مبانی بینایی کامپیوتری

درس ۵ب

فیلتر کردن در حوزه‌ی مکان در متلب

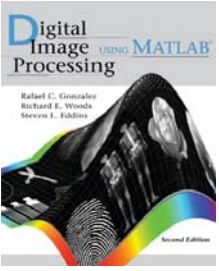
Spatial Filtering in MATLAB®

کاظم فولادی قلعه

دانشکده مهندسی، پردیس فارابی

دانشگاه تهران

http://courses.fouladi.ir/fcvision



Chapter 3
Intensity Transformations and Spatial Filtering

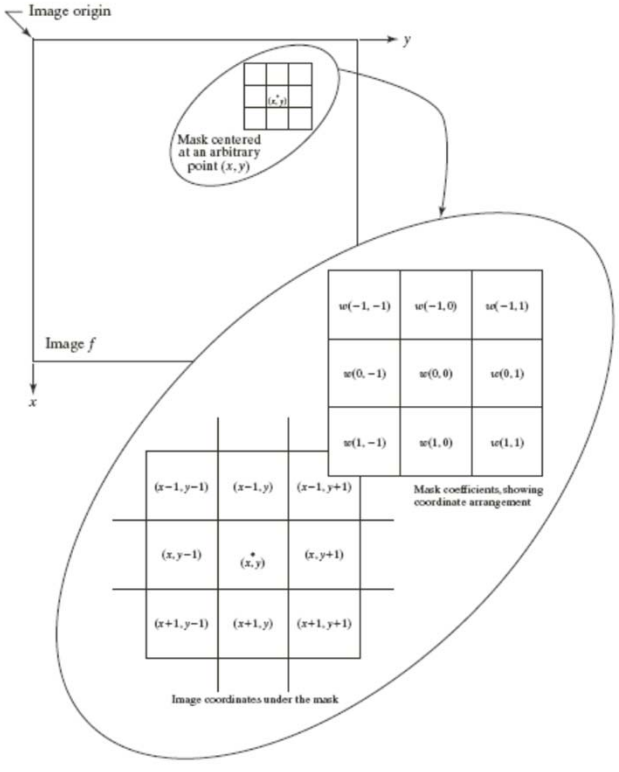
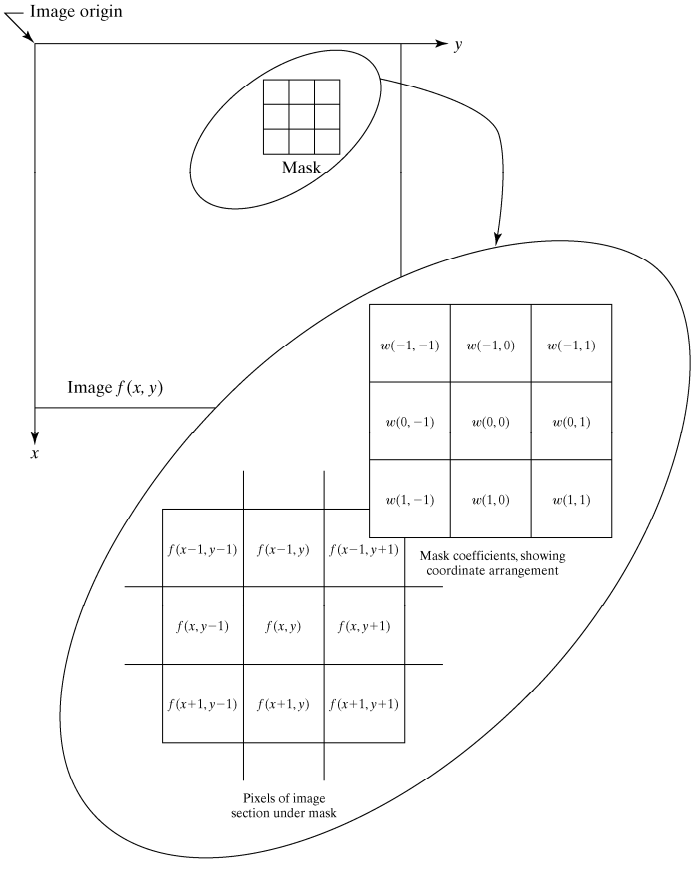
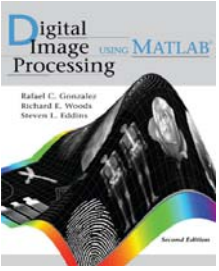


FIGURE 2.13 The mechanics of linear spatial filtering. The magnified drawing shows a 3×3 filter mask and the corresponding image neighborhood directly under it. The image neighborhood is shown displaced out from under the mask for ease of readability.

ساز و کار فیلتر کردن مکانی خطی



The mechanics of linear spatial filtering. The magnified drawing shows a 3×3 mask and the corresponding image neighborhood directly under it. The neighborhood is shown displaced out from under the mask for ease of readability.



Chapter 3
Intensity Transformations and Spatial Filtering

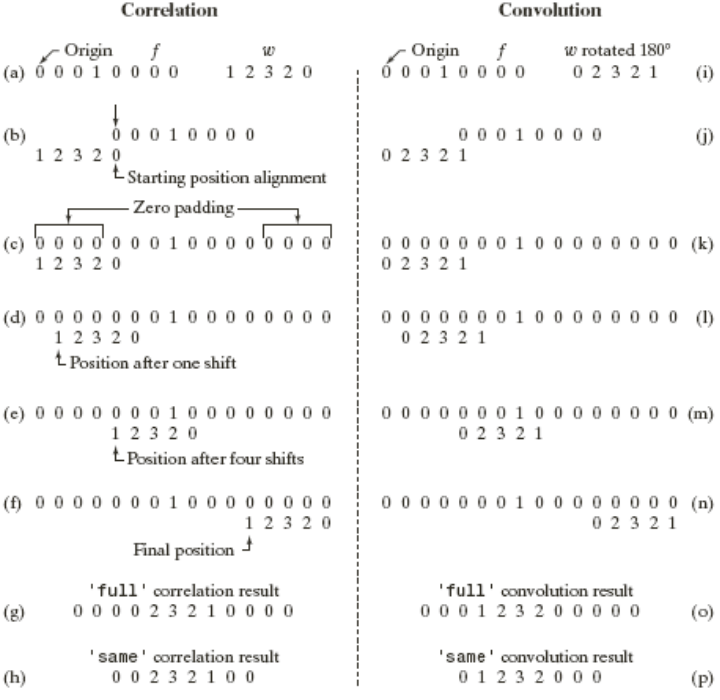


FIGURE 2.14 Illustration of one-dimensional correlation and convolution.

کانوولوشن و همبستگی

CONVOLUTION & CORRELATION

$$w(x,y) \circ f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

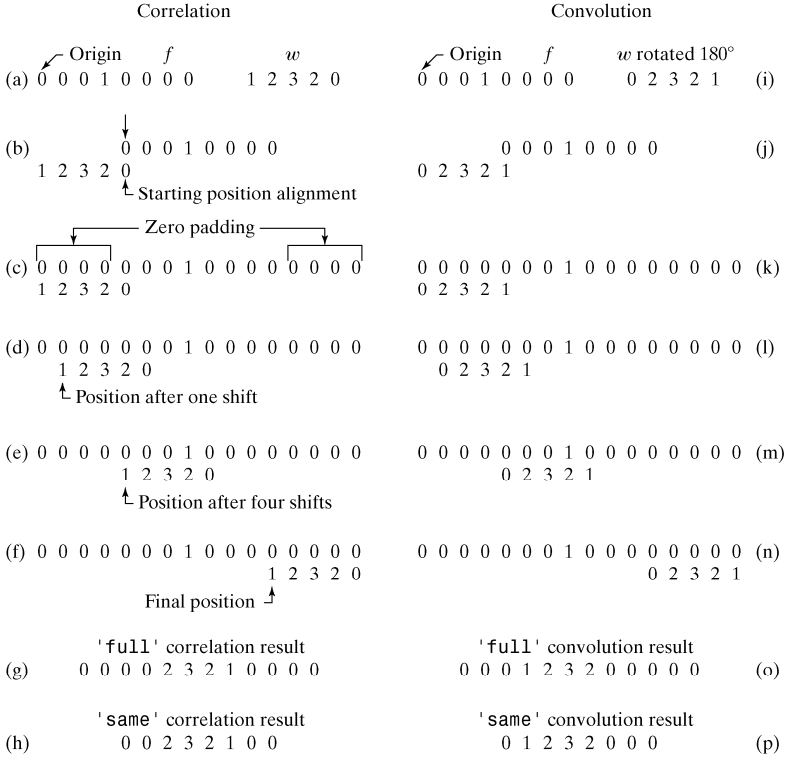
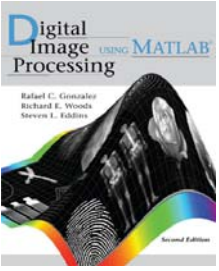


Illustration of one-dimensional correlation and convolution.



