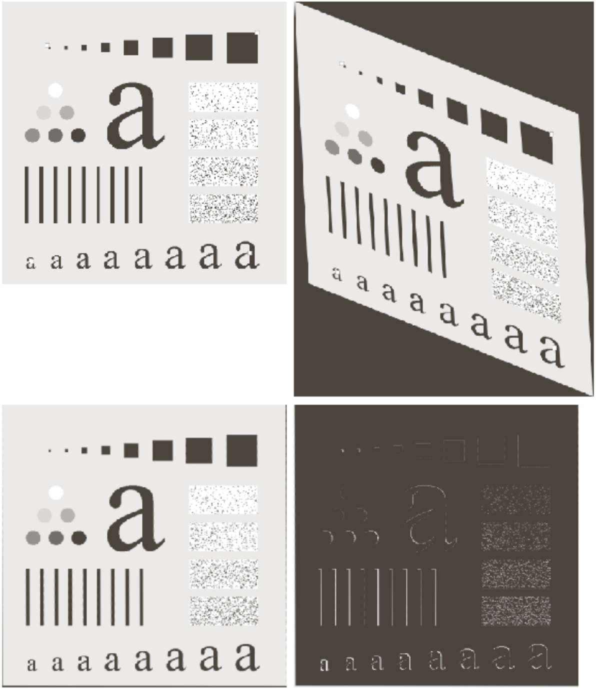
União do original com
uma imagem constante
= {max(a,b)}

TABLE 2.2
Affine transformations based on Eq. (2.6.–23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

a b c d

FIGURE 2.36 (a) A 300 dpi image of the letter T. (b) Image rotated 21° clockwise using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.



a b
c d

FIGURE 2.37 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the borders). (d) Difference between (a) and (c), showing more registration errors.

- One pixel p in the position (x,y) can have the following types of neighbors:
  - N_4 (4-neighborhood of p) – Includes the 4 horizontal and vertical neighbors of p, which have the following coordinates:

$$(x+1, y),\ (x-1, y),\ (x, y+1),\ (x, y-1)$$

  - N_d (diagonal(d)-neighboring of p) – Includes he 4 diagonal neighbors of p, which have the following coordinates:

$$(x+1, y+1),\ (x+1, y-1),\ (x-1, y+1),\ (x-1, y-1)$$

  - N_8 (8-neighboring of p) – Includes the 8 neighbors of p, including the 4 horizontal and vertical neighbors and the 4 diagonal neighbors, which have the following coordinates.

Relation Among Pixels

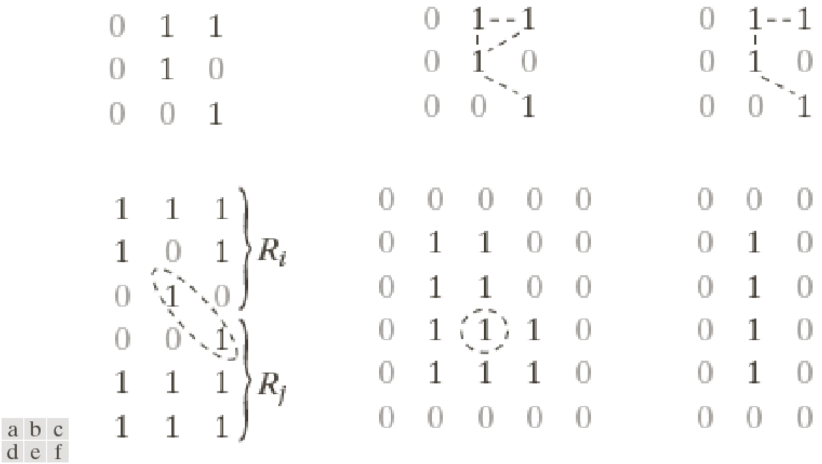


FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) *m*-adjacency. (d) Two regions that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between 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.

- Neighborhood:
  - N_4, N_8, N_d
- Adjacency

Relation Among Pixels

- Neighborhood:
  - N_4, N_8, N_d
- Adjacency
  - Set of neighboring pixels with intensity values from a set V

- Neighborhood:
  - N_4,N_8, N_d
- Adjacency
  - Set of neighboring pixels with intensity values from a set V
  - 4-Adjacency: 2 pixels p and q, with values from the set V, are 4-Adjacent if q is in set $N_4(p)$.

- Neighborhood:
  - N_4,N_8, N_d
- Adjacency
  - Set of neighboring pixels with intensity values from a set V
  - 4-Adjacency: 2 pixels p and q, with values from the set V, are 4-Adjacent if q is in set $N_4(p)$.
  - 8-Adjacency: 2 pixels p e q, with values from the set V, are 8-Adjacent if q is in set $N_8(p)$.

