



دانشگاه آزاد اسلامی واحد دشتستان
گروه کامپیوتر

پردازش تصاویر دیجیتال

1395

منبع :

کتاب پردازش تصویر رقمی

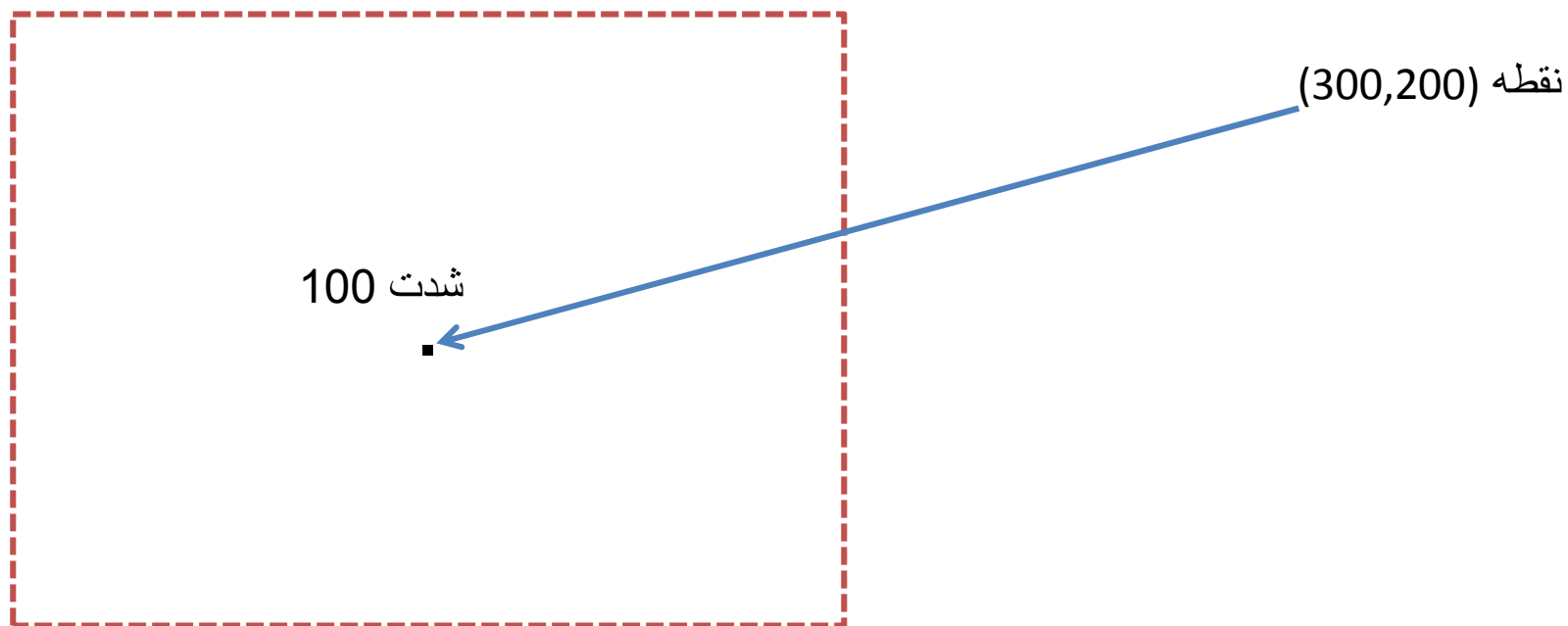
تالیف گنزالس

کاربردها :

- پزشکی : مانند دستگاه MRI
- صنعتی : مانند تشخیص شکستگی قطعات صنعتی
- عکاسی : مانند بازیابی تصاویر قدیمی و ترمیم عکس
- امنیتی : مانند تشخیص اثر انگشت یا تشخیص چهره
- نظامی : تعقیب هدف موشک یا رادار
- حمل و نقل : شمارش تعداد خودروها در یک بزرگراه

نمایش تصویر تک رنگ

تعریف: تابعی دو بعدی مانند $f(x,y)$ است که x و y مختصات مکانی تصویر را مشخص می کند و مقدار این تابع در هر نقطه شدت تصویر را در آن نقطه نشان می دهد



تصویر تک رنگ رقمی

تصویر تک رنگ را رقمی گوئیم هرگاه هم در مختصات مکانی و هم در تعداد سطوح خاکستری رقمی باشد 

به رقمی کردن مختصات مکانی تصویر نمونه برداری یا Sampling گفته می شود 

به رقمی کردن تعداد سطوح خاکستری چندی کردن تصویر یا Quantization گفته میشود 

Sampling و Quantization

هر چه Sampling تصویر بیشتر باشد یعنی جزئیات بیشتری را از تصویر می توانیم نشان دهیم 

Sampling و Quantization باعث میشوند نه در نهایت یک تصویر تک رنگ رقمی بصورت یک ماتریس دو بعدی درآید که هر عنصر این ماتریس یک پیکسل نامیده میشود 

بالا بردن Sampling و Quantization کیفیت تصویر را بالا می برد اما از طرفی هزینه حافظه و زمان پردازش تصویر را بیشتر می کند (برای ذخیره کردن تصویری با ماتریس $M*N$ و با L سطح خاکستری به $M*N*\log_2 L$ فضا نیاز داریم) 

اثر تکرار پیکسل

اگر بخواهیم یک تصویر با ابعاد ماتریس کوچک را در یک صفحه نمایش بزرگ نمایش دهیم باید برخی از پیکسل ها را چند بار تکرار کنیم (برای مثال اگر صفحه نمایش $1000 * 1000$ و تصویر $500 * 500$ باشد هر پیکسل باید ۴ بار تکرار شود)

هر چقدر ابعاد تصویر کوچکتر باشد برای نمایش در همان صفحه نمایش یک پیکسل باید بیشتر تکرار شود و وقتی تعداد تکرار از حدی بگذرد کیفیت تصویر به شدت کاهش پیدا می کند و برای چشم نیز قابل رویت است و باعث می شود تصویر به صورت شطرنجی دیده شود به این اثر تکرار پیکسل گفته می شود

اثر تعداد سطوح خاکستری

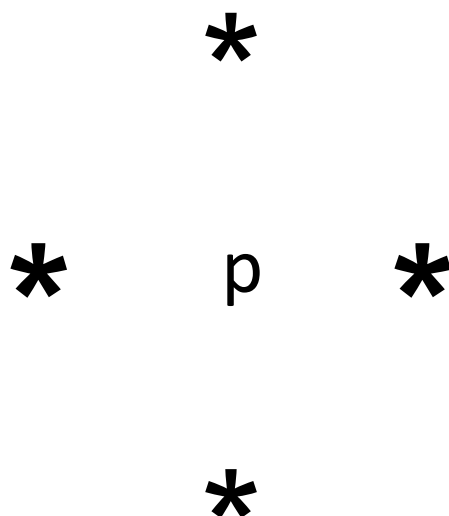
اگر تعداد سطوح خاکستری را پایین بیاوریم در نواحی آرام تصویر جزئیات تصویر را از دست می دهیم یعنی دو پیکسل که دارای دو سطح متفاوت و شیء نزدیک به هم دارند با یک سطح (یک شدت) بیان می شوند یعنی قسمتی از جزئیات از بین رفته است به این اثر اثر منحنی سازی اشتباه یا false contouring گفته می شود