Chapter 3
Intensity Transformations and Spatial Filtering

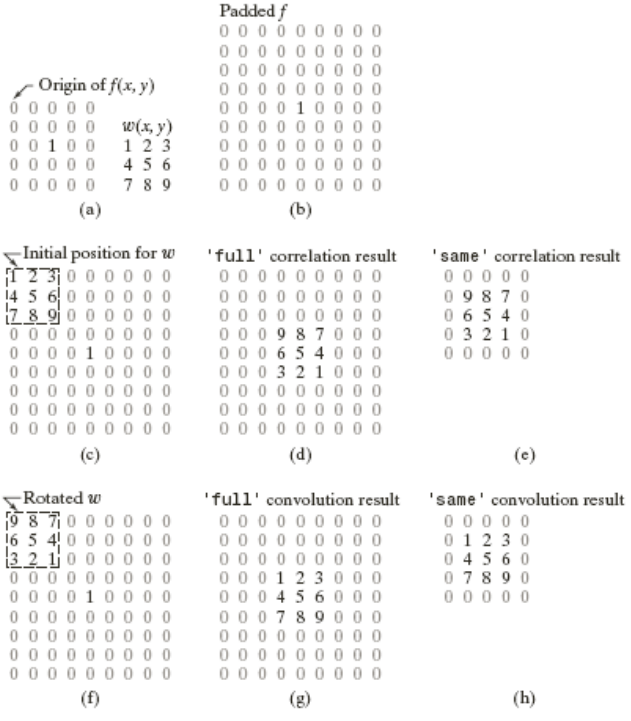


FIGURE 2.15 Illustration of two-dimensional correlation and convolution. The 0s are shown in gray to simplify viewing.

کانوولوشن و همبستگی

2D Convolution & Correlation

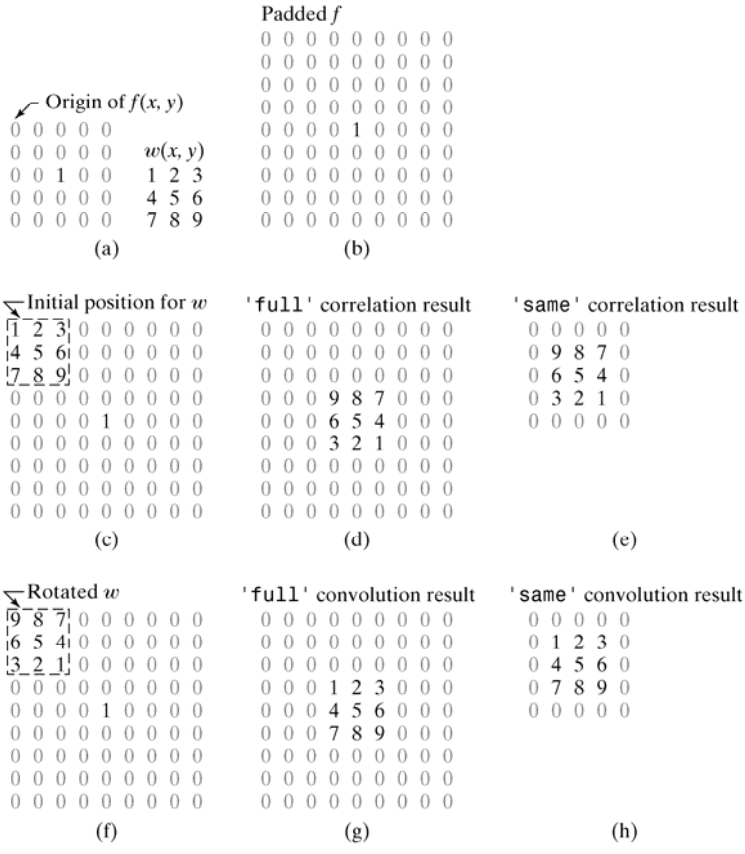
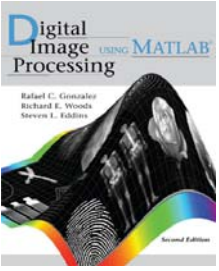


Illustration of two-dimensional correlation and convolution. The 0s are shown in gray to simplify viewing.



Chapter 3
Intensity Transformations and Spatial Filtering

Options	Description
Filtering Mode	
`'corr'`	Filtering is done using correlation (see Figs. 2.14 and 2.15). This is the default.
`'conv'`	Filtering is done using convolution (see Figs. 2.14 and 2.15).
Boundary Options	
`P`	The boundaries of the input image are extended by padding with a value, `P` (written without quotes). This is the default, with value 0.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'circular'`	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
`'full'`	The output is of the same size as the extended (padded) image (see Figs. 2.14 and 2.15).
`'same'`	The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image (see Figs. 2.14 and 2.15). This is the default.

TABLE 2.3
Options for function `imfilter`.

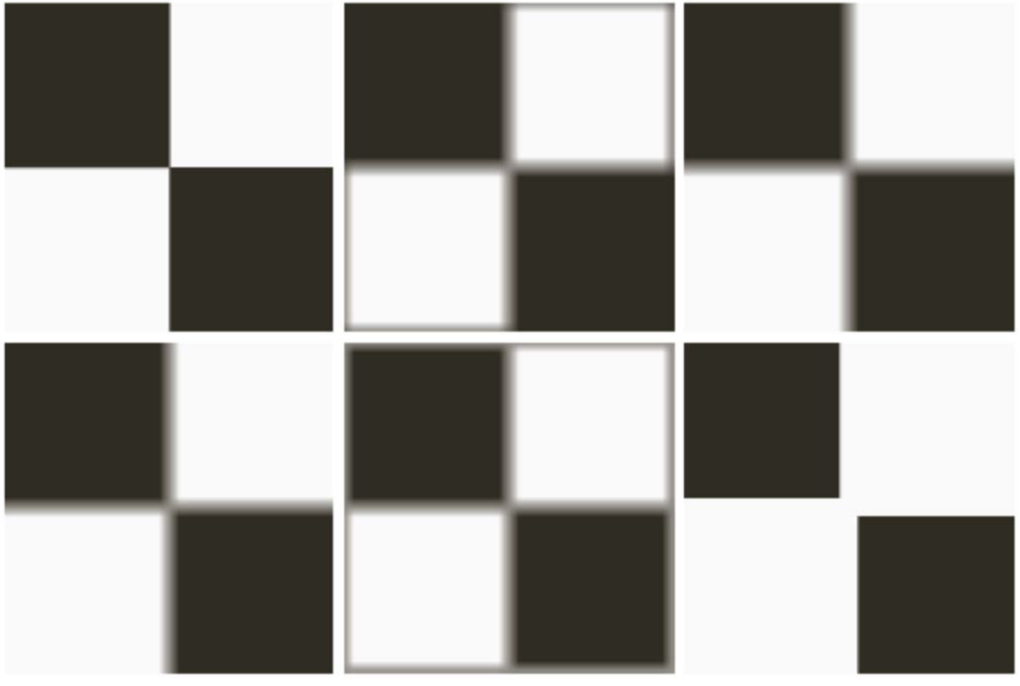
تابع فیلتر تصویر

```
g = imfilter(f, w, filtering_mode, boundary_options, size_options)
```

Options	Description
Filtering Mode	
`'corr'`	Filtering is done using correlation (see Figs. 3.13 and 3.14). This is the default.
`'conv'`	Filtering is done using convolution (see Figs. 3.13 and 3.14).
Boundary Options	
`P`	The boundaries of the input image are extended by padding with a value, `P` (written without quotes). This is the default, with value 0.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'circular'`	The size of the image is extended by treating the image as one period a 2-D periodic function.
Size Options	
`'full'`	The output is of the same size as the extended (padded) image (see Figs. 3.13 and 3.14).
`'same'`	The output is of the same size as the input. This is achieved by limiting the excursions of the center of the filter mask to points contained in the original image (see Figs. 3.13 and 3.14). This is the default.

Options for function `imfilter`.