Relation Among Pixels

- Neighborhood:
  - N_4,N_8, N_d
- Adjacency
  - Set of neighboring pixels with intensity values from a set V
  - 4-Adjacency: 2 pixels p and q, with values from the set V, are 4-Adjacent if q is in set $N_4(p)$.
  - 8-Adjacency: 2 pixels p e q, with values from the set V, are 8-Adjacent if q is in set $N_8(p)$.
  - m-Adjacency (mixed adjacency): 2 pixels p and q, with values from the set V, are m-Adjacent if:
    - q is in $N_4(p)$, OR
    - q is in $N_d(p)$ and $N_4(p) \cap N_4(q)$ have no pixels whose values are from V

Relation Among Pixels

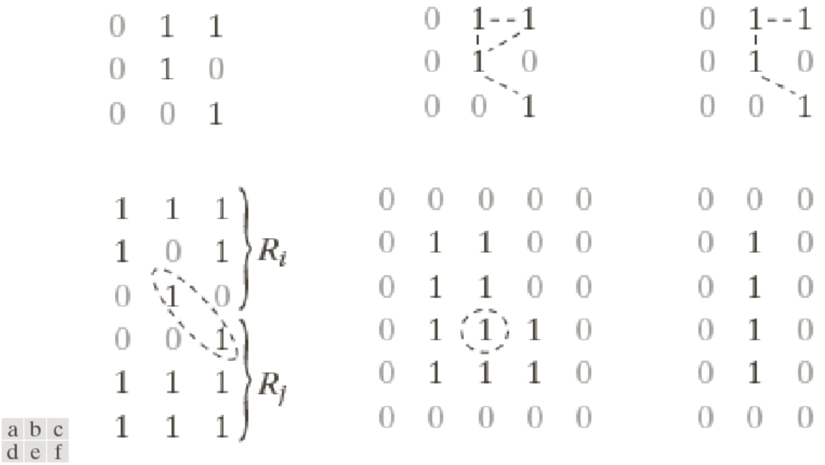


FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) *m*-adjacency. (d) Two regions that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between 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.

- A digital path (or curve) from pixel p with coordinates (x, y) to pixel q with coordinates (s, t) is a sequence of distinct pixels with coordinates:

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

where $(x_0, y_0) = (x, y)$, $(x_n, y_n) = (s, t)$ and the pixels (x_i, y_i) e (x_{i-1}, y_{i-1}) are adjacent for $i \leq n$.

Paths

- A digital path (or curve) from pixel p with coordinates (x, y) to pixel q with coordinates (s, t) is a sequence of distinct pixels with coordinates:

$$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

  where $(x_0, y_0) = (x, y)$, $(x_n, y_n) = (s, t)$ and the pixels (x_i, y_i) e (x_{i-1}, y_{i-1}) are adjacent for $i \leq n$.
  - n is the length of the path;
  - closed path: $(x_0, y_0) = (x_n, y_n)$
  - We can define paths with 1, or m Adjacency

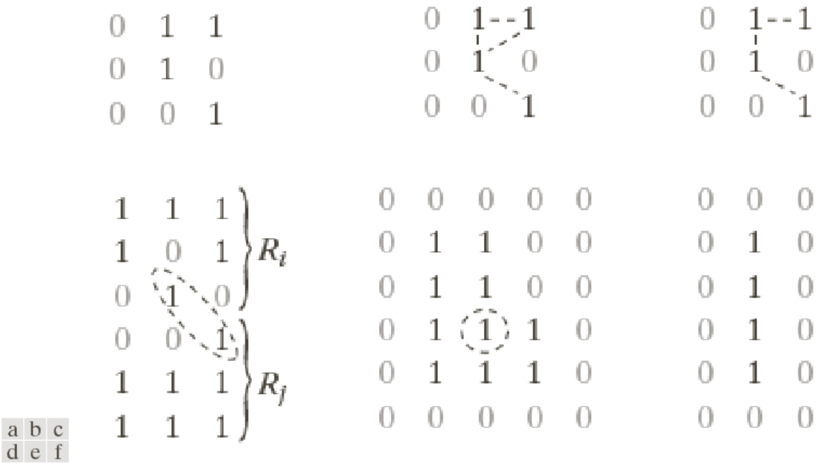


FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) *m*-adjacency. (d) Two regions that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between 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.

- Conectivity: Be S a subset of pixels:
  - p and q are a *connected set* if there is a path between p and q, which consists of pixels from S;
  - if p has only one connected set, the set S is connected;
  - For any pixel p in S, the set of pixels that are connected to it in S is known as a the *connected component* of S.
  - R is a *region* of the image if R is a connected set.
  - Two regions R_i e R_j are *adjacent* if their union forms a connected set. Regions that are not adjacent are said to be *disjoint*.
  - We consider 4- and 8-adjacency when referring to regions.

Adjacency

- Background vs. foreground
- Border or contour
  - Inner Border – Set of points that are adjacent to the points in the points in the complement of R (at leat one neighbor in the background)
  - Outer Border
- Distance Measures D

- Considering the pixels $p(x, y)$, $q(s, t)$, and $z(v, w)$, D is a distance measure:

$$D(p, q) \geq 0$$

$$D(p, q) = D(q, p)$$

$$D(p, z) \leq D(p, q) + D(q, z)$$

- D_4 (city block)

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

- Euclidian Distance:

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

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

D_4 distance ≤ 2 from (x, y)

Distances

- D_8 (chessboard distance):

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

- D_m (shortest path m)
  - $p = p_2 = p_4 = 1$
  - If $p_1 = p_3 = 0$, then $D_m(p, p_4) = 2$
  - and if $p_1 = 1$? $D_m(p, p_4) = 3$

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

D_8 distance ≤ 2 from (x, y)

	p_3	p_4
p_1	p_2	
p		