رابطه بین پیکسل ها

همسایگی پیکسل ها

همسایگی چهارگانه

برای پیکسل p با مختصات $p(i,j)$ چهار همسایه وجود دارد که بنام همسایگی چهارگانه شناخته می شوند و با $N_4(p)$ نمایش داده می شوند . این پیکسل ها عبارتند از سطر و ستون p

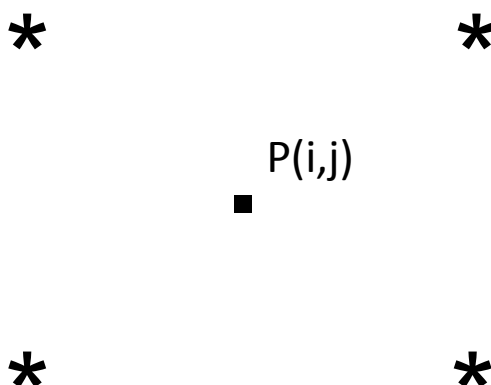


این مختصات ها عبارتند از : $(i+1,j)$, $(i,j+1)$, $(i,j-1)$, $(i-1,j)$

همسایگی قطری

برای پیکسل p با مختصات $p(i,j)$ چهار همسایه وجود دارد که بنام همسایگی قطری یا $N_D(p)$ گفته می شود.

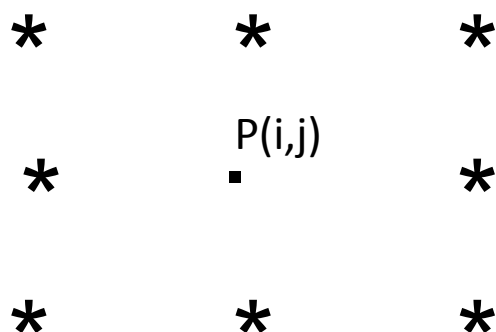
این پیکسل ها عبارتند از :



این مختصات عبارتند از : $(i-1, j-1)$, $(i-1, j+1)$, $(i+1, j-1)$, $(i+1, j+1)$

همسایگی هشتگانه

به مجموعه همسایگی های چهارگانه و قطری هشتگانه گفته می شود و با $N_8(p)$ نمایش داده می شود



این مختصات عبارتند از : $(i+1, j)$, $(i-1, j-1)$, $(i-1, j+1)$, $(i+1, j-1)$, $(i+1, j+1)$, $(i, j+1)$, $(i, j-1)$, $(i-1, j)$

اتصال پیکسل ها

اتصال چهارگانه

فرض کنید مجموعه S شامل چند سطح خاکستری (شدت) باشد پیکسل q با پیکسل p در اتصال چهارگانه است اگر:

1. همسایگی چهارگانه p باشد

2. شدت p و شدت q هر دو عضو مجموعه S باشد

اتصال چهارگانه

$$S=\{1\}$$

مثال :

m	\	q	\	n	•
x	•	p	\	k	\
z	\	y	•	t	•

k و q هر دو همسایگی چهارگانه p هستند و شدت آنها در مجموعه S است پس متصل چهارگانه p هستند

X و y همسایگی چهارگانه p هستند اما شدت هایشان عضو مجموعه S نیست پس متصل چهارگانه p نیستند

اتصال قطری

پیکسل p و q با هم در اتصال قطری هستند اگر :

1. همسایگی قطری هم باشند

2. شدت p و شدت q عضو مجموعه S باشند

اتصال مرکب یا اتصال m گانه

می گوئیم پیکسل p و پیکسل q متصل m گانه هستند اگر:

❖ q در $N_4(p)$ باشد.

❖ q در $N_D(p)$ باشد و مجموعه $N_4(p) \cap N_4(q)$ فاقد پیکسلی باشد که مقادیر آنها در S باشد.

اتصال مرکب یا اتصال m گانه

مثال : مشخص کنید که کدام پیکسل ها متصل m گانه p هستند

$$S = \{1\}$$

0^m	1^n	1^k
0^g	1^p	0^r
0^w	0^t	1^u

متصل های m گانه p : u

رزولیشن و پیکسل

🌹 پیکسل : به هر عنصر ماتریس تصویر یک پیکسل گویند.

🌹 (Resolution) : تعداد پیکسل هایی که یک واحد از صحنه را نمایش دهد

فاصله دو پیکسل

تابع $D(p,q)$ را فاصله بین دو پیکسل p و q می گوئیم اگر این تابع دارای شرایط زیر باشد :

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

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

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

فواصل مختلفی بین دو پیکسل تعریف می شود

فاصله اقلیدسی

فرض کنید دو پیکسل $p(x,y)$ و $q(s,t)$ وجود داشته باشد ، فاصله اقلیدسی بین p و q بصورت زیر تعریف می شود :

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

تمام پیکسل هایی که از پیکسل p دارای فاصله اقلیدسی یکسان هستند تشکیل دایره می دهند

فاصله چهارگانه

فاصله چهارگانه بین p و q بصورت زیر تعریف می شود :

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

تمام پیکسل هایی که از پیکسل p دارای فاصله یکسان هستند، تشکیل لوزی می دهند

فاصله 8 گانه

فاصله 8 گانه بین p و q بصورت زیر تعریف می شود :

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

تمام پیکسل هایی که از پیکسل p دارای فاصله یکسان هستند، تشکیل مربع می دهند

تبدیلات تصویر

تبدیلات تصویر

تبدیل های تصویر که به صورت پیش پردازش انجام می شوند به منظور بهبود کیفیت تصویر یا ارتقاء تصویر انجام می شود

هدف از این کار آماده سازی تصویر برای عملیات اصلی یا کارهای بعدی می باشد

اگر پردازش را به صورت کلی و روی کلیه پیکسل های یک تصویر باشد آن را تبدیل یک تصویر گوئیم

اگر پردازش به صورت محلی باشد و عملیات بر روی قسمت خاصی از پیکسل ها انجام شود آن را فیلتر کردن تصویر می گویند

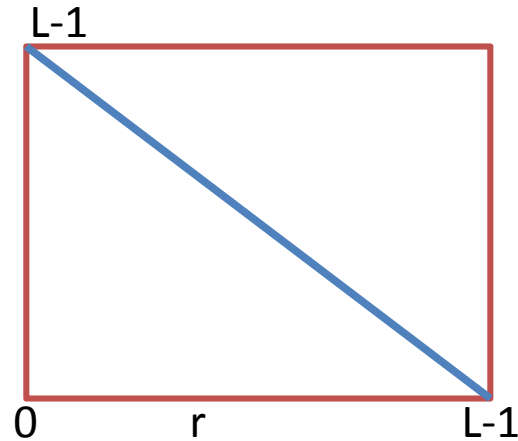
بطور کلی هر تصویر بصورت زیر می باشد



تبدیل منفی سازی تصویر (negative)