Chapter 3
Intensity Transformations and Spatial Filtering

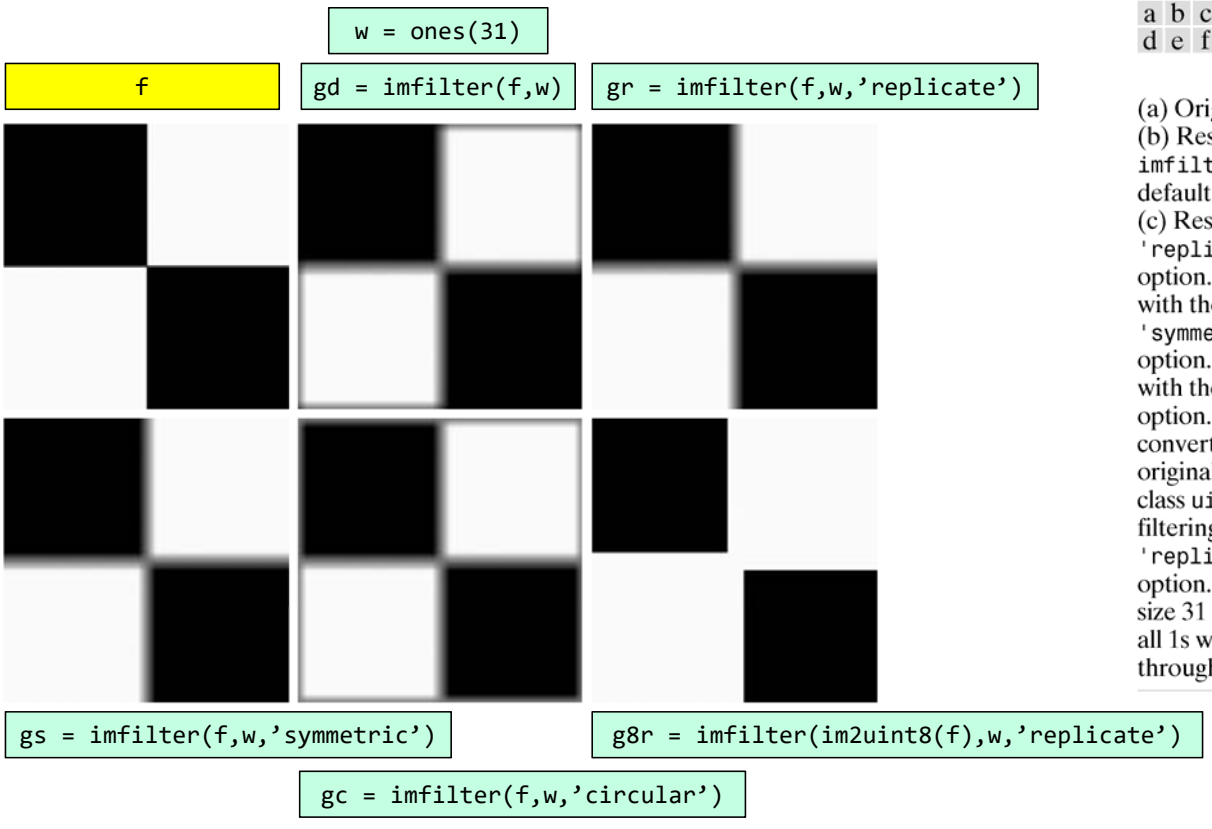


a b c
d e f

FIGURE 2.16
(a) Original image. (b) Result of using `imfilter` with default zero padding. (c) Result with the `'replicate'` option. (d) Result with the `'symmetric'` option. (e) Result with the `'circular'` option. (f) Result of converting the original image to class `uint8` and then filtering with the `'replicate'` option. A filter of size 31×31 with all 1s was used throughout.

تابع فیلتر تصویر

مثال



a b c
d e f

(a) Original image. (b) Result of using `imfilter` with default zero padding. (c) Result with the `'replicate'` option. (d) Result with the `'symmetric'` option. (e) Result with the `'circular'` option. (f) Result of converting the original image to class `uint8` and then filtering with the `'replicate'` option. A filter of size 31 × 31 with all 1s was used throughout.

Chapter 3
Intensity Transformations and Spatial Filtering

Options	Description
Method	
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'circular'`	The size of the image is extended by treating the image as one period of a 2-D periodic function.
Direction	
`'pre'`	Pad before the first element of each dimension.
`'post'`	Pad after the last element of each dimension.
`'both'`	Pad before the first element and after the last element of each dimension. This is the default.

TABLE 2.4 Options for function `padarray`.

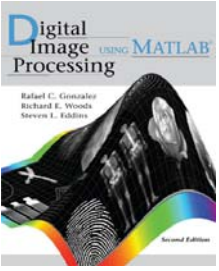
تابع پدینگ

PADDING FUNCTION

```
fp = padarray(f, [r c], method, direction)
```

Options	Description
Method	
`'symmetric'`	The size of the image is extended by mirror-reflecting it across its border.
`'replicate'`	The size of the image is extended by replicating the values in its outer border.
`'circular'`	The size of the image is extended by treating the image as one period of a 2-D periodic function.
Direction	
`'pre'`	Pad before the first element of each dimension.
`'post'`	Pad after the last element of each dimension.
`'both'`	Pad before the first element and after the last element of each dimension. This is the default.

Options for function `padarray`.



Chapter 3
Intensity Transformations and Spatial Filtering

Type	Syntax and Parameters
`'average'`	`fspecial('average', [r c])`. A rectangular averaging filter of size `r` × `c`. The default is 3 × 3. A single number instead of `[r c]` specifies a square filter.
`'disk'`	`fspecial('disk', r)`. A circular averaging filter (within a square of size `2r + 1`) with radius `r`. The default radius is 5.
`'gaussian'`	`fspecial('gaussian', [r c], sig)`. A Gaussian lowpass filter of size `r` × `c` and standard deviation `sig` (positive). The defaults are 3 × 3 and 0.5. A single number instead of `[r c]` specifies a square filter.
`'laplacian'`	`fspecial('laplacian', alpha)`. A 3 × 3 Laplacian filter whose shape is specified by `alpha`, a number in the range [0, 1]. The default value for `alpha` is 0.2.
`'log'`	`fspecial('log', [r c], sig)`. Laplacian of a Gaussian (LoG) filter of size `r` × `c` and standard deviation `sig` (positive). The defaults are 5 × 5 and 0.5. A single number instead of `[r c]` specifies a square filter.
`'motion'`	`fspecial('motion', len, theta)`. Outputs a filter that, when convolved with an image, approximates linear motion (of a camera with respect to the image) of `len` pixels. The direction of motion is `theta`, measured in degrees, counterclockwise from the horizontal. The defaults are 9 and 0, which represents a motion of 9 pixels in the horizontal direction.
`'prewitt'`	`fspecial('prewitt')`. Outputs a 3 × 3 Prewitt filter, `wv`, that approximates a vertical gradient. A filter mask for the horizontal gradient is obtained by transposing the result: `wh = wv'`.
`'sobel'`	`fspecial('sobel')`. Outputs a 3 × 3 Sobel filter, `sv`, that approximates a vertical gradient. A filter for the horizontal gradient is obtained by transposing the result: `sh = sv'`.
`'unsharp'`	`fspecial('unsharp', alpha)`. Outputs a 3 × 3 unsharp filter; `alpha` controls the shape; it must be in the range [0, 1]; the default is 0.2.

TABLE 2.5
Spatial filters supported by function `fspecial`. Several of the filters in this table are used for edge detection in Section 10.1.

فیلترهای مکانی استاندارد جعبه‌ابزار پردازش تصویر

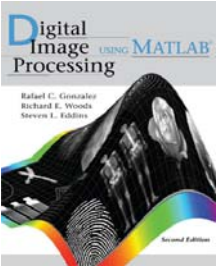
فیلترهای خطی

```
w = fspecial(
  ‘type’,
  ’parameter’)
```

