فصل سیزدهم

پردازش تصویر

دکتر بهناز بیگدلی

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱-مقدمه

- در متلب تصاویر بصورت ماتریسهای دو، سه و یا چهاربعدی تعریف می‌شوند.
- کیفیت تصویر: کیفیت تصویر به دو پارامتر یکی دقت ابعادی و دیگری دقت عمقی در هنگام تصویربرداری و یا ذخیره‌سازی تصویر بستگی دارد.
- دقت عمقی(Depth): منظور از دقت عمقی تعداد بیتهایی است که از حافظه کامپیوتر به هر نقطه(پیکسل) از تصویر اختصاص داده می‌شود.
- دقت ابعادی(Resolution): منظور تعداد نقاط نمونه‌برداری شده در واحد طول یا عرض تصویر است. دقت ابعادی افقی و عمودی یک تصویر ممکن است متفاوت باشند اما معمولا چنین نیست. واحد دقت ابعادی dpi یا نقطه بر اینچ است.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱-مقدمه-ادامه-

- انواع تصاویر: انواع تصاویر عبارتند از :
  - تصاویر اندیس‌دار
  - تصاویر شدت
  - تصاویر باینری
  - تصاویر RGB
  - تصاویر چندفریمی

  که در ادامه فصل مفصلا به هریک خواهیم پرداخت
- فرمتهای گرافیکی: تصاویر با فرمتهای مختلفی می‌توانند بر روی دیسک ذخیره شوند. مهمترین فرمتهای گرافیکی در زمان حاضر عبارتند از: PNG، JPG، BMP، TIFF، GIF که تمامی آنها بعلاوه چندین فرمت دیگر توسط متلب پشتیبانی می‌شوند.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۲- انواع تصاویر

تصاویر شدت(Intensity Image)

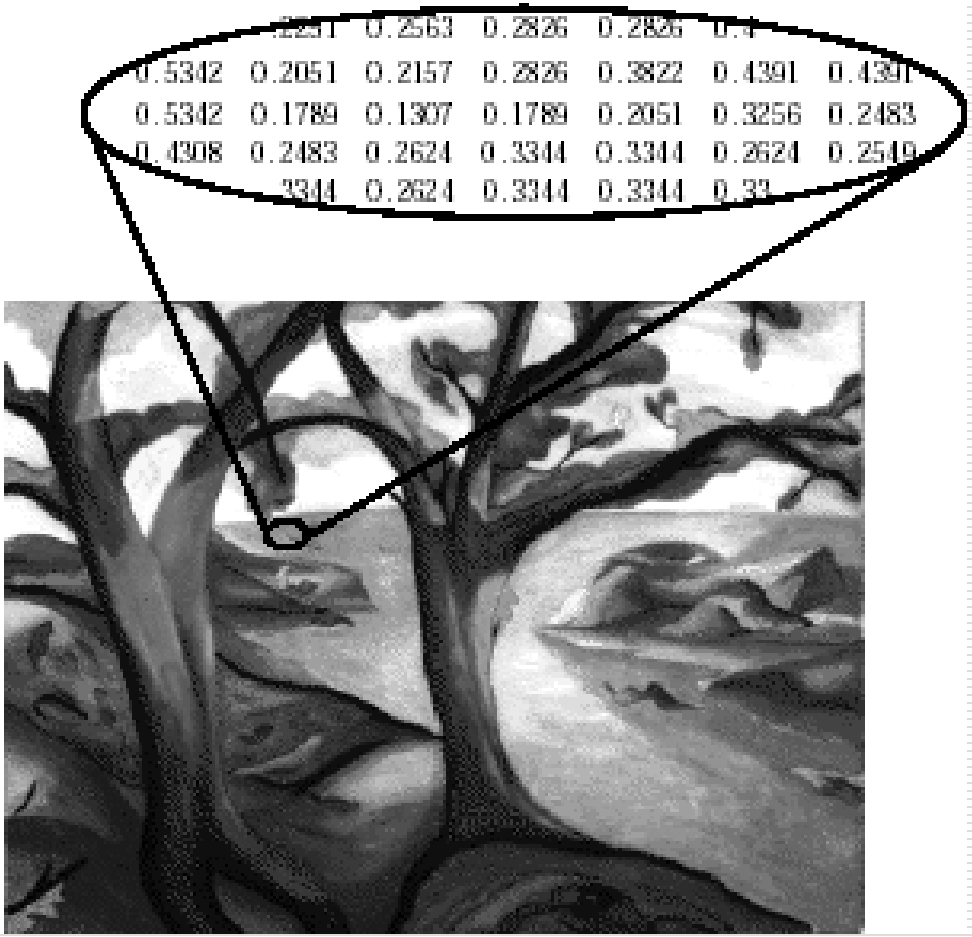
تصویر شدت یا تصویر سطح خاکستری، به تصویری گفته می‌شود که تنها دارای مقادیر روشنایی باشد و فاقد خصوصیات رنگ مانند: فام و خلوص باشد. در متلب این تصاویر توسط ماتریسهای دو بعدی تعریف می‌شوند بطوریکه مقدار هر عنصر از این ماتریس معرف میزان روشنایی پیکسل متناظرش در تصویر مربوطه می‌باشد. دامنه تغییرات عناصر این ماتریس ممکن است بین ۰ تا ۱ و یا بین ۰ تا ۲۵۵ تغییر کند. در حالت اول داده‌های ماتریس از نوع دقت مضاعف و در حالت دوم از نوع uint8 خواهد بود. بجز توابع تعریف شده در جعبه‌ابزار images و بعضی از توابع خود متلب، سایر عملیات ریاضی بر روی نوع uint8 در حال حاضر امکانپذیر نمی‌باشد. لذا در صورت نیاز، این نوع باید به نوع دقت مضاعف تبدیل شود که میزان حافظه مورد نیاز آن چهار برابر نوع uint8 است.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۲- انواع تصاویر-ادامه

تصاویر شدت-ادامه

نمونه‌ای از یک تصویر شدت:



فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۲- انواع تصاویر-ادامه

تصاویر اندیس‌شده(Indexed Image)

این تصاویر توسط دو ماتریس زیر مشخص می‌شوند:

1. ماتریس اندیس: ماتریسی است که ابعاد آن برابر با ابعاد تصویر بر حسب پیکسل می‌باشد. مقادیر این ماتریس معمولا بین ۱ تا ۲۵۶ تغییر می‌کند و مقدار هر درایه از این ماتریس معرف شماره سطری از ماتریس نقشه‌رنگ است.
2. ماتریس نقشه‌رنگ(map): این ماتریس دارای ۳ ستون می‌باشد و هر سطر از آن معرف یکی از رنگهای موجود در تصویر است. بطوریکه عنصر اول هر سطر معرف نسبت اولیه قرمز، عنصر دوم معرف اولیه سبز و عنصر سوم معرف اولیه آبی است.

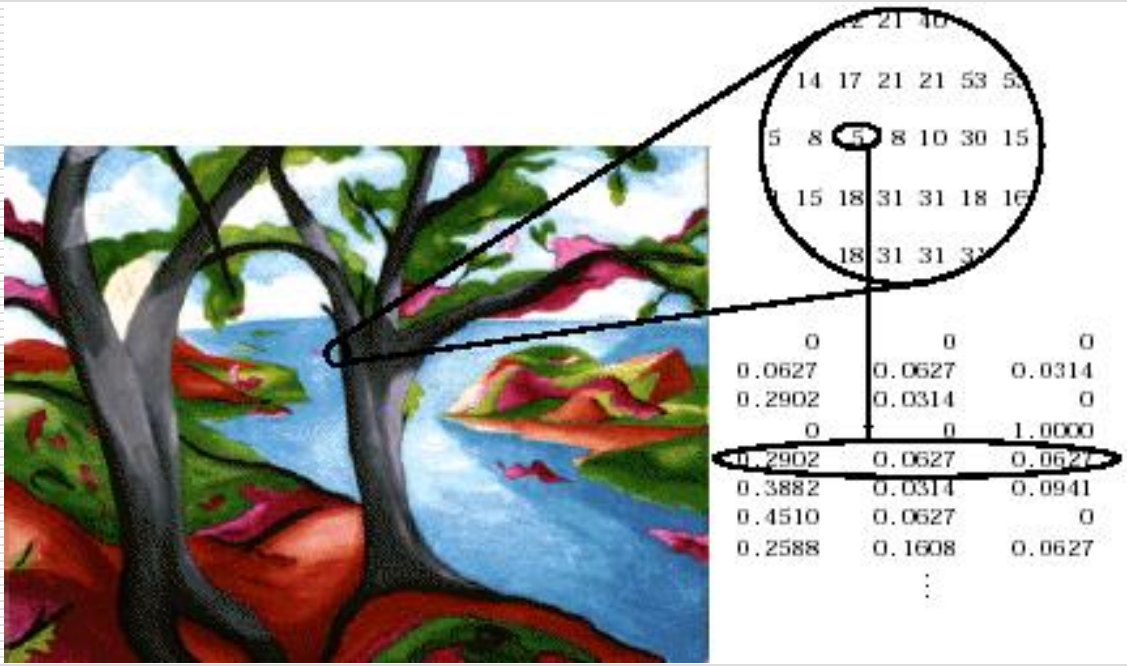
یک تصویر اندیس‌شده بسته به مقادیر ماتریس نقشه‌رنگ، ممکن است رنگی یا سطح خاکستری باشد.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۲- انواع تصاویر-ادامه