اگر شدت هر پیکسل را از 255 کم کنیم شدت منفی شده آن پیکسل بدست می آید

$$T(r) = (L-1) - r$$



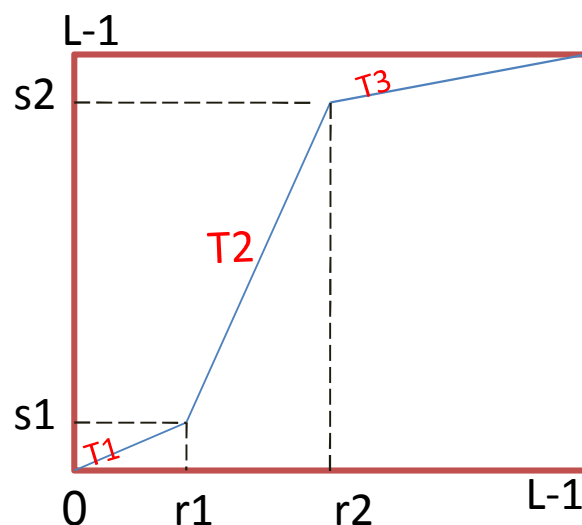
for i=1 to 256

for j=1 to 256

$$S(i,j) = 255 - f(i,j)$$

تبدیل گشش تمایز (contrast stretching)

هدف از این تبدیل زیاد کردن اختلاف بیشترین شدت با کمترین شدت موجود در تصویر می باشد تا بهتر دیده شود

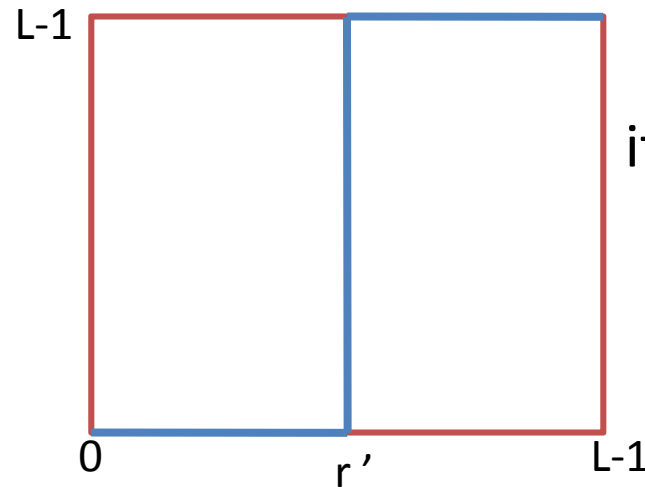


```
if r < r1 .  
    S(i,j) = T1(p(i,j)) .  
else .  
    if r1 < r < r2 .  
        S(i,j) = T2(p(i,j)) .  
    else .  
        S(i,j) = T3( p(i,j) ) .
```

باینری کردن

حالت خاصی از تبدیل کشش تمایز است که در آن :

$$S1=0, S2=L-1, r1=r2$$



if ($p(i,j) < r'$)

$S(i,j) = 0$

else

$S(i,j) = L-1$

.
. .
. .
. .

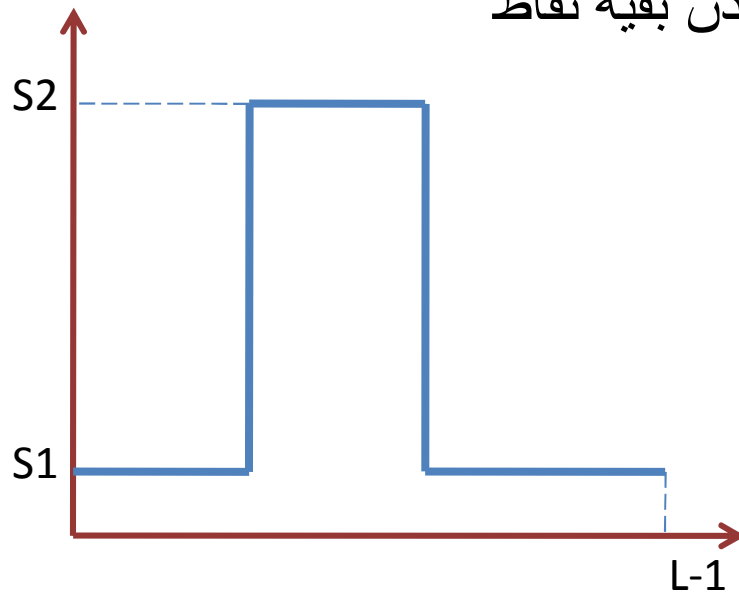
تبدیل بخش بندی سطح خاکستری

Gray Level Slicing

هدف از این تبدیل برجسته کردن محدوده خاصی از سطوح خاکستری موجود در تصویر است. به عنوان مثال اگر بخواهیم بدانیم که چند درصد از مساحت یک کشور را آب تشکیل داده است می توان این تبدیل را بر روی تصویر ماهواره ای گرفته شده از این کشور اعمال کرد. این تبدیل دو نوع است:

1- تقویت نقاط مورد نظر و تضعیف بقیه نقاط

2- تقویت نقاط مورد نظر و بدون تغییر ماندن بقیه نقاط



if $r1 < r < r2$

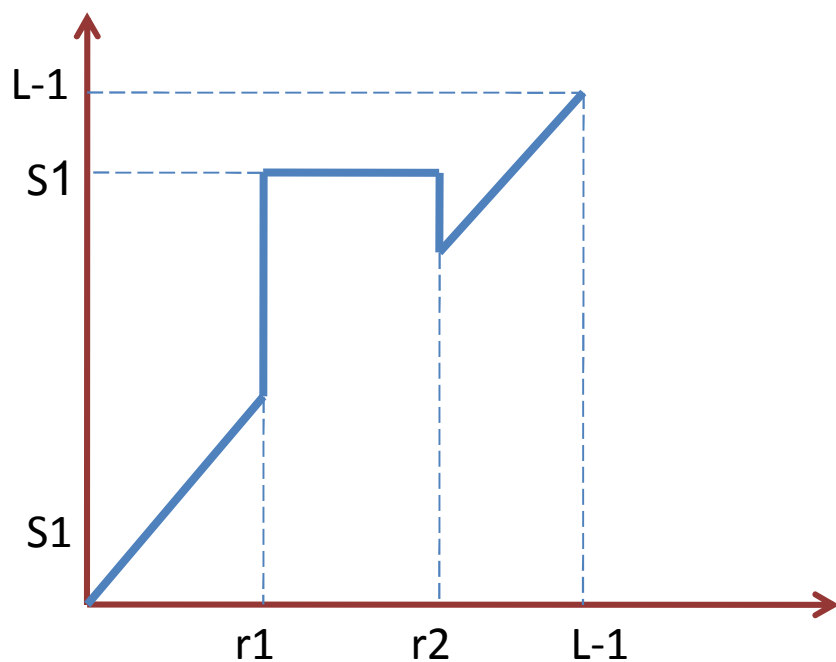
$S(i,j) = S2$

else

$S(i,j) = S1$

تبدیل بخش بندی سطح خاکستری

Gray Level Slicing



if $r_1 < r < r_2$
 $S(i,j) = S_1$

- تعديل هيستوگرام
- MATLAB: در

```
f=imread('pout.tif');  
imhist(f);  
g=histeq(f);  
imshow(g);
```

تفریق دو تصویر

اگر دو تصویر هم اندازه را به صورت پیکسل به پیکسل از هم کم کنیم این تابع را تفریق دو تصویر گوئیم .