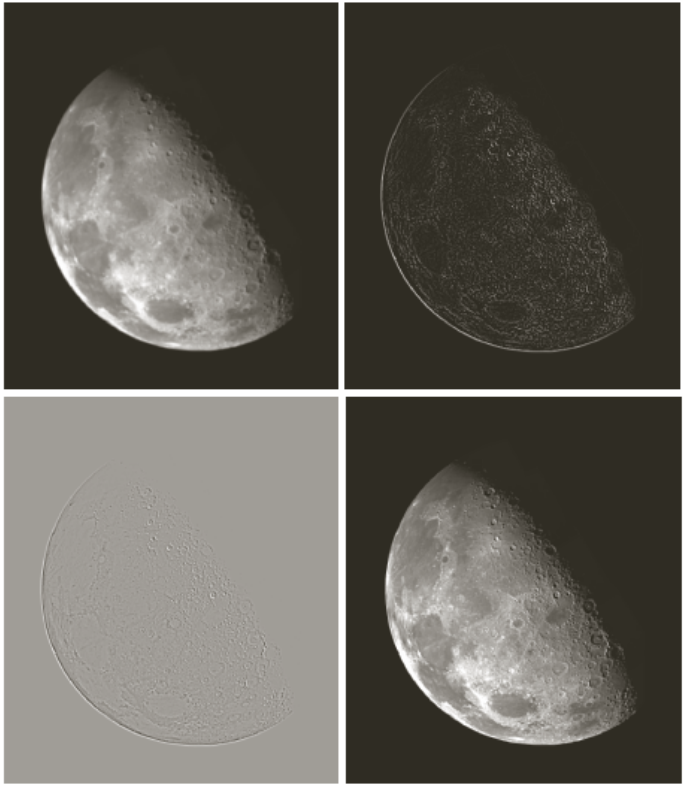
تابع برای تولید ماسک

Type	Syntax and Parameters
`'average'`	`fspecial('average', [r c])`. A rectangular averaging filter of size `r` × `c`. The default is 3 × 3. A single number instead of `[r c]` specifies a square filter.
`'disk'`	`fspecial('disk', r)`. A circular averaging filter (within a square of size `2r + 1`) with radius `r`. The default radius is 5.
`'gaussian'`	`fspecial('gaussian', [r c], sig)`. A Gaussian lowpass filter of size `r` × `c` and standard deviation `sig` (positive). The defaults are 3 × 3 and 0.5. A single number instead of `[r c]` specifies a square filter.
`'laplacian'`	`fspecial('laplacian', alpha)`. A 3 × 3 Laplacian filter whose shape is specified by `alpha`, a number in the range [0, 1]. The default value for `alpha` is 0.5.
`'log'`	`fspecial('log', [r c], sig)`. Laplacian of a Gaussian (LoG) filter of size `r` × `c` and standard deviation `sig` (positive). The defaults are 5 × 5 and 0.5. A single number instead of `[r c]` specifies a square filter.
`'motion'`	`fspecial('motion', len, theta)`. Outputs a filter that, when convolved with an image, approximates linear motion (of a camera with respect to the image) of `len` pixels. The direction of motion is `theta`, measured in degrees, counterclockwise from the horizontal. The defaults are 9 and 0, which represents a motion of 9 pixels in the horizontal direction.
`'prewitt'`	`fspecial('prewitt')`. Outputs a 3 × 3 Prewitt mask, `wv`, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: `wh = wv'`.
`'sobel'`	`fspecial('sobel')`. Outputs a 3 × 3 Sobel mask, `sv`, that approximates a vertical gradient. A mask for the horizontal gradient is obtained by transposing the result: `sh = sv'`.
`'unsharp'`	`fspecial('unsharp', alpha)`. Outputs a 3 × 3 unsharp filter. Parameter `alpha` controls the shape; it must be greater than 0 and less than or equal to 1.0; the default is 0.2.

Spatial filters supported by function `fspecial`.



Chapter 3
Intensity Transformations and Spatial Filtering



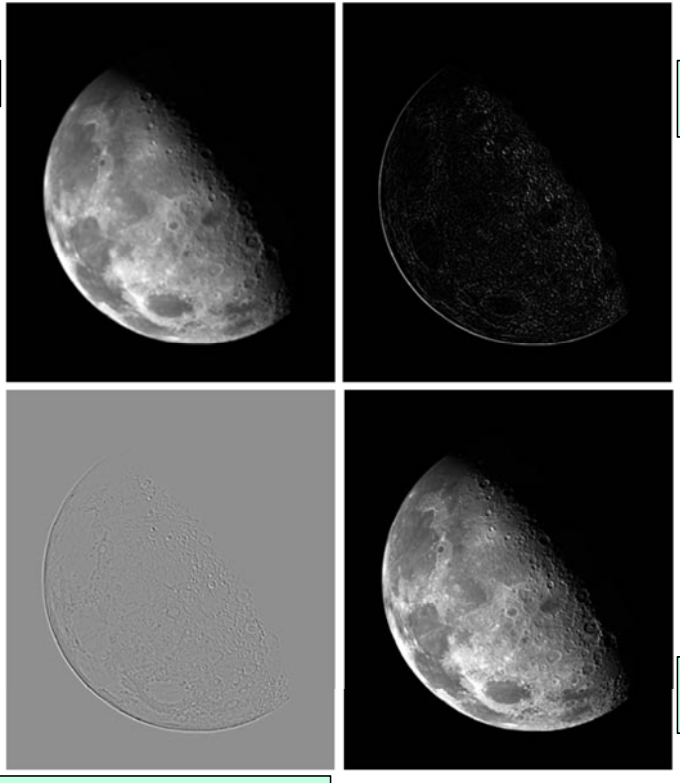
a b
c d

FIGURE 2.17
(a) Image of the North Pole of the moon.
(b) Laplacian filtered image, using `uint8` format. (Because `uint8` is an unsigned type, negative values in the output were clipped to 0.)
(c) Laplacian filtered image obtained using floating point.
(d) Enhanced result, obtained by subtracting (c) from (a).
(Original image courtesy of NASA.)

فیلترهای مکانی استاندارد جعبه‌ابزار پردازش تصویر

فیلترهای خطی: مثال

f



```
w = fspecial('Laplacian', 0);
g1 = imfilter(f,w,'replicate');
```

اشکال: عدم نمایش پیکسل‌های منفی

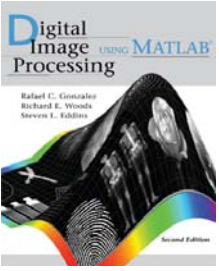
```
f2 = tofloat(f);
g2 = imfilter(f2,w,'replicate');
```

```
g = f2 - g2;
imshow(g)
```

a b
c d

(a) Image of the North Pole of the moon.
(b) Laplacian filtered image, using `uint8` formats.
(c) Laplacian filtered image obtained using `double` formats.
(d) Enhanced result, obtained by subtracting (c) from (a).

MATLAB



Chapter 3
Intensity Transformations and Spatial Filtering

a
b c

FIGURE 2.18
(a) Image of the North Pole of the moon. (b) Image enhanced using the Laplacian filter `'laplacian'`, which has a −4 in the center. (c) Image enhanced using a Laplacian filter with a −8 in the center.

فیلترهای مکانی استاندارد جعبه‌ابزار پردازش تصویر

فیلترهای خطی: مثال

```
f
```

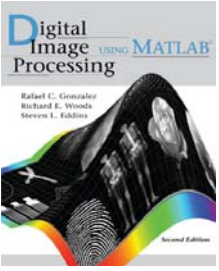
```
w4 = fspecial('Laplacian', 0);
w8 = [1 1 1; 1 -8 1; 1 1 1];
f = toflat(f);
```

```
g4 = f - imfilter(f,w4,'replicate');
```

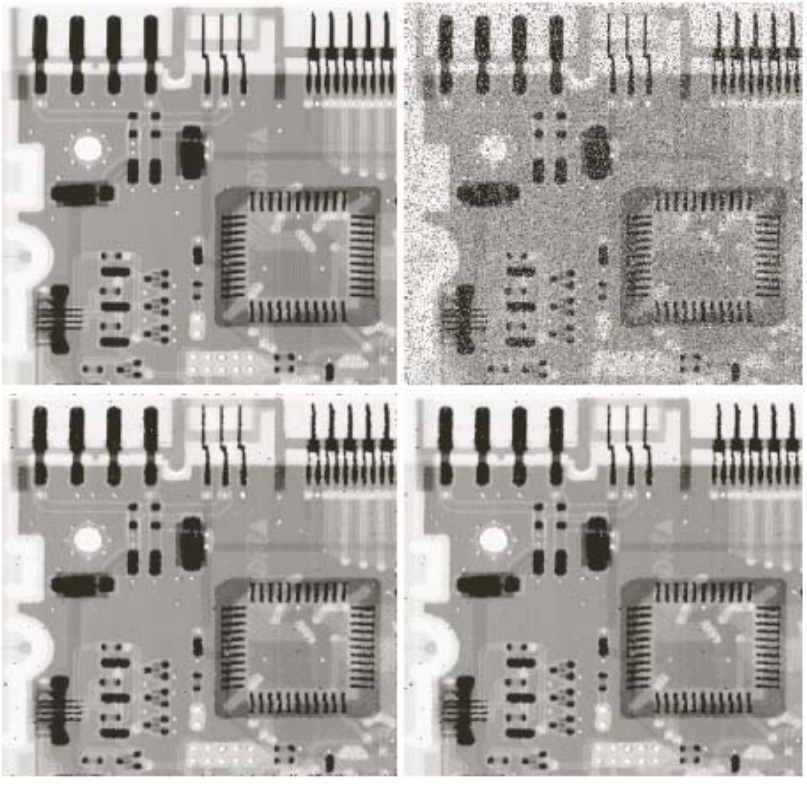
```
g8 = f - imfilter(f,w8,'replicate');
```

a
b c

Image of the North Pole of the moon. (b) Image enhanced using the Laplacian filter `'laplacian'`, which has a −4 in the center. (c) Image enhanced using a Laplacian filter with a −8 in the center.



Chapter 3
Intensity Transformations and Spatial Filtering



a b
c d

FIGURE 2.19 Median filtering: (a) X-ray image. (b) Image corrupted by salt-and-pepper noise. (c) Result of median filtering with `medfilt2` using the default settings. (d) Result of median filtering using the `'symmetric'` option. Note the improvement in border behavior between (d) and (c). (Original image courtesy of Lixi, Inc.)