تصاویر اندیس‌شده(Indexed Image)-ادامه

نمونه‌ای از یک تصویر اندیس‌شده

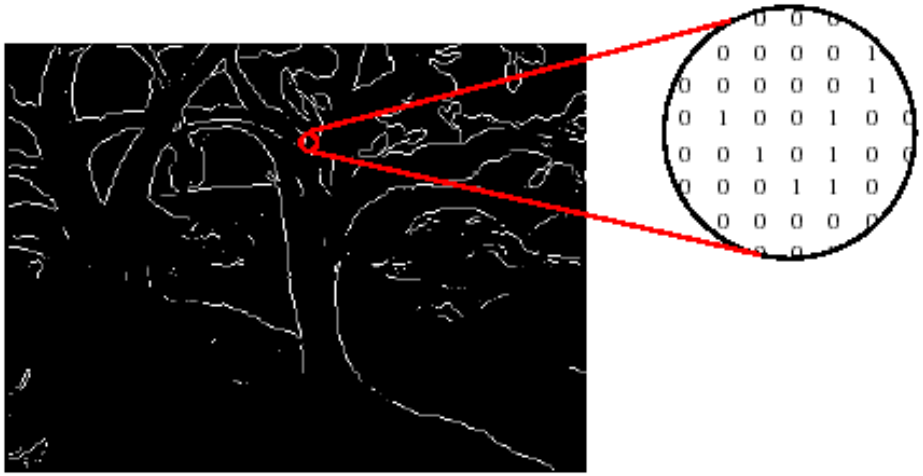


فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۲- انواع تصاویر-ادامه

تصاویر باینری

یک تصویر باینری به تصویری گفته می‌شود که هر پیکسل از آن تنها بتواند دارای یکی از دو مقدار ممکن (معمولا ۰ و ۱) باشد.در متلب این تصاویر می‌توانند با فرمت double و یا uint8 ذخیره‌سازی شوند. اما بطور پیش‌فرض متلب فرمت uint8 را بکار خواهد برد که مقادیر آن می‌تواند، ۰ و ۱ ویا ۰و ۲۵۵ باشد.



فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۲- انواع تصاویر-ادامه

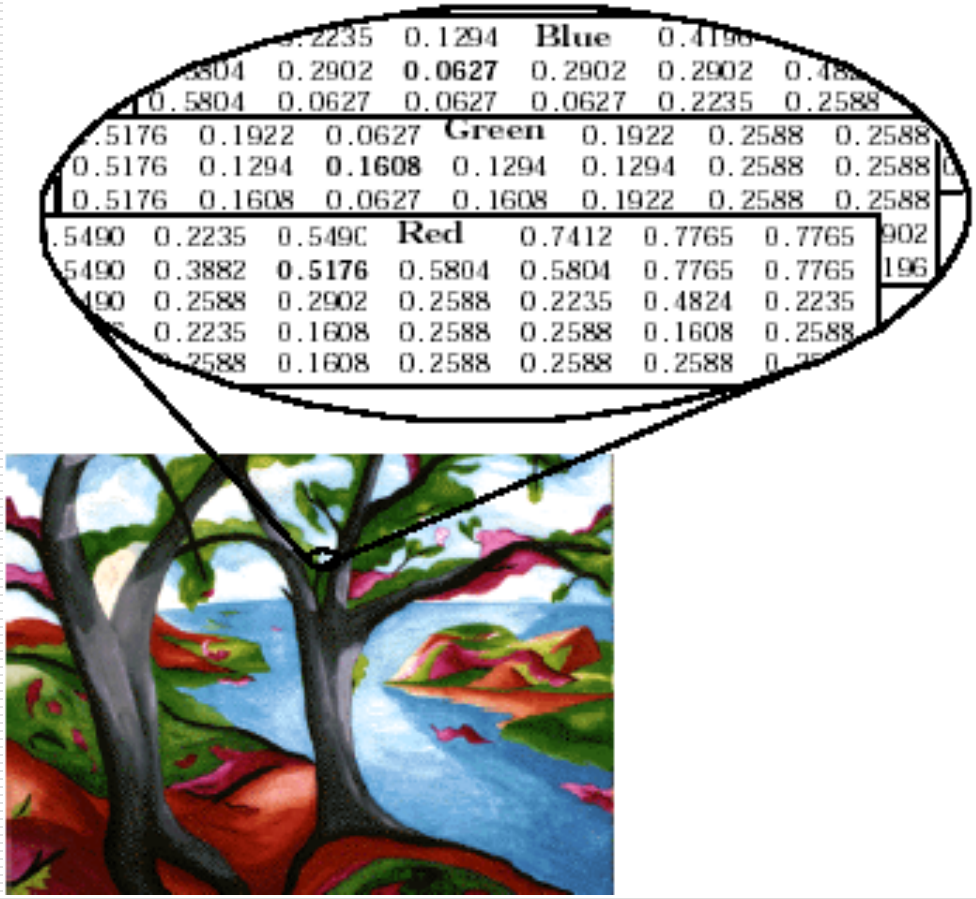
تصاویر RGB

یک تصویرRGB یا true color به تصویری گفته می‌شود که به ازای هر پیکسل از آن سه عدد بین ۰ تا ۲۵۵ در حافظه کامپیوتر ذخیره شده باشد که این اعداد معرف شدت هر یک از اولیه‌های قرمز، سبز و آبی می‌باشد. مثلا برای یک پیکسل سفید سه عدد ۲۵۵ و برای یک پیکسل سبز سه عدد ۰، ۲۵۵ و ۰ به ترتیب معرف شدت اولیه‌های قرمز، سبز و آبی ایجاد خواهد شد. بنابراین برای هر نقطه از تصویر بیش از ۱۶ میلیون(۲۵۶*۲۵۶*۲۵۶) حالت رنگی مختلف امکانپذیر خواهد بود. واضح است که یک تصویر rgb سه برابر یک تصویر شدت هم‌اندازه با آن حافظه کامپیوتر را اشغال خواهد کرد و به همان نسبت هم به زمان پردازش بیشتری نیاز دارد.

در متلب هر تصویر rgb بصورت یک ماتریس سه‌بعدی تعریف می‌شود که در بعد سوم آن مقادیر اولیه‌های رنگی هر نقطه (r,g,b) ذخیره می‌شوند. عناصر این ماتریس ممکن است بین ۰ تا ۱ (double)و یا بین ۰ تا ۲۵۵ (uint8)تغییر کند

دقت شود که یک تصویر rgb لزوما رنگی نیست اما می‌تواند رنگی باشد.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر



۱۳-۲- انواع تصاویر-ادامه

تصاویر RGB-ادامه

یک تصویر rgb نمونه

۱۳-۳- خواندن تصاویر-تابع imread

به منظور خواندن یک فایل گرافیکی در متلب می‌توان از تابع Imread استفاده کرد. بسته به نوع تصویر فرمت کلی استفاده از این تابع به یکی از صورتهای زیر است:

- برای تصاویر شدت، rgb و باینری: m=imread('filename')
- برای تصاویر اندیس‌شده:[m,map]=imread('filename')

که در رابطه اخیر m ماتریس اندیس و map ماتریس نقشه‌رنگ خواهد بود.

نکته: تابع imread را با تعداد آرگومانهای بیشتری نیز می‌توان فراخوانی کرد. جهت اطلاع بیشتر به راهنمای متلب رجوع کنید.