کاربردهای تفریق دو تصویر :

تشخیص تغییرات ایجاد شده در محیط

مثل شمردن تعداد ماشینها در خیابان

تشخیص مسیر حرکت دارو در رگهای بدن و . . .

جمع دو تصویر

- می توان دو یا چند تصویر را بصورت پیکسل به پیکسل با هم جمع کرد
- از جمع به تنهایی استفاده نمی شود و بیشتر برای بدست آوردن میانگین چند تصویر استفاده می شود
- کاربرد : استخراج پس زمینه تصویر
- برای مثال برای بدست آوردن تصویر چهارراه خالی از ماشین در مسأله شمارش تعداد ماشینهای چهارراه

فیلترهای تصویر

فیلترهای تصویر

- ❑ فیلترهای تصویر به دو دسته کلی تقسیم می شوند: فیلترهای حوزه مکان و فیلترهای حوزه فرکانس
- ❑ فیلترهای حوزه مکان مستقیماً روی پیکسل های تصویر اعمال می شود
- ❑ فیلترهای حوزه فرکانس بر روی تبدیل فوریه تصویر اعمال می شود
- ❑ فیلترهای تصویر از جهت دیگر نیز به دو دسته تقسیم می شوند: فیلترهای بالاگذر (HPF) و فیلترهای پایین گذر (LPF)
- ❑ فیلترهای بالاگذر فیلترهایی هستند که فرکانس های پایین را عبور می دهند و فرکانس های بالا را حذف می کند
- ❑ اما فیلترهای پایین گذر فرکانس های بالا را عبور می دهند و فرکانس های پایین را حذف می کند

فیلترهای آرام کننده و فیلترهای تشخیص لبه

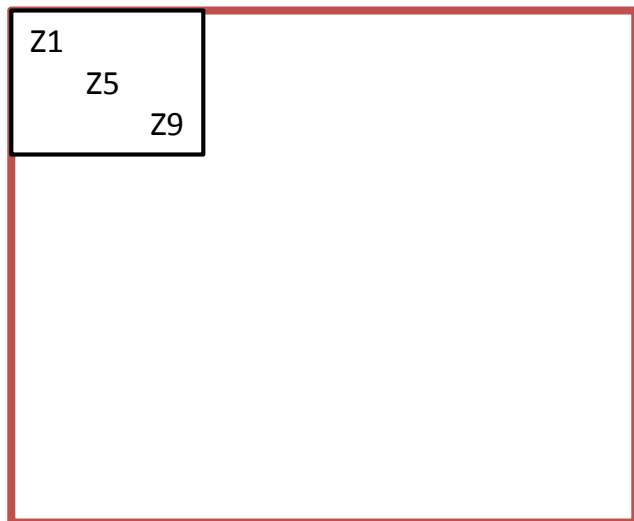
- ❑ به جاهایی که اختلاف شدت بالا است لبه می گویند، فیلترهای پایین گذر لبه ها را حذف می کنند، این فیلترها را فیلترهای آرام کننده می نامند
- ❑ فیلترهای بالا گذر لبه ها را تشدید می کنند، این فیلترها را فیلترهای تشخیص لبه می نامند

فیلترهای پایین گذر

فیلتر میانگین گیری معمولی

معمولا mask بصورت 3 در 3 تعریف می شود چون اگر یک پیکسل وسط را در نظر بگیریم تمام همسایه هایش را نشان می دهد

w1		
	w5	
		w9



$$z5 = \frac{\sum w_i z_i}{\sum w_i}$$

$$z5 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

میانگین گیری معمولی

فیلتر میانه گیری (Median)

فیلتر خطی که برای تشخیص نویز به کار می رود 

اول بصورت صعودی یا نزولی مرتب می کنیم و عنصر وسط را به عنوان مقدار جدید پیکسل قرار می دهیم 

به مثال زیر توجه کنید: 

12 , 34 , 33 , 190 , 230 , 125 , 1 , 48 , 53



$Z_5 = 48$

فیلتر میانه گیری در MATLAB

```
I = imread('eight.tif');  
figure, imshow(I)  
J = imnoise(I,'salt & pepper',0.02);  
K = medfilt2(J);  
imshowpair(J,K,'montage')
```

فیلتر گاوس

تأثیر همسایه های چهارگانه بیشتر از همسایه های قطری است

$$\frac{1}{16} * \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

فیلترهای بالا گذر

فیلتر prewitt

1	0	-1
1	0	-1
1	0	-1

m

1	1	1
0	0	0
-1	-1	-1

n

$$P = |m| + |n|$$

109	112	113	108	106	115	108	105									
106	111	116	110	111	107	112	113									
118	113	112	110	113	115	112	109									
110	111	114	117	110	115	107	110									
190	191	179	180	191	185	183	181									
178	180	185	187	190	188	185	183									
190	191	179	182	191	190	187	186									
183	179	184	189	194	189	188	183									

 $\otimes$

-1	-1	-1
0	0	0
1	1	1

 $=$

9	2	8	9	11	8
2	5	4	14	2	0
217	215	215	218	219	213
208	210	221	223	231	224
0	2	2	7	9	14
3	0	5	7	8	4

فیلتر sobel

این فیلتر معروفترین فیلتر بالا گذر است

1	0	-1
2	0	-2
1	0	-1

m

1	2	1
0	0	0
-1	-2	-1

n

$$P = |m| + |n|$$

در Matlab:

```
p=imread('rice.tif');  
re=edge(p,'sobel');  
imshow(re)
```

فیلتر frei-chan

1	0	-1
$\sqrt{2}$	0	$-\sqrt{2}$
1	0	-1

m

1	$\sqrt{2}$	1
0	0	0
-1	$-\sqrt{2}$	-1

n

$$P = |m| + |n|$$

فیلتر لاپلاس

اختلاف پیکسل وسط با همسایه ها محاسبه می شود و لبه ها مشخص می شود

0	-1	0
-1	4	-1
0	-1	0

فیلترها در Matlab

- از دستور `imfilter` استفاده می کنیم.

- `g=imfilter(f,w)`

`f` تصویر اصلی و `w` ماسک است

- یک راه برای ساخت ماتریس ماسک ، مقدار دهی به صورت دستی است.