فیلترهای مکانی استاندارد جعبه‌ابزار پردازش تصویر

فیلترهای غیرخطی

فیلترهای مرتبه آماری (رتبه)
Order-Statistic (Rank) Filters

```
g = ordfilt2(f, order, domain)
```

```
g = ordfilt2(f, 1, ones(m, n))
```

فیلتر می‌نیمم
Min Filter

فیلتر میانه
Median Filter

```
g = medfilt2(f, [m n], padopt)
```

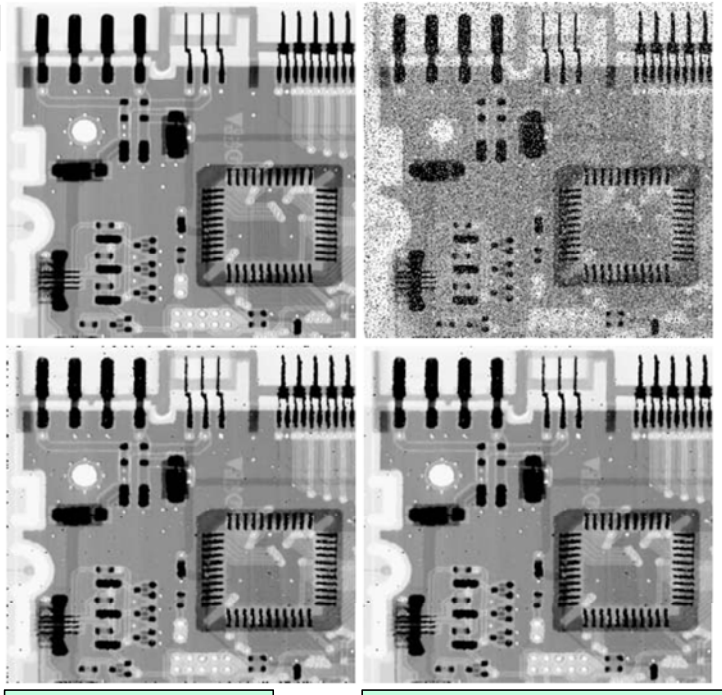
```
g = medfilt2(f)
```

فیلترهای مکانی استاندارد جعبه‌ابزار پردازش تصویر

فیلترهای غیرخطی: مثال

```
fn = imnoise(f, 'salt & pepper', 0.2);
```

f



```
gm = medfilt2(fn);
```

```
gms = medfilt2(fn, 'symmetric');
```

a b
c d

Median filtering. (a) X-ray image. (b) Image corrupted by salt-and-pepper noise. (c) Result of median filtering with `medfilt2` using the default settings. (d) Result of median filtering using the `'symmetric'` image extension option. Note the improvement in border behavior between (d) and (c).

نویز در کناره‌های تصویر بهتر شده است.

MATLAB

استفاده از تکنیک‌های فازی برای تبدیل‌های شدت رنگ و فیلتر کردن مکانی

USING FUZZY TECHNIQUES FOR INTENSITY TRANSFORMATIONS AND SPATIAL FILTERING

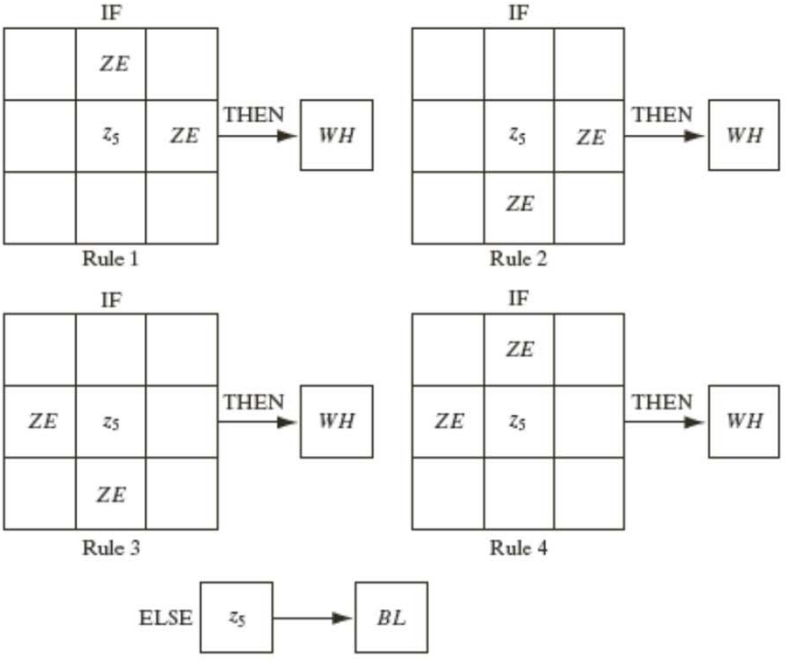
Fuzzy rules for boundary detection.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

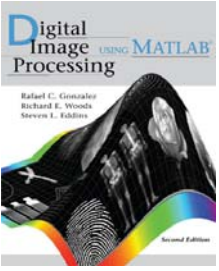
Pixel neighborhood

d_1	d_2	d_3
d_4	0	d_6
d_7	d_8	d_9

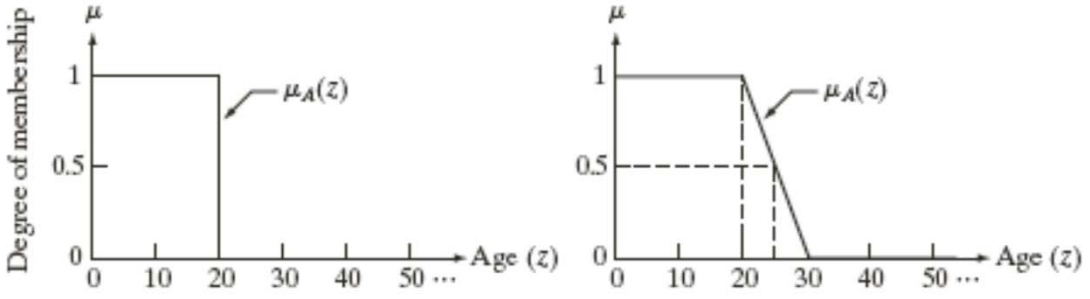
Intensity differences



MATLAB

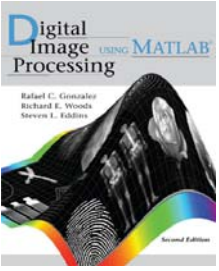


Chapter 3
Intensity Transformations and Spatial Filtering

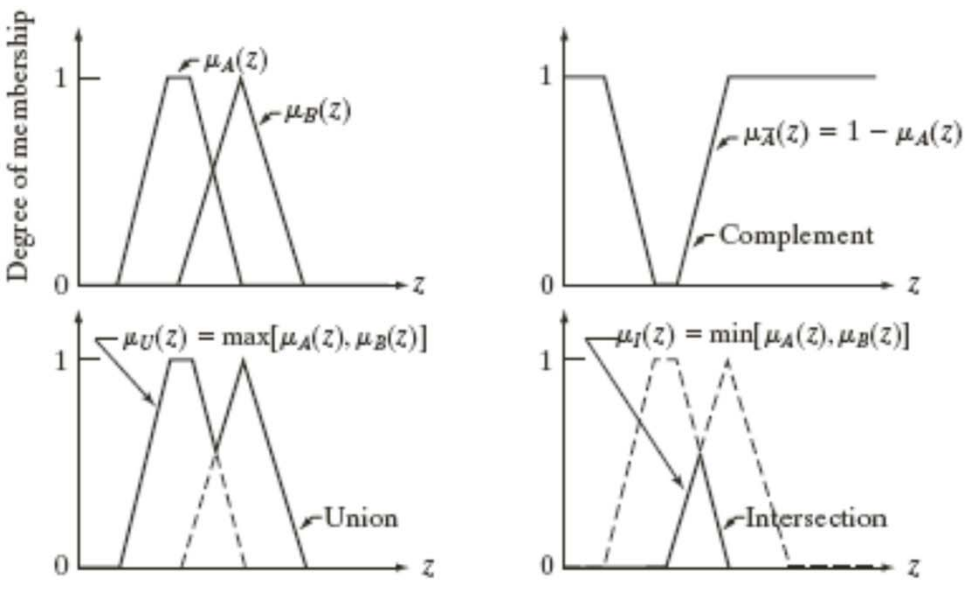


a b

FIGURE 2.20 Membership functions of (a) a crisp set, and (b) a fuzzy set.