۱۳-۴- نمایش تصاویر-تابع imshow

تابع imshow می‌توان یک تصویر خوانده شده و یا مستقیما یک فایل تصویری را نمایش دهد:

imshow(m); تصویر شدت یا rgb

imshow(I , map) تصویر اندیس‌شده

imshow('filename'); فایل گرافیکی

مثال:

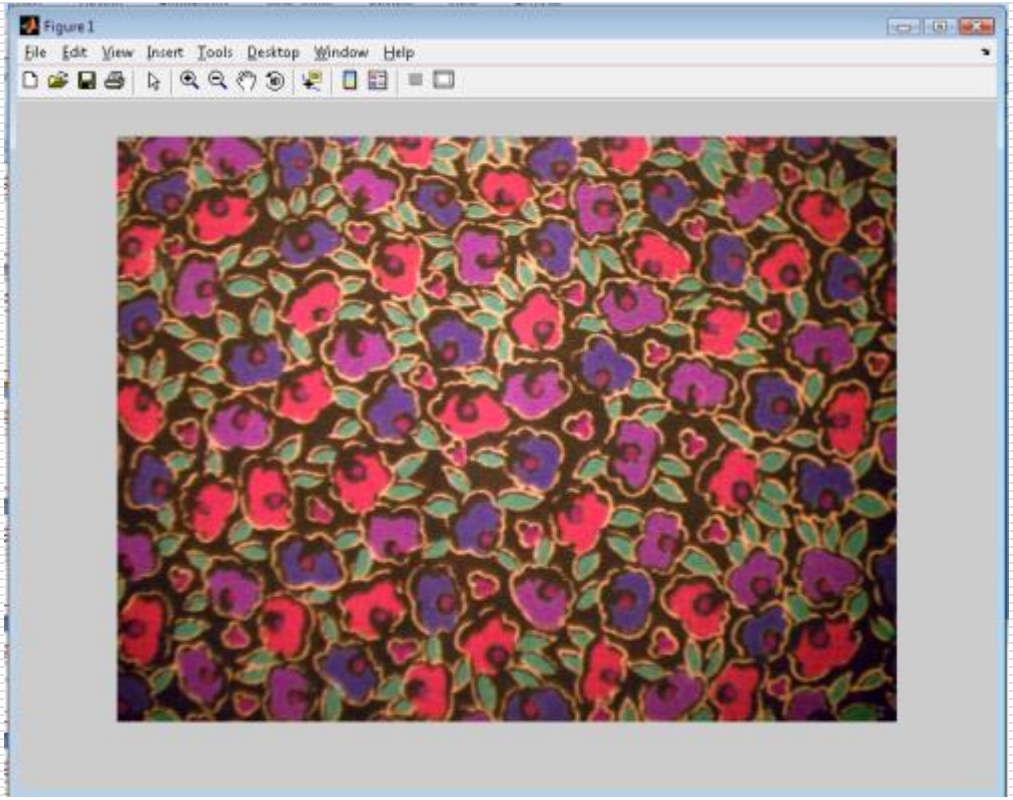
```
>> imshow('fabric.png')
```

یا:

```
>> m=imread('fabric.png');
imshow(m)
```

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۴- نمایش تصاویر-تابع imshow-ادامه



فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۴- نمایش تصاویر-تابع imtool

تابع دیگری که برای نمایش تصاویر در متلب وجود دارد تابع imtool است. روش استفاده از این تابع مانند تابع imshow است اما قابلیتهای بیشتری را در اختیار می‌گذارد:

```
>> imshow('fabric.png')
```

۱۳-۵- نوشتن فایلهای گرافیکی-imwrite

برای ایجاد یک فایل گرافیکی می‌توان از تابع imwrite استفاده کرد. این تابع بسته به نوع تصویر می‌تواند به یکی از روشهای زیر بکار برده شود:

```
imwrite(m , 'filename');
imwrite(X , map , 'filename');
```

۱۳-۶-تعیین مشخصات یک فایل گرافیکی-تابع imfinfo

این تابع اطلاعاتی از فایل گرافیکی مانند: ابعاد تصویر، دقت ابعادی و دقت عمقی، نحوه فشرده‌سازی و... را ارائه می‌دهد. این تابع بصورت زیر بکار برده می‌شود:

```
info=imfinfo('filename')
```

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۷- تبدیل تصاویر

با استفاده از توابع زیر می‌توان نوع یک تصویر را تغییر داد:

```
bw=im2bw(m , level)
bw=im2bw(x , map , level)
```

level سطح آستانه می‌باشد.(که باید بین ۰ تا ۱ باشد)

```
m=ind2gray(x , map);
[x,map]=gray2ind(m);
[x,map]=rgb2ind(m);
m=ind2rgb(x , map);
m=rgb2gray(m);
```

برای کسب اطلاعات بیشتر به راهنمای متلب مراجعه کنید.

۱۳-۸- عملیات ریاضی بر روی تصاویر

در صورتیکه نوع داده‌های تصویر از نوع uint8 باشد امکان بکاربردن عملگرهای ریاضی و بسیاری از توابع متلب بر روی آنها وجود نخواهد داشت. بدین‌منظور پیش از انجام عملیات ریاضی باید نوع داده‌ها را به double تبدیل کرد. پس از انجام عملیات ریاضی در صورت نیاز می‌توان نوع متغیر را به uint8 بازگرداند:

```
m=double(m);
m=im2uint8(m);
```

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۹- عملیات هندسی بر روی تصاویر

منظور از عملیات هندسی هرگونه تغییر در ابعاد تصویر و یا شکل هندسی آن می‌باشد.
سه نوع عملیات هندسی در متلب بر روی تصاویر امکانپذیر است:

- تغییر ابعاد تصویر: تابع imresize
- چرخش تصویر: تابع imrotate
- برش تصویر: تابع imcrop

که در ادامه به هریک خواهیم پرداخت.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۹- عملیات هندسی بر روی تصاویر-ادامه

تغییر ابعاد تصویر: تابع imresize

این تابع به یکی از دو صورت زیر قابل استفاده است:

```
y=imresize(x , a);
y=imresize(x , [m , n]);
```

درحالت اول متغیر a نسبت تغییر در ابعاد تصویر است. مثلا اگر برابر با ۲ باشد یعنی ابعاد تصویر دوبرابر خواهد شد. اگر این عدد کمتر از ۱ باشد تصویر کوچکتر خواهد شد و اگر بیشتر از یک باشد تصویر بزرگتر می‌شود.

در حالت دوم تعداد سطر و ستون جدید تصویر به تابع ارایه میشود که باید اعداد صحیح مثبت باشند.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۹- عملیات هندسی بر روی تصاویر-ادامه

چرخش تصویر-تابع imrotate

نحوه استفاده از این تابع بصورت زیر است:

```
m2=imrotate(m , d , ['Option'] , ['crop'])
```

آرگومان دوم میزان چرخش تصویر برحسب درجه می‌باشد. آرگومان سوم اختیاری بوده و می‌تواند یکی از مقادیر bilinear, nearest یا bicubic باشد. در صورتیکه این آرگومان بکار برده نشود، مقدار پیش‌فرض nearest خواهد بود. آرگومان چهارم نیز اختیاری می‌باشد و تنها می‌تواند مقدار 'crop' را داشته باشد. درصورتیکه بکار برده شود، ابعاد تصویر پس از چرخش تغییر نمی‌کند اما بخشی از تصویر برش داده و حذف می‌شود.