- `im = imread('rice.png');`
- `filter = ones(3,3) .* 0.11;`
- `imFilt = imfilter(im,filter);`
- `imshow(im),figure,imshow(imFilt);`

فیلترها در Matlab

راه دوم استفاده از دستور متلب است که از دستور `fspecial` استفاده می کنیم:

`w=fspecial('type',param)`

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

فیلترها در Matlab

استفاده از **fspecial** :

```
f=imread('rice.png');  
w=fspecial('sobel');  
g=imfilter(f,w);
```

فشرده سازی

آنتروپی

✖ متوسط تعداد بیت یا متوسط طول هر سِمبل (مثلا کاراکتر) را آنتروپی گویند

$$H(S) = \sum_i (p_i \log_2 (1/p_i))$$

✖ مثال : فرض کنید $1/3$ از پیکسل های تصویر دارای شدت x باشند و $2/3$ دیگر دارای شدت y باشند، آنتروپی را پیدا کنید

$$H(s) = \frac{1}{3} * \log 3 + \frac{2}{3} * \log \frac{3}{2}$$

الگوریتم های فشرده سازی

الگوریتم shannon-fano ✨

الگوریتم هافمن ✨

هر کدام از الگوریتم های فوق را با یک مثال بیان می کنیم

الگوریتم shannon-fano

مثال

Symbol	A	B	C	D	E
فراوانی	15	7	6	6	5
	0	0	1	1	1
	0	1	0	1	1
				0	1

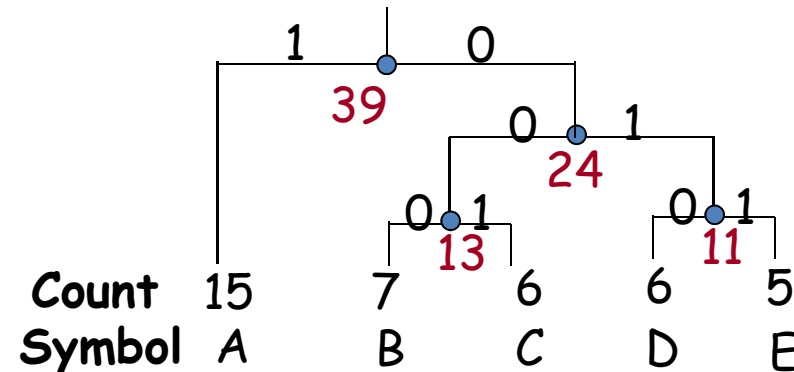
Symbol	Count	Info. $-\log_2(p_i)$	Code	Subtotal # of Bits
A	15	X 1.38	00	30
B	7	X 2.48	01	14
C	6	X 2.70	10	12
D	6	X 2.70	110	18
E	5	X 2.96	111	15
		85.25		89

- برای یک لیست داده شده از نمادها، یک لیست متناظر از احتمال‌ها یا فراوانی وقوع بساز به طوری که فراوانی نسبی وقوع هر نماد معلوم باشد.
- لیست نمادها را بر اساس فراوانیشان مرتب کن، به طوری که بیشترین وقوع نمادها از نظر فراوانی در سمت چپ و کمترین آن در سمت راست.
- لیست را به دو بخش تقسیم کن، به طوری که جمع فراوانی‌های قسمت چپ تا حد ممکن نزدیک به جمع فراوانی‌های قسمت راست باشد.
- برای قسمت چپ لیست رقم دودویی ۰ گذاشته می‌شود و برای قسمت راست رقم ۱.
- گام‌های ۳ و ۴ را به صورت بازگشتی برای هر کدام از دو نصف‌ها اعمال کن، گروه‌ها را دوباره تقسیم کن و بیت‌ها را به رمزها اضافه کن تا هر نماد به یک برگ رمز در درخت متناظر شود.

الگوریتم هافمن

- به ازای هر نماد برگی ایجاد کن و آن را به تعداد اتفاق افتادن آن اضافه کن
- تا وقتی بیش از یک گره در صف موجود است مراحل زیر را انجام بده:
 - دو گره که کمترین فراوانی را دارند از صف حذف کن.
 - ۰ و ۱ را به ترتیب به ابتدای کدی که قبلاً با این گره‌ها نسبت داده شده بود اضافه می‌کنیم.
 - یک گره داخلی جدید ایجاد می‌کنیم که این دو گره فرزندان آن باشند و احتمال آن را نیز برابر جمع احتمال این دو گره قرار می‌دهیم.
 - این گره جدید را به صف اضافه می‌کنیم.
- گره باقی‌مانده ریشه است و درخت کامل شده است.

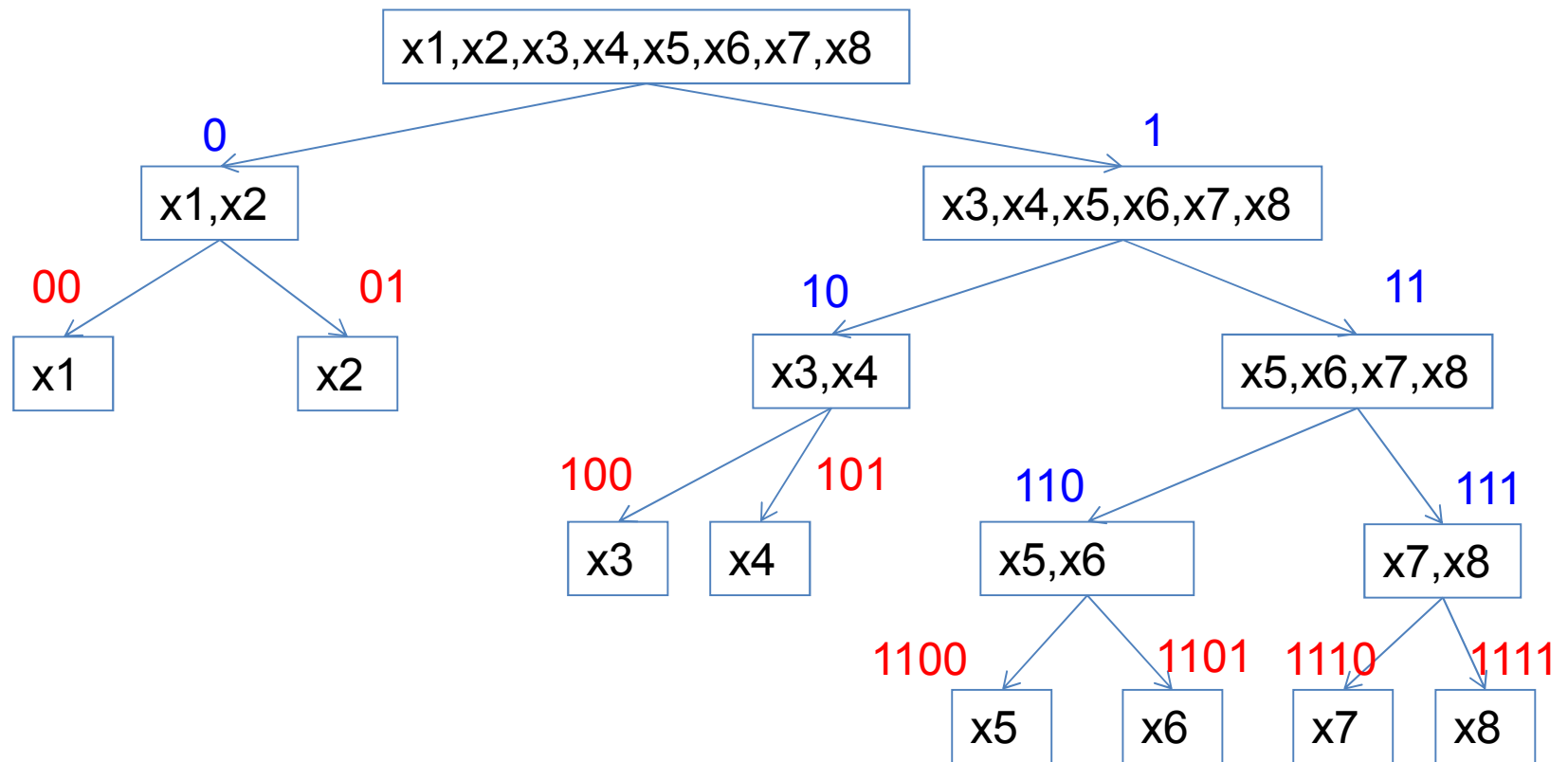
Example



Symbol	Count	Info. $-\log_2(p_i)$	Code	Subtotal # of Bits
A	15	X 1.38	1	15
B	7	X 2.48	000	21
C	6	X 2.70	001	18
D	6	X 2.70	010	18
E	5	X 2.96	011	15
		85.25		87