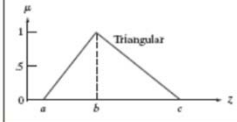
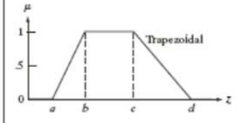
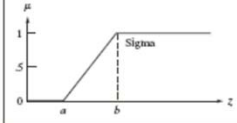
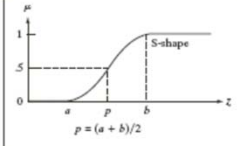
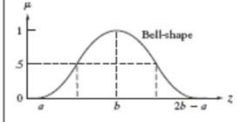
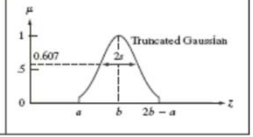
Chapter 3
Intensity Transformations and Spatial Filtering



a b
c d

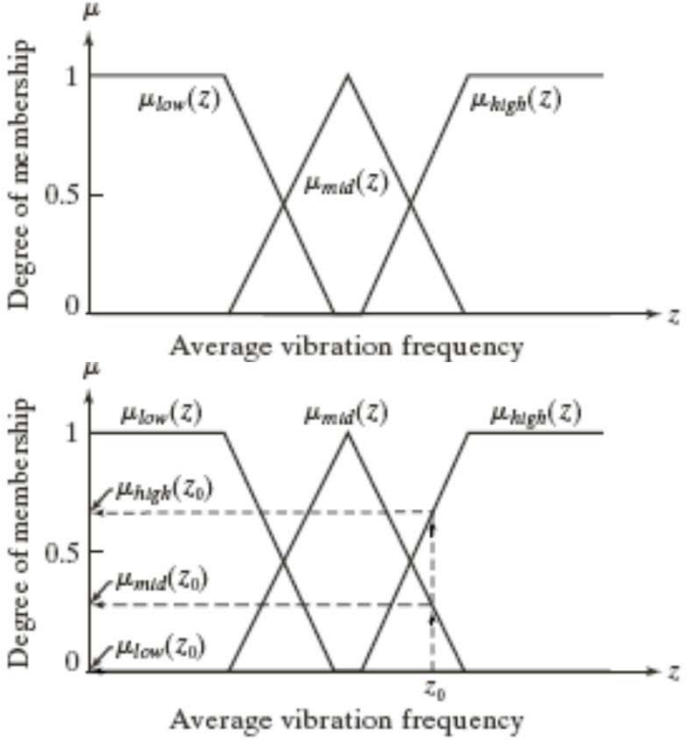
FIGURE 2.21
(a) Membership functions of two fuzzy sets, A and B. (b) Membership function of the complement of A. (c) and (d) Membership functions of the union and intersection of A and B.

TABLE 2.6 Some commonly-used membership functions and corresponding plots.

Name	Equation	Plot
Triangular	$\mu(z) = \begin{cases} 0 & z < a \\ (z-a)/(b-a) & a \le z < b \\ 1-(z-b)/(c-b) & b \le z < c \\ 0 & c \le z \end{cases}$	
Trapezoidal	$\mu(z) = \begin{cases} 0 & z < a \\ (z-a)/(b-a) & a \le z < b \\ 1 & b \le z < c \\ 1-(z-b)/(c-b) & c \le z < d \\ 0 & d \le z \end{cases}$	
Sigma	$\mu(z) = \begin{cases} 0 & z < a \\ (z-a)/(b-a) & a \le z < b \\ 1 & b \le z \end{cases}$	
S-shape†	$S(z,a,b) = \begin{cases} 0 & z < a \\ 2\left[\frac{z-a}{b-a}\right]^2 & a \le z < p \\ 1-2\left[\frac{z-b}{b-a}\right]^2 & p \le z < b \\ 1 & b \le z \end{cases}$	
Bell-shape	$\mu(z) = \begin{cases} S(z,a,b) & z < b \\ S(2b-z,a,b) & b \le z \end{cases}$	
Truncated Gaussian	$\mu(z) = \begin{cases} e^{-\frac{(z-b)^2}{\sigma^2}} & \lvert z-b \rvert \le (b-a) \\ 0 & \text{otherwise} \end{cases}$	

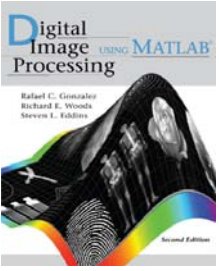
†Typically, only the independent variable, z, is used as an argument when writing $\mu(z)$ in order to simplify notation. We made an exception in the S-shape curve in order to use its form in writing the equation of the Bell-shape curve.

Chapter 3
Intensity Transformations and Spatial Filtering



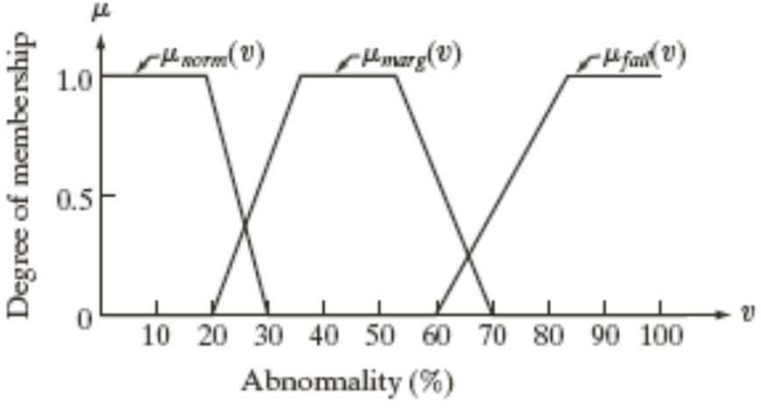
a
b

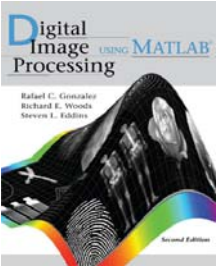
FIGURE 2.22
(a) Membership functions used to fuzzify frequency measurements. (b) Fuzzifying a specific measurement, z_0.



Chapter 3
Intensity Transformations and Spatial Filtering

FIGURE 2.23
Membership functions used for characterizing the fuzzy conditions *normal*, *marginal*, and *near failure*.

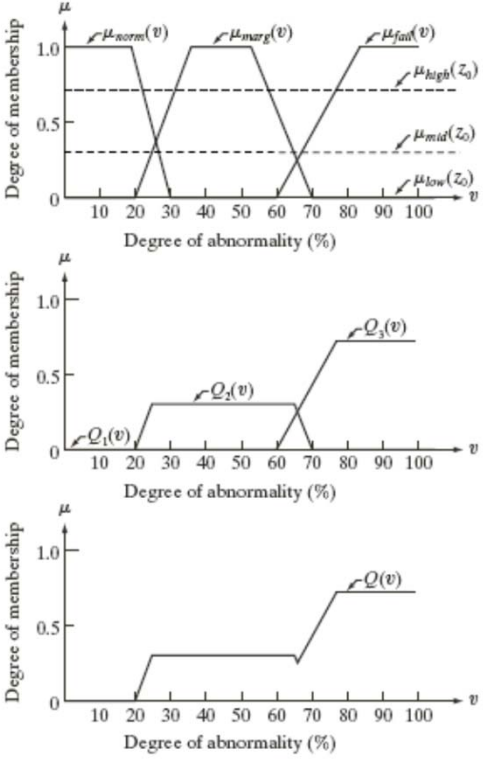


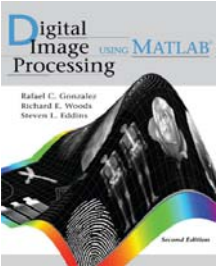


Chapter 3
Intensity Transformations and Spatial Filtering

a
b
c

FIGURE 2.24 Values of μ_{low}, μ_{mid}, and μ_{high} evaluated at z_0 (all three are constant values). (b) Individual outputs. (c) Aggregated output.





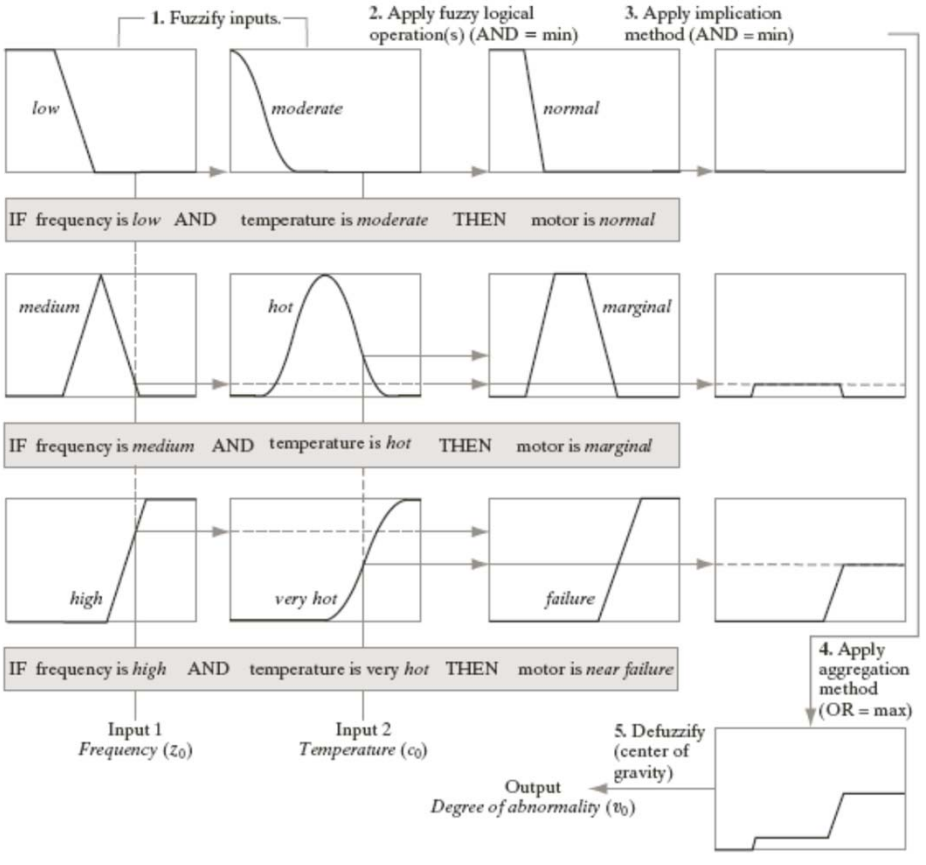
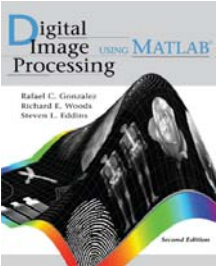


FIGURE 2.25 Example illustrating the five basic steps used typically to implement a fuzzy rule-based system: (1) fuzzification, (2) logical operations, (3) implication, (4) aggregation, and (5) defuzzification.