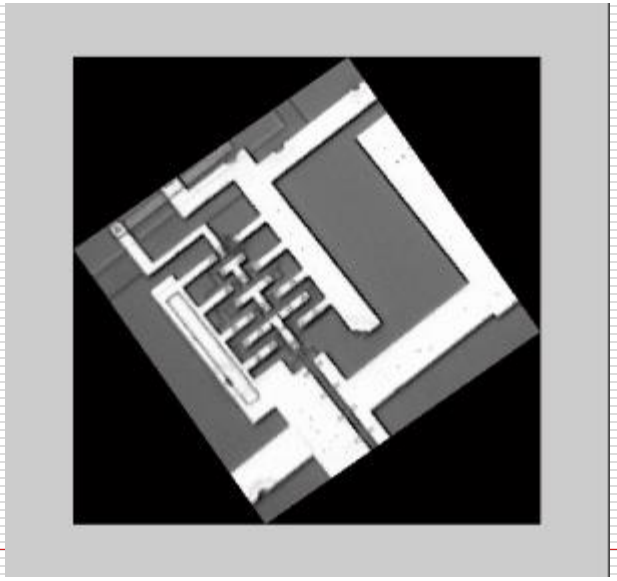
فصل سیزدهم: جعبه‌ابزار پردازش تصویر

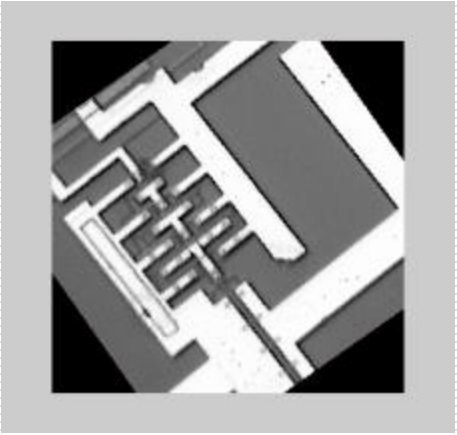
۱۳-۹- عملیات هندسی بر روی تصاویر-ادامه

چرخش تصویر-تابع imrotate-ادامه

مثال:

```
m=imread('ic.tif');
n=imrotate(m , 35); p=imrotate(m , 35,'crop');
imshow(n); figure; imshow(p);
```





فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۹- عملیات هندسی بر روی تصاویر-ادامه

برش تصویر: تابع imcrop

این تابع به یکی از شکلهای زیر قابل استفاده است:

```
I2 = IMCROP(I,RECT)
X2 = IMCROP(X,MAP,RECT)
RGB2 = IMCROP(RGB,RECT)
[A,RECT] = IMCROP(...)
```

که در این روابط rect یک بردار سطری است که مختصات یک ناحیه مستطیلی شکل که از تصویر برش داده می‌شود را مشخص می‌کند. درصورتیکه این آرگومان در ورودی مشخص نشود، تصویر نمایش داده شده و متلب منتظر می‌ماند تا کاربر یک ناحیه مستطیلی را با ماوس انتخاب کند.

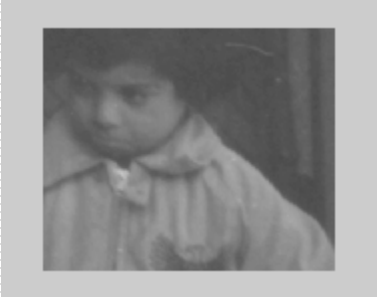
فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۹- عملیات هندسی بر روی تصاویر-ادامه

برش تصویر: تابع imcrop-ادامه

مثال:

```
m=imread('pout.tif');
imshow(m);figure;imcrop(m,[size(m)/4,size(m)/2]
```



۱۳-۱۰- فیلترهای خطی و طراحی فیلتر

برای اعمال یک فیلتر بر روی تصویر می‌توان از تابع filter2 استفاده کرد:

```
m2=filter2(h , m)
```

در رابطه h ماتریس فیلتر و m ماتریس تصویر اولیه است. h می‌تواند هر ماتریس با ابعاد دلخواه باشد، اما معمولا یک ماتریس ۳*۳ یا ۵*۵ است.

۱۳-۱۰- فیلترهای خطی و طراحی فیلتر-ادامه

- فیلترهای آماده

با استفاده از تابع fspecial می‌توان فیلترهای معمول در پردازش تصویر را برای استفاده با تابع filter2 ایجاد کرد. روش استفاده از این تابع بصورت زیر است:

h=fspecial(' نام فیلتر ' , ابعاد فیلتر)

بسته به نوع آرگومان اول ممکن است این تابع با یک یا بیش دو آرگومان نیز بکار برده شود.

نام فیلتر می‌تواند یکی از پارامترهای زیر باشد:

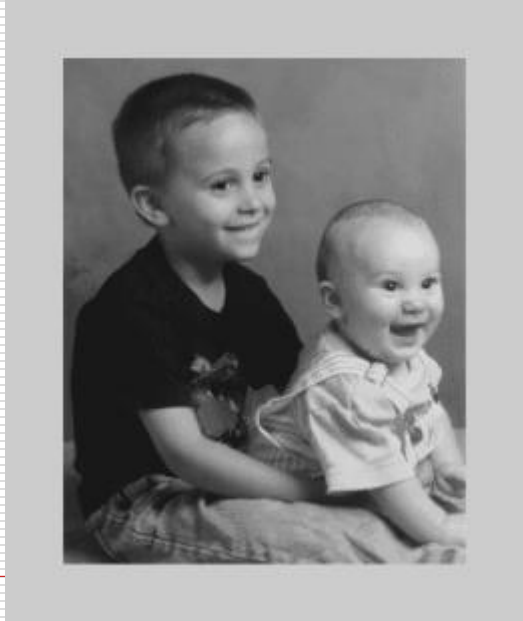
gaussian: پایین گذر
sobel: بالا گذر
prewitt: بالا گذر
laplacian: فیلتر لاپلاس
log: اعمال فیلتر گوسی و پس از آن لاپلاس
average: فیلتر میانگین
unsharp: پایین گذر

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۰- فیلترهای خطی و طراحی فیلتر-ادامه

- فیلترهای آماده-مثال

```
SobelFilter=fspecial('sobel');
[I,map]=imread('kids.tif');I=ind2gray(I,map);
      I2=filter2(SobelFilter,I);
imshow( I ); figure; imshow( I2 );
```



فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر

آنالیز و بهسازی تصویر شامل سه عملیات زیر است:

- بدست آوردن ارزش نقاط تصویر و اعمال عملیات آماری بر روی آنها
- آنالیز تصویر بمنظور استخراج اطلاعات در مورد ساختار کلی آن
- بهسازی تصویر بمنظور واضح‌تر شدن جزییات تصویر و حذف نویز بمنظور آماده‌سازی برای عملیات پردازشی بعدی

که در ادامه به هر یک خواهیم پرداخت

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

- بدست آوردن ارزش نقاط تصویر و اعمال عملیات آماری بر روی آنها

توابع pixval و impixel

با استفاده از تابع impixel می‌توان مشخصات رنگی پیکسلهایی از تصویر را بدست آورد. این تابع بصورتهای زیر بکار می‌رود:

```
P = IMPIXEL(I)
P = IMPIXEL(X,MAP)
P = IMPIXEL(RGB)
```

در این حالت این تابع پنجره تصویر را نمایان ساخته امکان انتخاب نقاط مورد نظر را به کاربر می‌دهد. پس از زدن یک کلید یا دکمه سمت راست ماوس، مشخصات این نقاط در ماتریس p ذخیره خواهد شد. البته این تابع بصورتهای دیگری نیز می‌توان بکار برد که برای کسب اطلاعات بیشتر می‌توانید به راهنمای متلب مراجعه کنید.

تابع pixval به پایین پنجره تصویر کادری را اضافه می‌کند که با حرکت ماوس بر روی تصویر مشخصات رنگی نقاط نصویر در این کادر نمایش داده می‌شود. این تابع باید پس نمایش تصویر با تابع imshow صدا زده شود.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

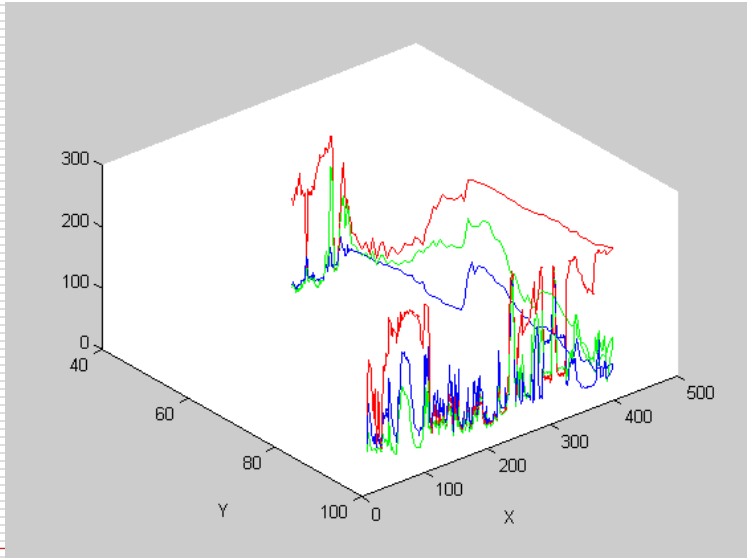
بدست آوردن ارزش نقاط تصویر و اعمال عملیات آماری بر روی آنها-ادامه