Shannon-Fano Encoding: Example

Message	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Probability	0.25	0.25	0.125	0.125	0.0625	0.0625	0.0625	0.0625



Shannon-Fano Encoding: Example

Message	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Probability	0.25	0.25	0.125	0.125	0.0625	0.0625	0.0625	0.0625
Encoding vector	00	01	100	101	1100	1101	1110	1111

- Entropy $H = -\left(2 \cdot \left(\frac{1}{4} \log \frac{1}{4}\right) + 2 \cdot \left(\frac{1}{8} \log \frac{1}{8}\right) + 4 \cdot \left(\frac{1}{16} \log \frac{1}{16}\right)\right) = 2.75$
- Average length of the encoding vector

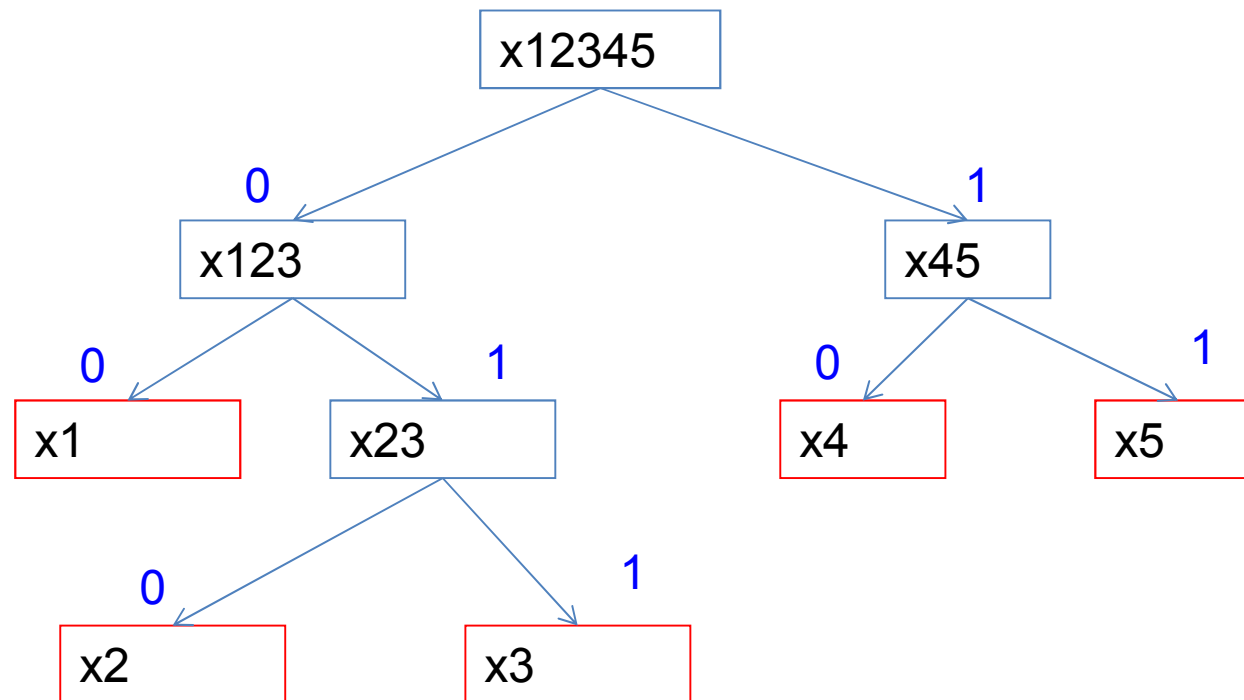
$$\bar{L} = \sum P\{x_i\} n_i = \left(2 \cdot \left(\frac{1}{4} \cdot 2\right) + 2 \cdot \left(\frac{1}{8} \cdot 3\right) + 4 \cdot \left(\frac{1}{16} \cdot 4\right)\right) = 2.75$$

- The Shannon-Fano code gives 100% efficiency

Huffman Encoding: Example

- Let us construct the Huffman code for the following set of messages: x_1, x_2, x_3, x_4, x_5 with the probabilities $p(x_1)=\dots=p(x_5)=0.2$
- 1) x_1 ($p=0.2$), x_2 ($p=0.2$), x_3 ($p=0.2$), x_4 ($p=0.2$), x_5 ($p=0.2$)
- 2) $x_4, x_5 \rightarrow x_{45}$ ($p=0.4$) $\Rightarrow x_{45}, x_1, x_2, x_3$
- 3) $x_2, x_3 \rightarrow x_{23}$ ($p=0.4$) $\Rightarrow x_{45}, x_{23}, x_1$
- 4) $x_1, x_{23} \rightarrow x_{123}$ ($p=0.6$) $\Rightarrow x_{123}, x_{45}$
- 5) $x_{123}, x_{45} \rightarrow x_{12345}$ ($p=1$)

Huffman Encoding: Example



Encoding vectors: $x1 \rightarrow (00)$; $x2 \rightarrow (010)$; $x3 \rightarrow (011)$; $x4 \rightarrow (10)$; $x5 \rightarrow (11)$

Huffman Encoding: Example

- Entropy $H(X) = -5(0.2 \log 0.2) = -5 \left(\frac{1}{5} \log \frac{1}{5} \right) = -\log \frac{1}{5} = \log 5 \approx 2.32$
- Average length of the encoding vector

$$\bar{L} = 3 \cdot \left(\frac{1}{5} \cdot 2 \right) + 2 \left(\frac{1}{5} \cdot 3 \right) = \frac{12}{5} = 2.4$$