Chapter 3
Intensity Transformations and Spatial Filtering

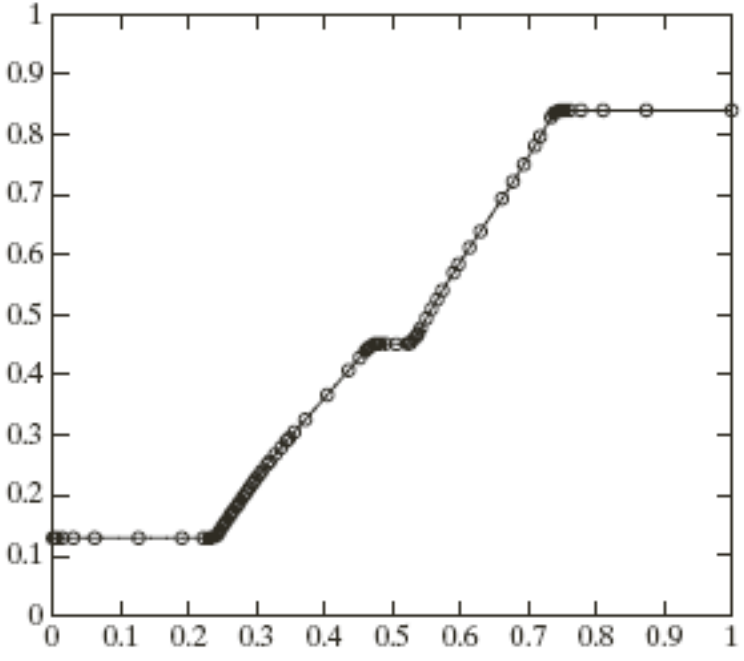
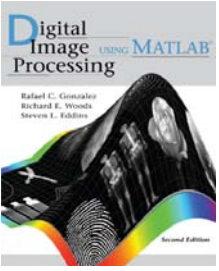


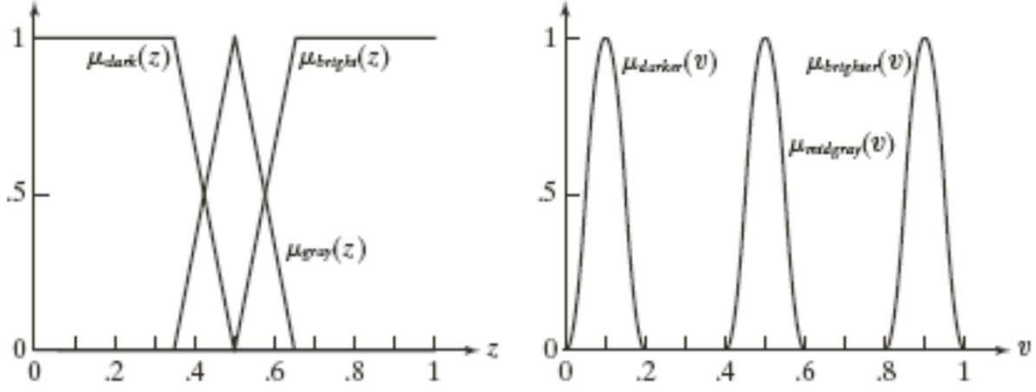
FIGURE 2.26 Comparison between the outputs of functions `fuzzysysfcn` (plotted as a solid line) and `approxfcn` (plotted as circles connected by dots). The results are visually indistinguishable. (Recall from Section 2.3.1 that `fplot` distributes the distance between plot point non-uniformly.)

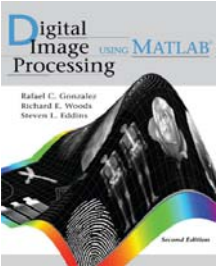


Chapter 3
Intensity Transformations and Spatial Filtering

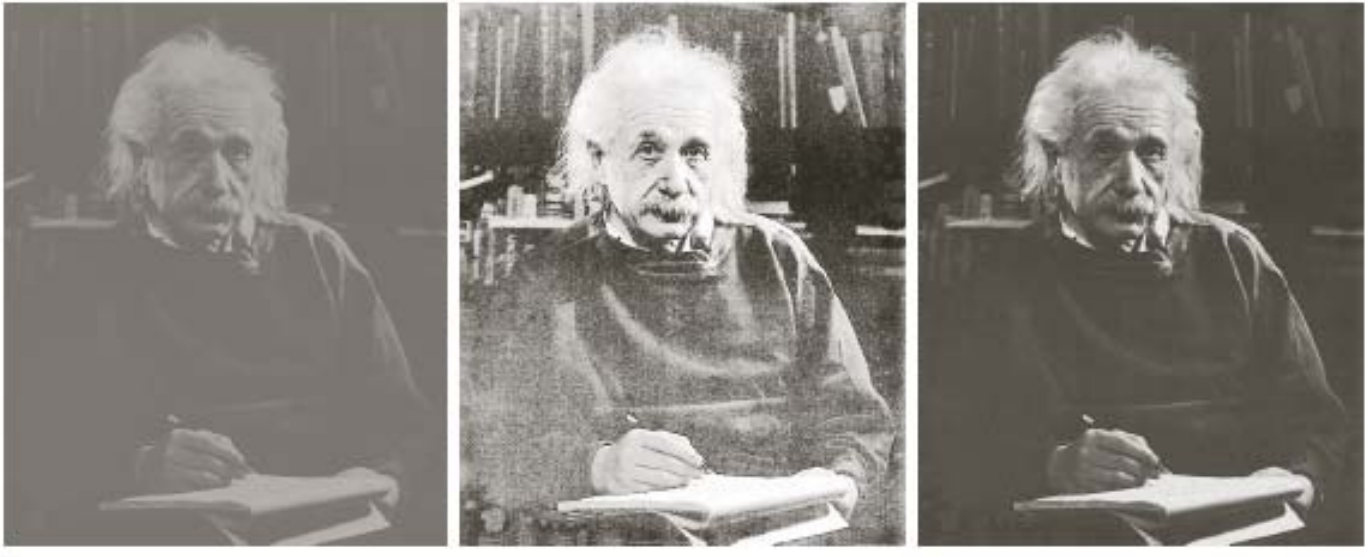
a b

FIGURE 2.27
(a) Input and (b) output membership functions for fuzzy, rule-based contrast enhancement.



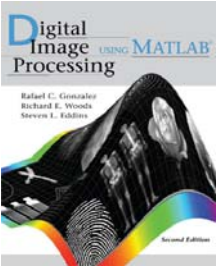


Chapter 3
Intensity Transformations and Spatial Filtering

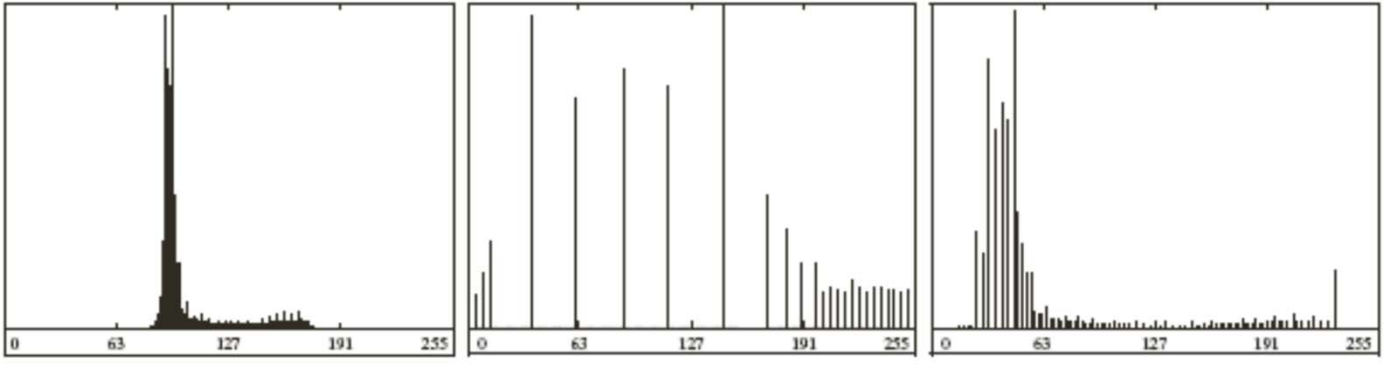


a b c

FIGURE 2.28 (a) Low-contrast image. (b) Result of histogram equalization. (c) Result of fuzzy, rule-based, contrast enhancement.

