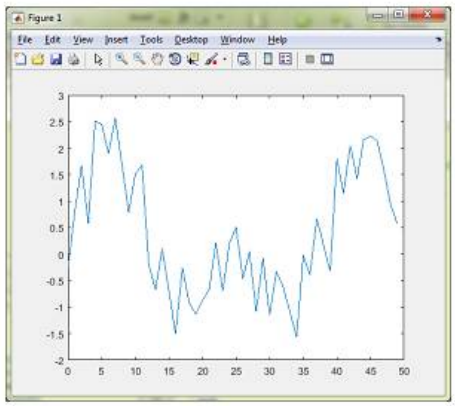
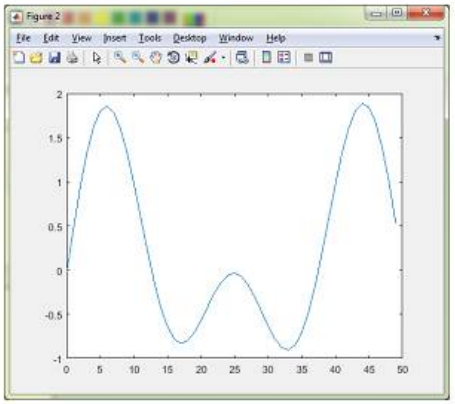
به نام خدا

جزوه برنامه نویسی متلب، جلسه هشتم

رفع نویز در فضای فرکانسی : هرگاه یک سیگنال داشته باشیم و به آن اندکی نویز اضافه کنیم سپس سری فوریه آن سیگنال را محاسبه نماییم مشاهده می‌کنیم در نمودار توان سیگنال بر حسب فرکانس توان مربوط به فرکانس‌های اصلی بالا و توان مرتبط با فرکانس‌های ناشی از سیگنال بسیار کم است لذا با حذف آن فرکانس‌ها و تبدیل فوریه معکوس می‌توانیم سیگنال اصلی را نویز-زدایی کنیم. البته در حالت واقعی باید از فیلترهای دیجیتال استفاده کـــرد.

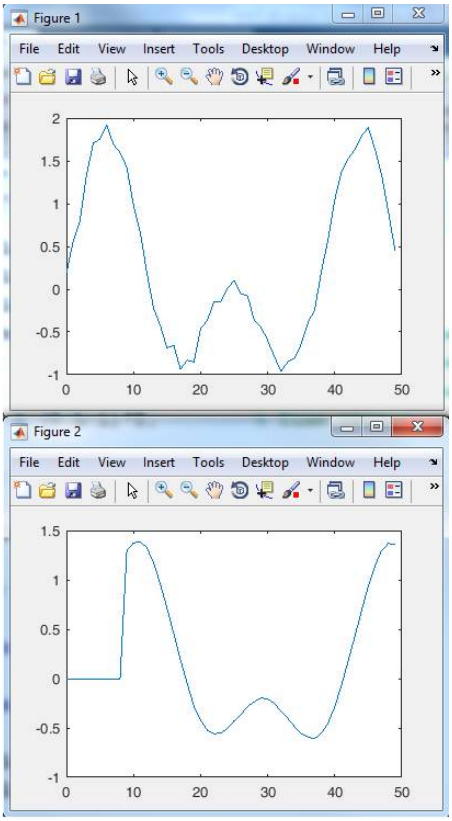
```
close all
clearvars
Fs = 1000;              % Sampling frequency
T = 1/Fs;               % Sampling period
L = 1500;               % Length of signal
t = (0:L-1)*T;          % Time vector

x = sin(2*pi*50*t) + sin(2*pi*30*t);
x = x + 0.5*randn(size(t));
plot(1000*t(1:50),x(1:50))
y = fft(x);
P = abs(y);

figure

for k = 1 : 1500
    if(abs(P(k)<100))
        y(k)=0;
    end
end

w = ifft(y);
plot(1000*t(1:50),w(1:50))
```





رفع نویز در فضای زمانی : هرگاه یک سیگنال داشته باشیم و به آن اندکی نویز اضافه کنیم سپس روی محور زمان به ازای هر نقطه میانگین ۱۰ نقطه قبل را قرار دهیم عمل میانگین متحرک moving average را انجام داده‌ایم و این عمل منجر به کاهش نویز می‌گردد. کد زیر نشان دهنده این روش است.

```
clc
close all
clearvars
Fs = 1000;            % Sampling frequency
T = 1/Fs;             % Sampling period
L = 1500;             % Length of signal
t = (0:L-1)*T;        % Time vector

x = sin(2*pi*50*t) + sin(2*pi*30*t);
x = x + 0.08*randn(size(t));
plot(1000*t(1:50),x(1:50))

w = [];

for k = 10 : 1500
    w(k) = mean(x(k:-1:k-9));
end

figure
plot(1000*t(1:50),w(1:50))
```