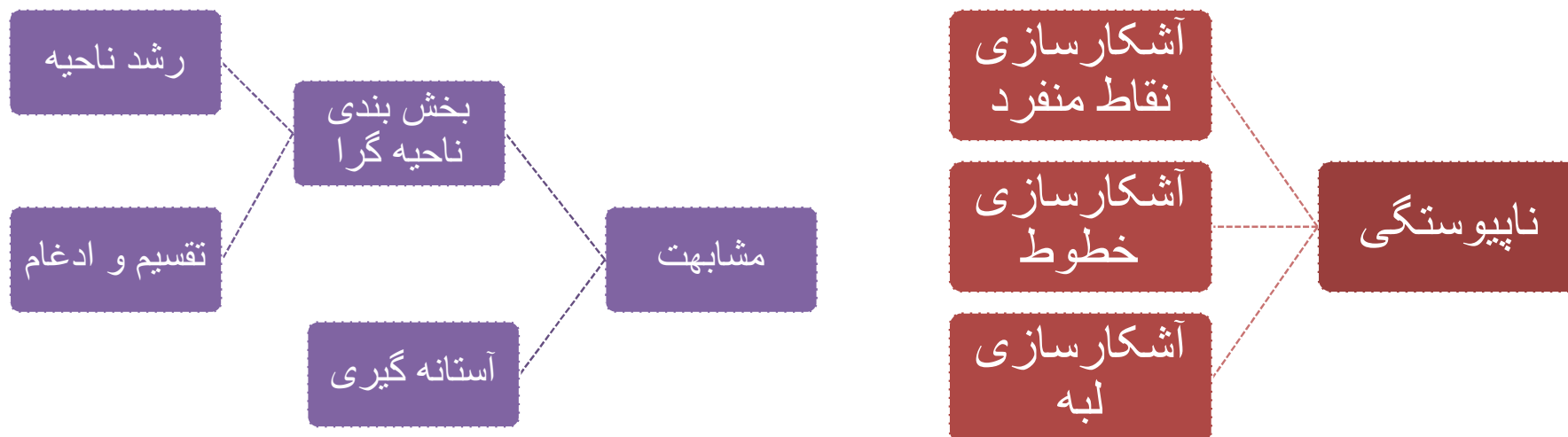
- The Huffman code gives $(2.32/2.4)100\% = 97\%$ efficiency

بخش بندی تصویر

بخش بندی تصویر

تصویر را به قسمت های تشکیل دهنده تقسیم می کنیم، مثلا برای شمردن تعداد اتمبیل ها نیاز به بخش بندی تصویر به بخش های مختلف است

بخش بندی دو معیار دارد : ناپیوستگی و مشابهت



آشکار سازی نقاط منفرد

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

W1	W2	W3
W4	W5	W6
W7	W8	W9

$$R = \sum_{i=1}^9 Z_i W_i$$

اگر $|R| > T$ باشد یعنی ناپیوستگی وجود دارد وگرنه ناپیوستگی وجود ندارد 
مقدار T یک مقدار از قبل تعیین شده است 

آشکار سازی نقاط منفرد

Mask زیر یک Mask خوب برای بدست آوردن پیوستگی است

-1	-1	-1
-1	8	-1
-1	-1	-1

آشکار سازی خطوط

☀ از چهار mask برای تشخیص خطهای افقی، عمودی و مورب استفاده می کنیم

R1	2	2	2		2	0	2	R2
	0	0	0		2	0	2	
	2	2	2		2	0	2	
R3	0	2	2		2	2	0	R4
	2	0	2		2	0	2	
	2	2	0		0	2	2	

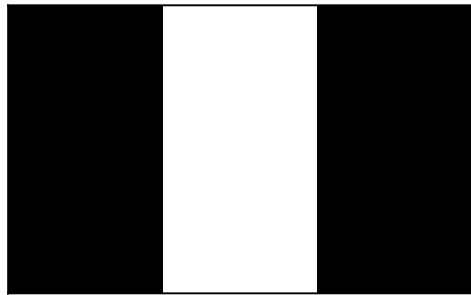
☀ اگر R1 از بقیه Rها بزرگتر باشد خط افقی وجود دارد

☀ اگر R2 از بقیه Rها بزرگتر باشد خط عمودی وجود دارد

☀ اگر R3 از بقیه Rها بزرگتر باشد خط مورب (نیمساز ربع 2 و 4) وجود دارد

☀ اگر R4 از بقیه Rها بزرگتر باشد خط مورب (نیمساز ربع 1 و 3) وجود دارد

آشکار سازی لبه



اگر از محیط تیره به محیط روشن برویم مشتق اول مثبت و اگر از محیط روشن به محیط تیره برویم مشتق اول منفی است



در هر نقطه که مشتق دوم از صفر عبور کند حتما لبه وجود دارد

مشتق دوم همیشه در قسمت های تیره مثبت و در قسمت های روشن منفی است

آشکارسازی مرز

اگر لبه ها را پیدا کرده و آنها را به هم وصل کنیم مرز نامیده می شود 

برای آشکارسازی مرز دو نوع پردازش وجود دارد : پردازش محلی و پردازش سراسری 

پردازش محلی

نقاط (x,y) را در نظر میگیریم در هرکدام از این نقاط اندازه گرادیان و جهت گرادیان و زاویه گرادیان را محاسبه می کنیم و برای هر حالت یک حد آستانه در نظر می گیریم ، اگر اختلاف اندازه گرادیان دو نقطه شرایط مورد نظر را برآورده کند، آن دو نقطه را به هم وصل می کنیم، برای مثال:

$$| \nabla f(x,y) - \nabla f(x',y') | < E$$

$$| \alpha f(x,y) - \alpha f(x,y) | < A$$

اگر هر دو شرط برقرار باشد می توانیم به هم وصل کنیم

پردازش سراسری

با استفاده از نظریه گراف انجام می شود در این حالت اختلاف شدت دو پیکسل همسایه را می توان معیاری برای گراف در نظر گرفت ، اگر H بالاترین شدت موجود در تصویر باشد و $f(p)$ مقدار شدت پیکسل p باشد، برای هر دو پیکسل p, q که با هم همسایه هستند فرمول زیر را برای یک معیار محاسبه می کنیم:

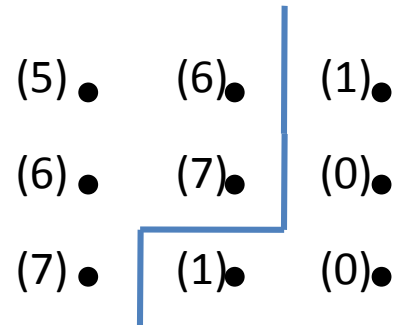
$$H - |f(p) - f(q)|$$

در هر قسمت از گراف این مقدار برای تمام زوج پیکسل های همسایه محاسبه می شود و هر کدام که دارای مقدار کمتری باشند یعنی معیار مورد نظر را برآورده کرده است و آنرا متصل می کنیم

پردازش سراسری

مثال:

$H=7$



آستانه گیری

آستانه گیری یکی از روشهای بخش بندی است که معمولاً بر اساس هیستوگرام تصویر مشخص می شود

فرض کنید در یک تصویر یک شیء روشن در یک زمینه تاریک قرار دارد، در این حالت مرز بین دو ناحیه را بصورت یک آستانه در نظر می گیریم در حالت کلی آستانه گیری بصورت تابع زیر تعریف می شود:

$$T [x , y , p(x,y) , f(x,y)]$$

⊗ x و y مختصات مکانی پیکسل p می باشد

⊗ $f(x,y)$ شدت در نقطه p می باشد

⊗ $P(x,y)$ یک خاصیت محلی است

آستانه گیری

اگر آستانه گیری بر اساس شدت پیکسل باشد آستانه گیری را سراسری یا ساده می گوئیم

اگر $p(x,y)$ نیز دخالت داده شود آستانه گیری محلی نامیده می شود

اگر مختصات مکانی هم تاثیر داده شود آستانه گیری دینامیک گفته می شود

بخش بندی ناحیه گرا

در روش بخش بندی ناحیه گرا معمولاً تصویر به n ناحیه تقسیم می شود 

برای بخش بندی ناحیه گرا دو الگوریتم وجود دارد : الگوریتم رشد ناحیه و  الگوریتم تقسیم و ادغام