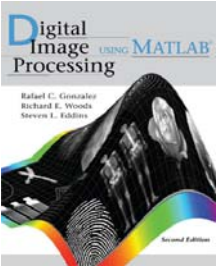


Chapter 3
Intensity Transformations and Spatial Filtering



a b c

FIGURE 2.29 Histograms of the images in Fig. 2.28(a), (b), and (c), respectively.



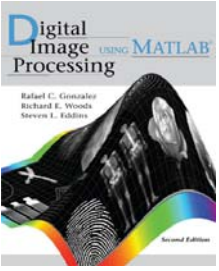
Chapter 3
Intensity Transformations and Spatial Filtering

z_1	z_2	z_3	d_1	d_2	d_3
z_4	z_5	z_6	d_4	0	d_6
z_7	z_8	z_9	d_7	d_8	d_9

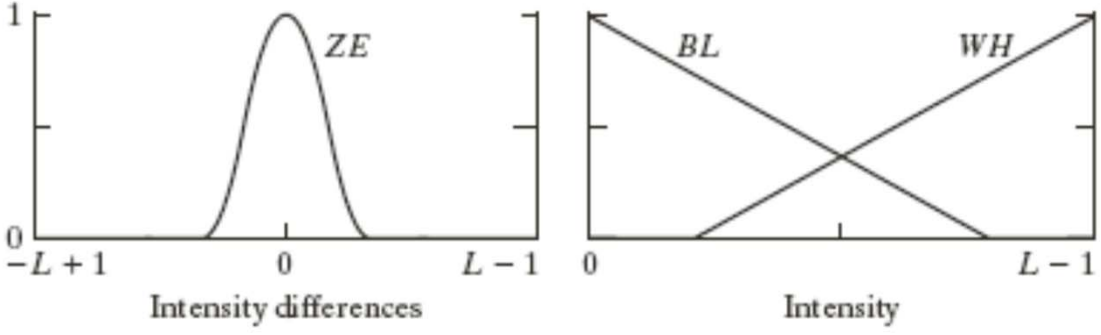
Pixel neighborhood Intensity differences

a b

FIGURE 2.30 (a) A 3×3 pixel neighborhood, and (b) corresponding intensity differences between the center pixels and its neighbors. Only d_2, d_4, d_6, and d_8 are used here to simplify the discussion.



Chapter 3
Intensity Transformations and Spatial Filtering

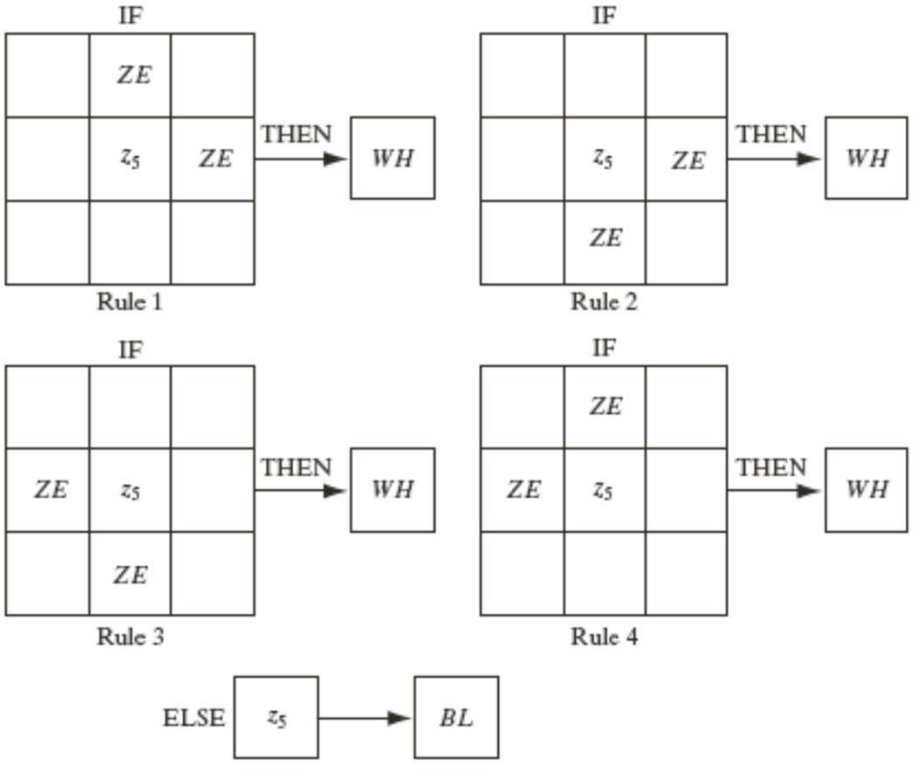


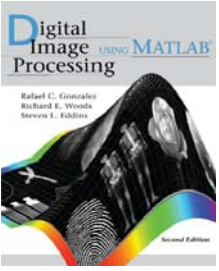
a b

FIGURE 2.31 Membership function of the fuzzy set *zero* (*ZE*). (b) Membership functions of the fuzzy sets *black* (*BL*) and *white* (*WH*).

Chapter 3
Intensity Transformations and Spatial Filtering

FIGURE 2.32 Fuzzy rules for boundary detection.

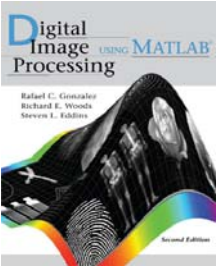




Chapter 3
Intensity Transformations and Spatial Filtering

a b c

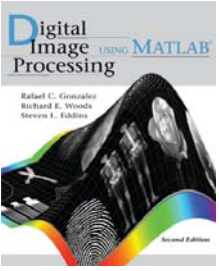
FIGURE 2.33 (a) CT scan of a human head. (b) Result of fuzzy spatial filtering using the membership functions in Fig. 2.31 and the rules in Fig. 2.32. (c) Result after intensity scaling. The thin black picture borders in (b) and (c) were added for clarity; they are not part of the data. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)



Chapter 3
Intensity Transformations and Spatial Filtering

```
clear all
im=imread('lena.jpg');
i=0;
while i<8;
    t=double(rgb2gray(im));
    yes=mod(floor(t/2^i),2);
    figure,imshow(yes);
    i=i+1;
end;
```

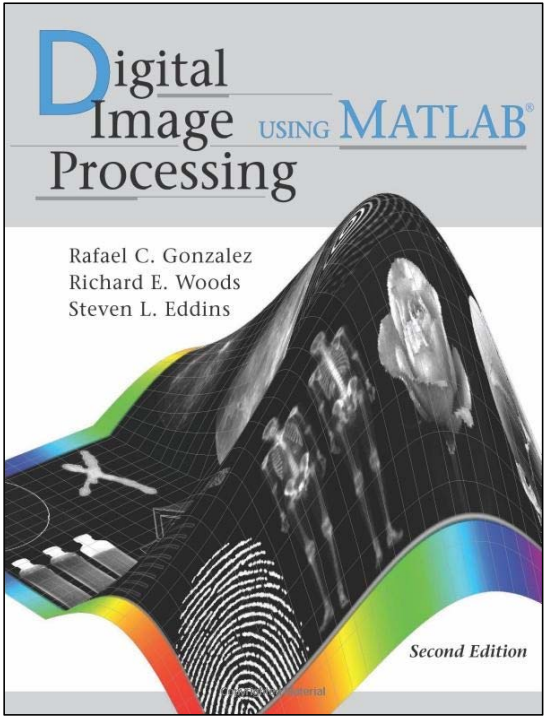
```
Lena=double(imread('Lena.jpg'))+1;
LenaR=Lena(:,:,1);
LenaG=Lena(:,:,2);
LenaB=Lena(:,:,3);
Y=0.299*LenaR+0.587*LenaG+0.114*LenaB;
pic1=mod(floor(Y/1),2);
pic2=mod(floor(Y/2),2);
pic3=mod(floor(Y/4),2);
pic4=mod(floor(Y/8),2);
pic5=mod(floor(Y/16),2);
pic6=mod(floor(Y/32),2);
pic7=mod(floor(Y/64),2);
pic8=mod(floor(Y/128),2);
```



Chapter 3
Intensity Transformations and Spatial Filtering

```
M = 500:500:20000; t = zeros(1,40);
for k = 1:numel(M)
    %f = @() sinfun1(M(k));
    t0 = cputime;
    sinfun1(M(k));
    t(k) = cputime - t0;
end
plot(M, t);
end
function y = sinfun1(M)
x = 0:M-1;
for k = 1:numel(x)
    y(k) = sin(x(k)/(100*pi));
end
```

منبع اصلی



Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins,
Digital Image Processing Using MATLAB®,
Second Edition, Pearson Prentice Hall, 2008.
Chapter 3 (3-4 .. 3-6)