تابع improfile:

این تابع نمودار تغییرات رنگ تصویر را در یک مسیر دلخواه که با ماوس انتخاب می‌شود رسم می‌کند:

مثال:

```
imshow('flowers.tif');improfile;
```



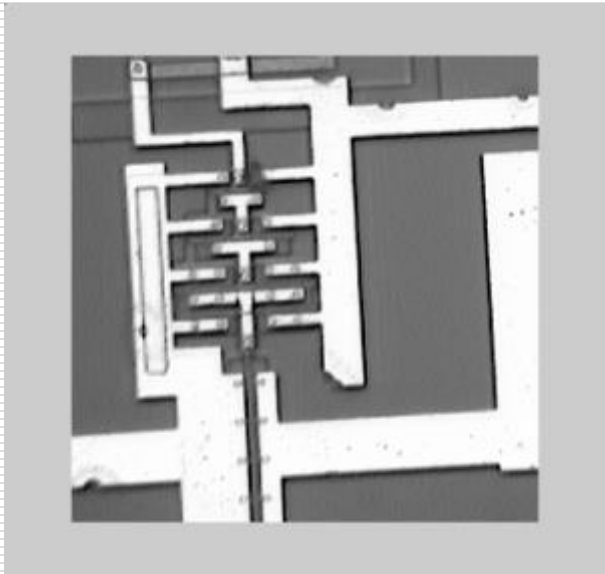
فصل سیزدهم: جعبه‌ابزار پردازش تصویر

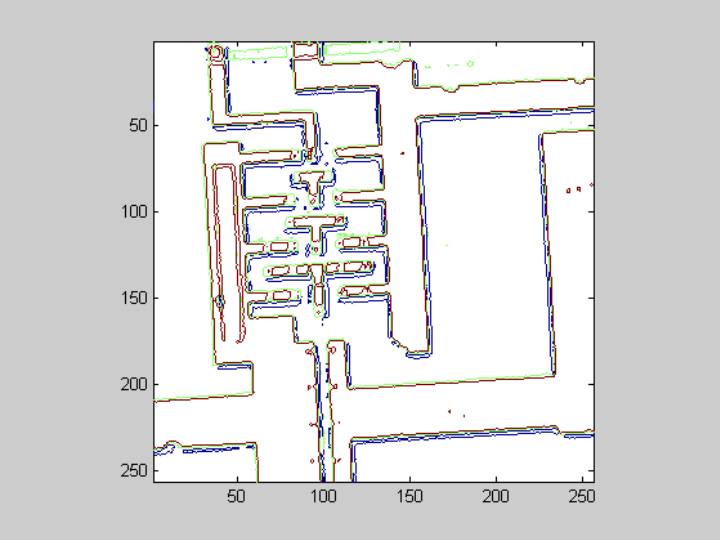
۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بدست آوردن ارزش نقاط تصویر و اعمال عملیات آماری بر روی آنها-ادامه

تابع imcontour: رسم نمودار تراز داده‌های تصویر:

```
im=imread('ic.tif');
imshow(im);figure;imcontour(im,3);
```





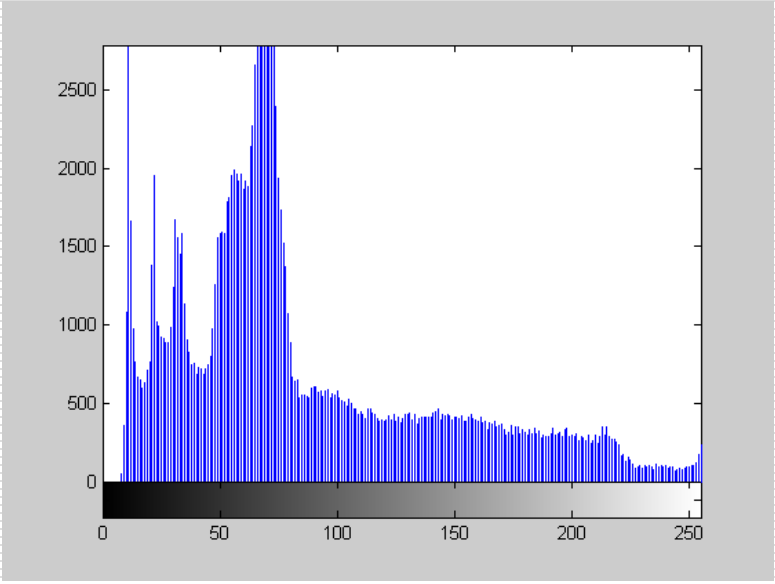
فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بدست آوردن ارزش نقاط تصویر و اعمال عملیات آماری بر روی آنها-ادامه

تابع imhist: رسم نمودار فراوانی نقاط تصویر:

```
I=imread('flowers.tif');I=rgb2gray(I);
imshow(I);figure;imhist(I);
```



فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بدست آوردن ارزش نقاط تصویر و اعمال عملیات آماری بر روی آنها-ادامه

توابع mean2 و std2:

توابع mean و std در متلب به ترتیب برای بدست آوردن میانگین و انحراف معیار بکار برده می‌شوند. اما این توابع بصورت برداری عمل می‌کنند یعنی میانگین یا انحراف معیار عناصر یک بردار را محاسبه می‌کنند. اگر این توابع را بر روی یک ماتریس اعمال کنیم مانند اکثر توابع متلب بصورت ستونی روی عناصر آن ماتریس عمل خواهند کرد. یعنی میانگین یا انحراف معیار هر ستون ماتریس را بصورت جداگانه بدست می‌آورند. برای آنکه بتوان میانگین یا انحراف معیار تمامی نقاط یک ماتریس را بدست آورد باید از توابع mean2 و std2 استفاده کرد.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

آنالیز تصویر:

از آنجاییکه آنالیز تصویر بیشتر بر روی تصاویر باینری انجام می‌گردد این مبحث به سرفصل "عملیات بر روی تصاویر باینری " ارجاع می‌شود.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بهسازی تصویر:

این عملیات که به عملیات پیش‌پردازش نیز مشهور است معمولا پیش از عملیات پردازش اصلی یا عملیات آنالیز تصویر انجام می‌گیرد. در این عملیات بهبودهایی بر روی داده‌های تصویر اعمال می‌شود تا امکان استخراج دقیقتر و صحیح‌تر اطلاعات میسر گردد. این عملیات در سه بخش زیر شرح داده خواهد شد:

- تنظیم شدت
- متعادل کردن هیستوگرام یا بهسازی تباین
- حذف نویز

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بهسازی تصویر-ادامه

تنظیم شدت-تابع imadjust

با استفاده از این تابع می‌توان دامنه تغییرات روشنایی یک تصویر را تغییر داد. شکل کلی کاربرد این تابع بصورت زیر است:

```
J=imadjust(I , [low , high] , [bottom , top])
```

آرگومان دوم برداری دو عنصری است که بیانگر دامنه حاوی روشنایی‌هایی از تصویر است که عملیات تنظیم شدت بر روی آنها باید اعمال گردد. آرگومان سوم، دامنه تغییرات جدید روشنایی برای نقاط فوق است.