الگوریتم رشد ناحیه

در این روش می توان یک پیکسل اولیه با یک سطح مشخص را به عنوان بذر انتخاب کرد و سپس از بین پیکسل های فعلی، پیکسل هایی که معیار مشابهت را دارند انتخاب می کنیم

سپس برای هر کدام از اعضای این مجموعه عملیات فوق را تکرار می کنیم، مشخص است که در هر مرحله ناحیه رشد می کند

این کار را تا زمانی انجام می دهیم که هیچ پیکسلی پیدا نشود که معیار مشابهت را داشته باشد

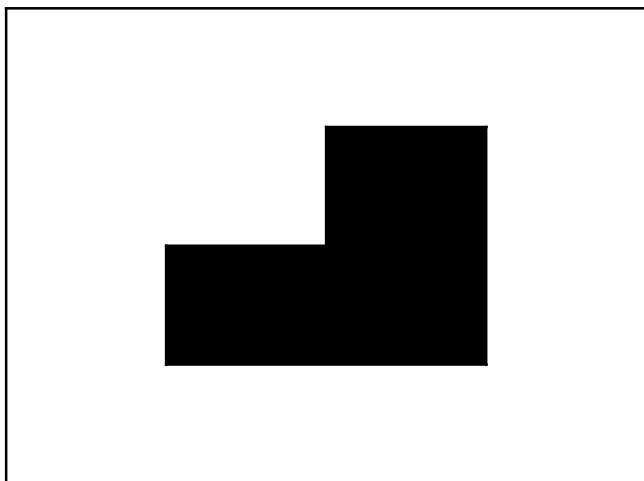
← تعیین نقطه بذر مهمترین مشکل این روش است

الگوریتم تقسیم و ادغام

ابتدا تصویر را به چهار ناحیه تقسیم می کنیم، اگر در هر ناحیه $p(R_i) = \text{True}$  نباشد مجدداً آن ناحیه را به چهار قسمت تقسیم می کنیم
اگر $p(R_i) \wedge p(R_j) = \text{True}$ باشد آن دو ناحیه را ادغام می کنیم

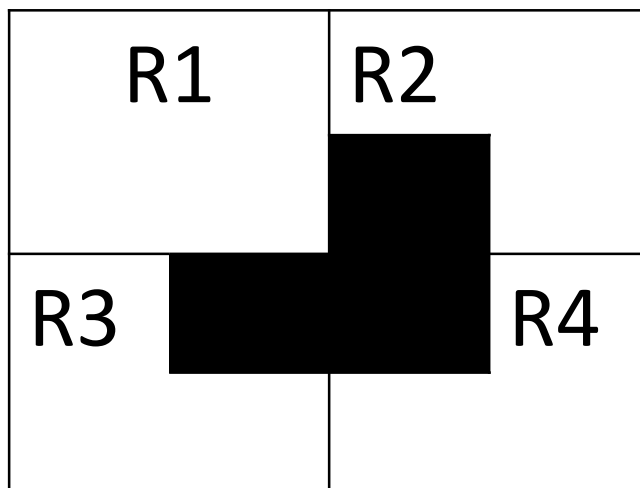
الگوریتم تقسیم و ادغام

مثال : الگوریتم تقسیم و ادغام را بر روی شکل زیر اعمال کنید



الگوریتم تقسیم و ادغام

شکل را به چهار قسمت تقسیم می کنیم :



الگوریتم تقسیم و ادغام

ناحیه R1 نیازی به تقسیم ندارد اما بقیه ناحیه ها تقسیم می شوند :

R1		R ₂₁	R ₂₂
			R ₂₄
R ₃₁			R ₄₂
R ₃₃	R ₃₄	R ₄₃	R ₄₄

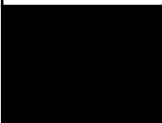
الگوریتم تقسیم و ادغام

ناحیه R_{21} و R_{22} با هم ادغام می شوند :

R1		R_{21} و R_{22}	
			R_{24}
R_{31}			R_{42}
R_{33}	R_{34}	R_{43}	R_{44}

الگوریتم تقسیم و ادغام

ناحیه حاصل از ادغام R_{21} و R_{22} با R_{24} ادغام می شوند :

R1		و R_{21} و R_{22}	
			R_{24}
R_{31}			R_{42}
R_{33}	R_{34}	R_{43}	R_{44}

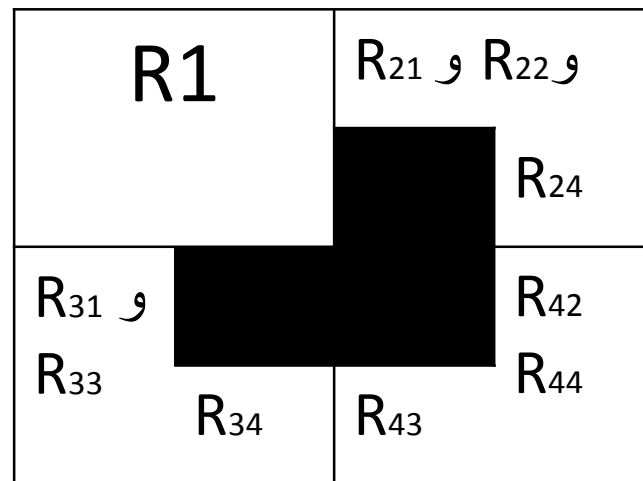
الگوریتم تقسیم و ادغام

ناحیه R_{31} و R_{32} و R_{34} ادغام می شوند :

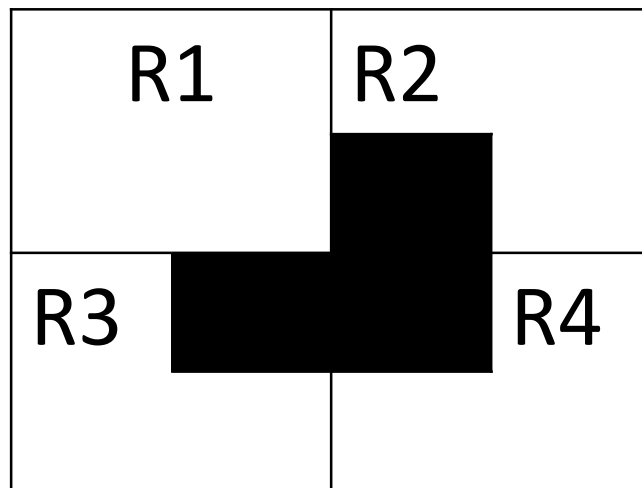
R_1		R_{21} و R_{22}	
			R_{24}
R_{31} و R_{33}			R_{42}
	R_{34}	R_{43}	R_{44}

الگوریتم تقسیم و ادغام

ناحیه R_{42} و R_{43} و R_{44} ادغام می شوند :



الگوریتم تقسیم و ادغام



الگوریتم تقسیم و ادغام

نواحی R1 و R2 و R3 و R4 با هم ادغام می شوند :