مثال:

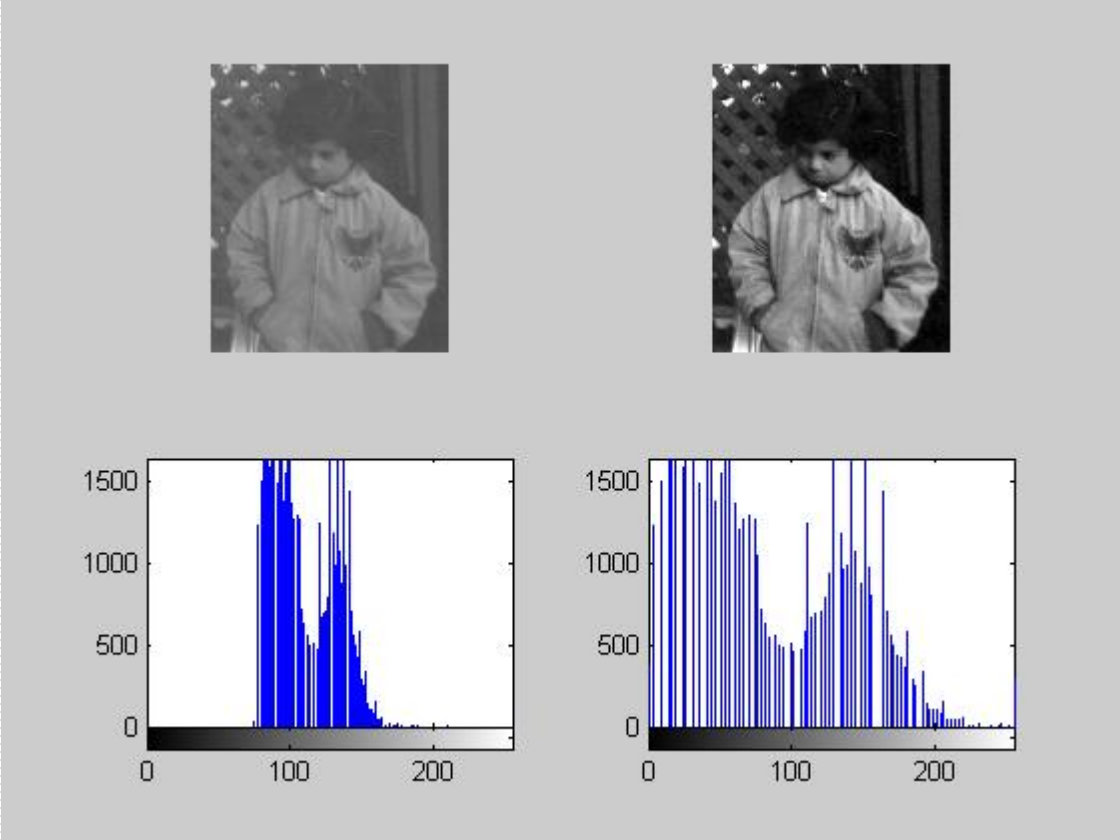
```
I=imread('pout.tif');
J=imadjust(I , [0.3 , 0.7] , [0 ,1]);
subplot(2,2,1);imshow(I); subplot(2,2,2);imshow(J);
subplot(2,2,3); imhist(I); subplot(2,2,4); imhist(J)
```

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بهسازی تصویر-ادامه

تنظیم شدت-تابع imadjust-ادامه



فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بهسازی تصویر-ادامه

متعادل کردن هیستوگرام یا بهسازی تباین-تابع histeq

تابع histeq بصورت اتوماتیک بهترین تنظیم هیستوگرام را بر روی تصویر انجام می‌دهد و معمولا کیفیت روشنایی تصویر را به میزان زیادی بهبود می‌بخشد.

مثال:

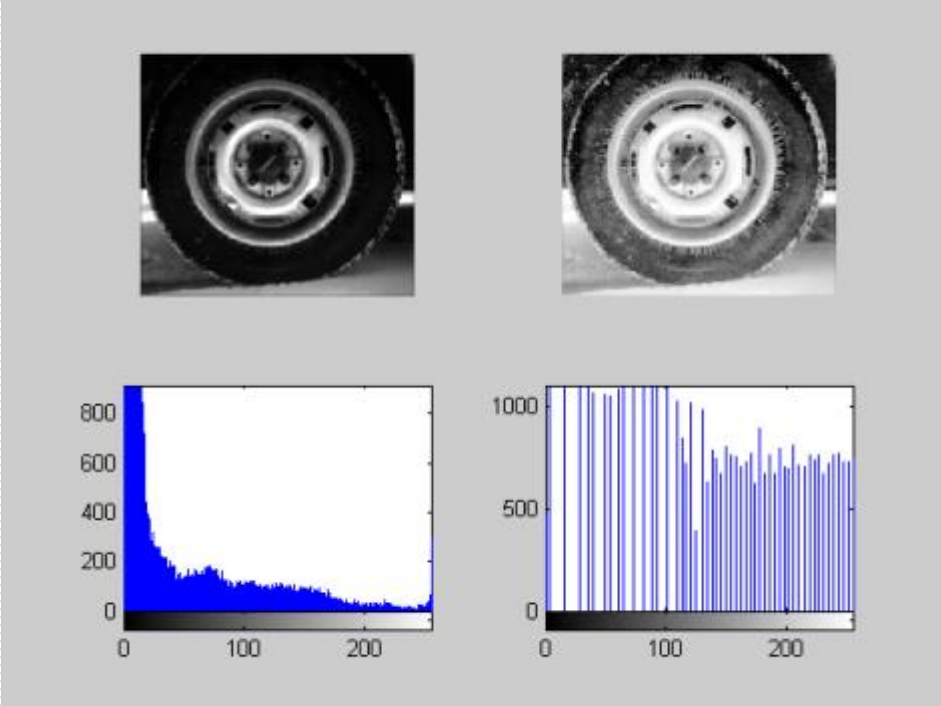
```
I=imread('tire.tif');
J=histeq(I);figure;
subplot(2,2,1);imshow(I);
subplot(2,2,2);imshow(J);
subplot(2,2,3);imhist(I);
subplot(2,2,4);imhist(J);
```

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بهسازی تصویر-ادامه

متعادل کردن هیستوگرام یا بهسازی تباین-تابع histeq-ادامه



فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بهسازی تصویر-ادامه

حذف نویز

معمولا تصاویر دیجیتال کم و بیش دارای نویز هستند. حذف نویز قبل از هرگونه عملیات پردازشی باید انجام گیرد. فیلترهای متعددی برای حذف نویز طراحی شده‌اند. در متلب نیز چندین فیلتر برای حذف نویز وجود دارد که از این میان به ساده‌ترین آنها اشاره خواهیم کرد:

- فیلتر میانگین
- فیلتر میانه

برای ایجاد فیلتر میانگین از تابع fspecial که قبلا توضیح داده شد و تابع filter2 می‌توان استفاده کرد. برای اعمال فیلتر میانه از تابع medfilt2 استفاده کنید. بطورکلی تمامی فیلترهای حذف نویز از وضوح (sharpness) تصویر می‌کاهند. در میان دو فیلتر میانگین و میانه، فیلتر میانه معمولا نتیجه بهتری ایجاد می‌کند و وضوح تصویر را نیز کمتر تحت تاثیر قرار می‌دهد.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بهسازی تصویر-ادامه

حذف نویز-مثال: مقایسه فیلتر میانه و فیلتر میانگین

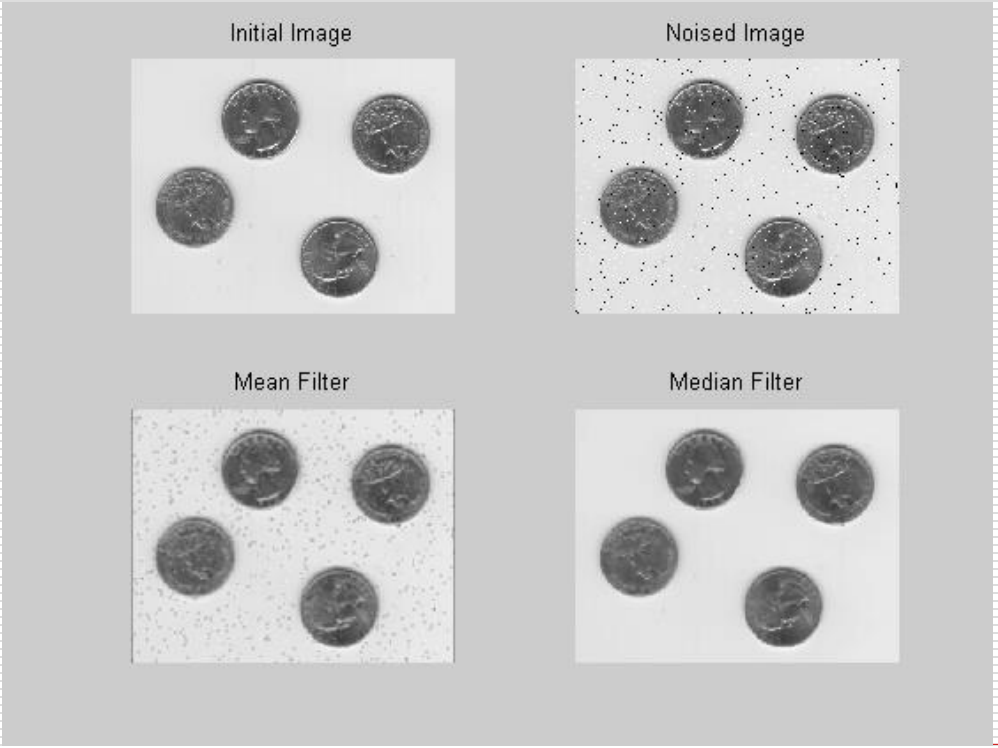
```
I = imread('eight.tif');
J= imnoise(I , 'Salt & pepper' , 0.02); % افزودن نویز
K= filter2(fspecial('average' , 3) , J) / 255; % فیلتر میانگین
L=medfilt2(J , [3 , 3]); % فیلتر میانه
subplot(2,2,1); imshow( I ); title('Initial Image')
subplot(2,2,2); imshow( J ); title('Noised Image');
subplot(2,2,3); imshow( K ); title('Mean Filter');
subplot(2,2,4); imshow( L ); title('Median Filter');
```

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۱-آنالیز و بهسازی تصویر-ادامه

بهسازی تصویر-ادامه

حذف نویز-مثال: مقایسه فیلتر میانه و فیلتر میانگین-ادامه



فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۲- عملیات بر روی تصاویر باینری

اگرچه عملیات بر روی تصاویر باینری زیرمجموعه مبحث آنالیز تصویر است لکن بخاطر اهمیت تصاویر باینری در علم پردازش تصویر، این مبحث را در بخش جدیدی ارایه نموده‌ایم.

همانگونه که قبلا گفته شد تصویر باینری به تصویری گفته می‌شود که پیکسلهای آن تنها دارای یکی از دو مقدار ممکن ۰ و ۱ یا ۰ و ۲۵۵ باشند. در متلب تصاویر باینری می‌توانند بصورت تصاویر شدت و یا بصورت تصاویر اندیس‌شده ذخیره و معرفی شوند. در حالت دوم ماتریس نقشه رنگ تنها دارای دو سطر خواهد بود.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۲- عملیات بر روی تصاویر باینری

نمایش تصاویر باینری

برای نمایش تصاویر باینری نیز از تابع imshow استفاده می‌شود. درصورتیکه تصویر از نوع شدت باشد فرم : imshow(m) و اگر از نوع اندیس‌شده باشد فرم : imshow(I ,map) بکار برده خواهد شد.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۲- عملیات بر روی تصاویر باینری

عملیات ساختاری Morphological Operations

عملیات ساختاری به عملیاتی گفته می‌شود که بر روی تصاویر باینری اعمال شده و هدف از آن ایجاد تغییر و یا تصحیح در اجزا داخل یک تصویر باینری باشد. این عملیات معمولا یک مرحله قبل از عملیات پردازش نهایی انجام میشود. منظور از عملیات پردازش نهایی عملیاتی است که در آن اطلاعاتی از تصویر استخراج میشود. مثلا محیط یا مساحت اجزا تصویر محاسبه می‌گردد.

از میان این عملیات در ادامه چهار نوع از مهترین آنها شرح داده خواهد شد که عبارتند از:

- عملیات افزایش
- عملیات فرسایش
- عملیات گشودن
- عملیات بستن

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۲- عملیات بر روی تصاویر باینری-ادامه

عملیات ساختاری Morphological Operations-ادامه

عملیات افزایش و فرسایش(Dilation & Erosion)

منظور از عملیات افزایش عملیاتی است که باعث افزایش ابعاد اجزا داخل تصویر به اندازه یک یا چند پیکسل می‌گردد. در اثر این عمل ممکن است نقاطی که از یک تصویر باینری در اثر عواملی چون تاثیر نویز یا اعمال حد آستانه نامطلوب جا افتاده است، تصحیح گردند. مثلا ممکن است دو جزء از تصویر به یکدیگر متصل گردند. الگوریتم اعمال فیلتر افزایش بدین صورت است که تمامی نقاط سیاه تصویر بررسی شده در صورتیکه حداقل یکی از همسایگان انتخابی نقطه مورد بررسی سفید باشند، نقطه مزبور نیز سفید خواهد شد در غیر اینصورت سیاه باقی خواهد ماند.

عملیات فرسایش دقیقا عکس عملیات افزایش است. در این عملیات معمولا نقاط ناخواسته تصویر باینری حذف می‌شوند و سایر اجزا تصویر نیز به اندازه یک یا چند پیکسل نازکتر خواهند شد. عملا تمامی نقاط سفید تصویر بررسی شده در صورتیکه حداقل یکی از همسایگان انتخابی آن سیاه باشد، آن نقطه نیز سیاه خواهد شد.

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۲- عملیات بر روی تصاویر باینری-ادامه

عملیات ساختاری Morphological Operations-ادامه

عملیات افزایش و فرسایش-ادامه

ابعاد همسایگی و انتخاب همسایه‌ها توسط یک ماتریس ماسک (Mask) مشخص می‌شوند. مثلا اگر ماتریس ماسک یک ماتریس ۳* ۳ باشد که تمای عناصر آن برابر با ۱ باشد. یعنی یک همسایگی ۳*۳ بکار برده شود و تمامی ۹ همسایه نقطه مورد بررسی برای عملیات افزایش یا فرسایش مد نظر قرار گیرند.

برای عملیات افزایش در متلب از تابع imdilate و برای عملیات فرسایش از تابع imerode استفاده کنید. اگرچه هردو عملیات را با استفاده از تابع کلی‌تر bwmorph نیز می‌توان انجام داد.

فرمول کلی استفاده از این توابع بصورت زیر است:

```
bw2=imerode(bw1, se);
bw2=imdilate(bw1 , se);
```

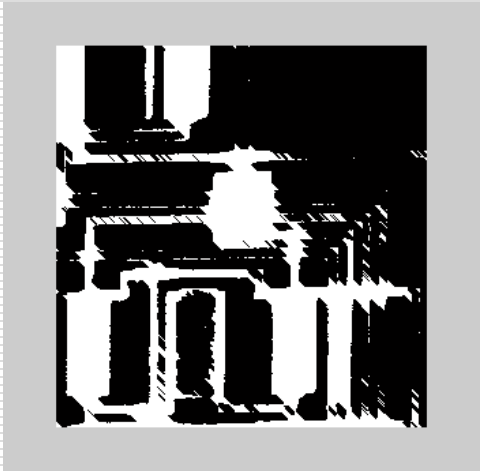
فصل سیزدهم: جعبه‌ابزار پردازش تصویر

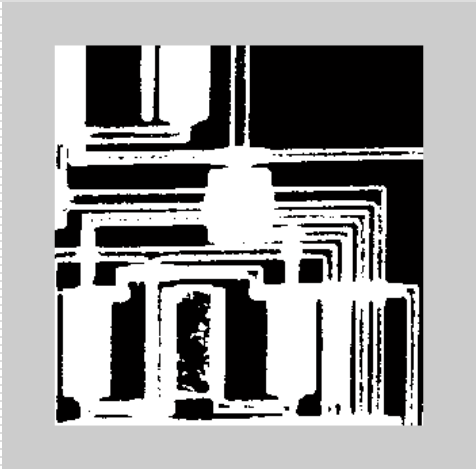
۱۳-۱۲- عملیات بر روی تصاویر باینری-ادامه

عملیات ساختاری Morphological Operations-ادامه

عملیات افزایش و فرسایش-مثال

```
bw1=imread('circbw.tif'); SE=eye(5);
bw2=imerode(bw1 , SE);
imshow(bw1); figure; imshow(bw2);
```





فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۲- عملیات بر روی تصاویر باینری-ادامه

عملیات ساختاری Morphological Operations-ادامه

عملیات گشودن و بستن Open & Close

از ترکیبهای مختلف دو عملیات افزایش و فرسایش می‌توان عملیات دیگری ایجاد کرد. مهمترین این عملیات، عملیات گشودن و بستن است. در عملیات گشودن اجزایی از تصویر باینری که از یک اندازه تعیین شده کوچکتر باشند حذف می‌شوند بدون آنکه ابعاد سایر اجزا تغییر کند. در عملیات بستن نیز نواحی جاافتاده تصویر باینری بدون تغییر در ابعاد سایر اجزا ترمیم می‌گردند.

عملا در صورتیکه ابتدا عملیات فرسایش و سپس افزایش بر یک تصویر باینری اعمال شود، نتیجه، عملیات گشودن خواهد بود اما اگر ابتدا افزایش و سپس فرسایش اعمال گردد، عملیات بستن حاصل خواهد شد.

در متلب برای اعمال عملیات گشودن و بستن و همچنین سایر عملیات مورفولوژی از تابع bwmorph باید استفاده کرد. اگرچه می‌توان این دو عملیات را از عملیات فرسایش و افزایش نیز بدست آورد.(همانگونه که در مثال بعدی عمل شده است)

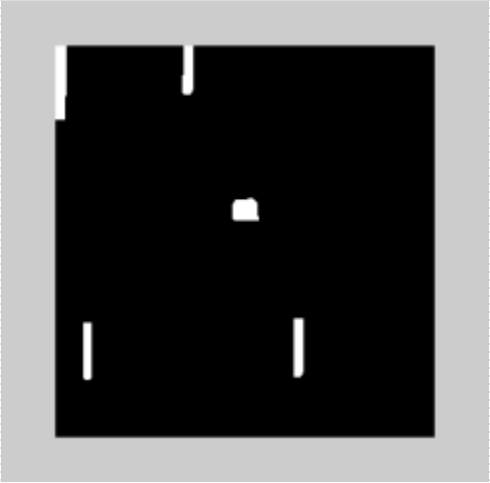
فصل سیزدهم: جعبه‌ابزار پردازش تصویر

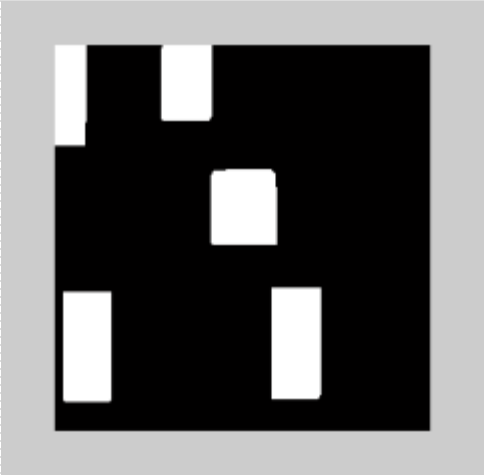
۱۳-۱۲- عملیات بر روی تصاویر باینری-ادامه

عملیات ساختاری Morphological Operations-ادامه

عملیات گشودن و بستن Open & Close-مثال

```
bw1=imread('circbw.tif');
se= ones(40 , 30); bw2= imerode(bw1 , se);
bw3=imdilate(bw2 , se);
imshow(bw2); figure; imshow(bw3);
```





فصل سیزدهم: جعبه‌ابزار پردازش تصویر

۱۳-۱۲- عملیات بر روی تصاویر باینری-ادامه

عملیات ساختاری Morphological Operations-ادامه

عملیات از پیش تعریف شده: تابع immorph

با استفاده از تابع immorph می‌توان بسیاری از عملیات ساختاری معروف پردازش تصویر را اعمال نمود. شکل کلی استفاده از این تابع بصورت زیر است:

```
bw2 = bwmorph(bw1 , operation , [n]);
```

آرگومان سوم اختیاری بوده و بیانگر ابعاد ماسک مورد استفاده یا فاکتور دیگری با توجه نوع آرگومان دوم در عملیات است. در صورت حذف آرگومان سوم، مقدار پیش فرض آن بکار برده خواهد شد. مقدار آرگومان دوم یکی از رشته‌های زیر است:

erode fill hbreak open skel remove close dilate

مثال بعدی نتیجه عملیات اسکلتون را بر روی تصویر قبلی نشان می‌دهد

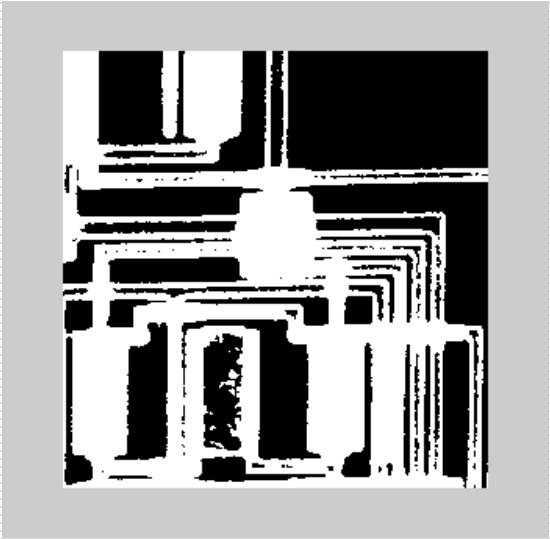
فصل سیزدهم: جعبه‌ابزار پردازش تصویر

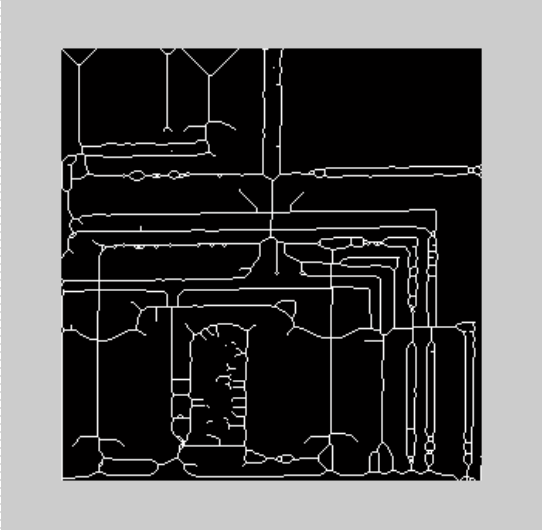
۱۳-۱۲- عملیات بر روی تصاویر باینری-ادامه

عملیات ساختاری Morphological Operations-ادامه

عملیات از پیش تعریف شده: تابع immorph- مثال:

```
bw1= imread('circbw.tif'); bw2= bwmorph(bw1 , 'skel' , inf)
imshow(bw1); figure; imshow(bw2);
```





فصل سیزدهم: جعبه‌ابزار پردازش تصویر

تکلیف ۱۳-۱- تصویری به نام flower.tif از نوع **rgb** در دست است. این تصویر شامل یک گل به رنگ قرمز و ساقه و برگ به رنگ سبز بر روی یک زمینه آبی است. برنامه‌ای بنویسید که :

الف – تصویر فوق را خوانده و داده‌های آنرا در ماتریسی به نام **m** بریزد

ب- با استفاده از حد آستانه ۱۲۰ برای جزء سبز و حد آستانه ۱۸۰ برای جز قرمز، دو تصویر باینری بنامهای **b1** و **b2** ایجاد کند که در اولی تنها تصویر گل و در دومی تنها اجزاء ساقه و برگ وجود داشته باشند.

راهنمایی: برای استخراج برگها تنها استفاده از یک شرط برای حد آستانه کافی نیست. مثلا شرط: **m(:,:,2)>120 & m(:,:,1) < 100** را امتحان کنید.

ج- مرز گل را در تصویر **b1** استخراج کرده و در **b11** بریزد.

د- تصاویر **b11** و **b2** را با استفاده از عملگر یای منطقی در متلب، با یکدیگر تلفیق نماید تا تصویر باینری **c** بدست آید.

ه- مساحت برگ و ساقه و مساحت و محیط گل را از تصاویر **b1**، **b11** و **b2** بدست آورد.

و- مختصات نخستین پیکسل سفید(نسبت به گوشه بالا-سمت چپ تصویر) در تصاویر **b1** و **b2** را بدست آورد.

ز- با استفاده از دستور **text** و نتایج قسمتهای "ه" و "و" پس از نمایش تصویر مساحت و محیط هر جز را در کنار آن نمایش دهد

فصل سیزدهم: جعبه‌ابزار پردازش تصویر

تکلیف ۱۳-۲- تصویر یک پارچه سفید با نام fabric.tif و از نوع شدت(grayscale) در دست است. این تصویر دارای یک طرح بافت خاص می‌باشد برنامه‌ای بنویسید که با استفاده از تبدیل فوریه یک بعدی فرکانس تکرار طرح مزبور در جهت افقی و عمودی و با استفاده از این فرکانسها و طول و عرض تصویر، ابعاد طرح فوق را محاسبه کند و نمایش دهد. رزولوشن تصویر را ۶۰۰ dpiدر نظر بگیرید.

راهنمایی:بدین منظور یک سطر و یک ستون از تصویر را انتخاب و طیف فوریه آنرا بدست آورید...