هـــرگاه سیگنال بصورت لایو دریافت شود فقط به مقادیر قبلی آن دسترسی داریم اما اگر سیگنال ذخیره شده باشد یا بتوانیم یک تاخیر کوچک را بپذیریم آنگاه می‌توانیم میانگین ۵ نقطه قبل و بعد را لحاظ کنیم در آن صورت کیفیت سیگنال نهایی بهبود خواهد یافت.

```
for k = 6 : 1495
    w(k) = mean(x(k-5:k+5));
end
```

پـــردازش تصویر دیجیتال : پردازش سیگنال دیجیتال شاخه ای از دانش مهندسی و ریاضیات کاربردی است که در آن سیگنالهای ورودی با استفاده از الگوریتمهای مختلف ریاضی تحلیل شده و مورد پردازش قرار می گیرند هرگاه تصویر به عنوان یک سیگنال دیجیتال ورودی در نظر گرفته شود مجموعه روشهای پردازش آن با بکارگیری الگوریتمهای ریاضی و به کمک استفاده از رایانه ***«پردازش تصویر دیجیتال»*** نام می گیرد. لذا پردازش تصویر دیجیتال شاخه‌ای از دانش رایانه است که در آن تصویر به عنوان سیگنال گسسته با بکارگیری الگوریتمهای مختلف تحلیل شده و مورد پردازش قرار می گیرد.

پردازش تصویر عمدتاً برای مقاصد ذیل انجام می گیرد:

الف) بهینه سازی روشهای ذخیره سازی و انتقال تصویر.

ب) بهبود کیفیت تصویر.

ج) تغییر رنگ، ابعاد و تفکیک پذیری تصویر.

د) تجزیه و تحلیل تصویر به منظور مناسب کردن آن برای کاربردهای بینایی ماشین.

کاربردهای پردازش تصویر بسیار گسترده است. امروزه می توان کاربردهای پردازش تصویر دیجیتال را در زندگی روزمره به وضوح دید. دوربین‌ها و تلویزیون‌های دیجیتال، گوشی های تلفن همراه با قابلیت ذخیره و پخش تصاویر ثابت و متحرک، رایانه های همراه، دستگاههای چاپگر و پویشگر و کلیه تجهیزات رایانه ای که به نحوی با تصاویر در ارتباط هستند جلوه هایی از کاربرد پردازش تصویر دیجیتال در زندگی روزمره می باشند. سایر کاربردهای پردازش تصویر را می توان در پزشکی، صنایع نظامی و اتوماسیون صنعتی یافت.

دانش پردازش تصویر به انواع مختلفی از تصاویر ثابت و متحرک (ویدئو) می پردازد. منشا تصاویر ورودی می‌تواند دوربین دیجیتال، پویشگر و یا خود رایانه باشد.

افزایش کیفیت تصویر دیجیتال به روشهای مختلف انجام می‌گیرد که اهم آن عبارتست از:

الف) کاهش نویز.

ب) کاهش تاری حرکت (motion blur) .

ج) بهبود لبه ها.

د) تغییر کنتراست و روشنایی.

تصویر دیجیتال دو بعدی ثابت به طول L و عرض W عبارتست از یک تابع دو متغیره گسسته که به ازای هر مختصات ورودی برداری را در خروجی بدست می‌دهد.

هر زوج x و y مبین یک نقطه در تصویر است که به آن *پیکسل* (Pixel) گفته می شود. هر پیکسل دارای خصوصیت‌هایی نظیر رنگ یا شدت نور است. بردار خروجی بر اساس نوع تصویر (رنگی، سیاه و سفیدو غیره) این خصوصیات را منتقل می‌کند. مثلا در تصویر سیاه و سفید (bi-tonal black-and-white) بردار خروجی تک بعدی است و فقط دارای یک مقدار صفر یا یک است. همچنین در تصویر مقیاس خاکستری (grayscale) بردار خروجی نیز تک بعدی است با این تفاوت که مقدار آن عددی صحیح

بین ۰ تا ۲۵۵ می‌باشدکه شدت روشنایی آن پیکسل را نشان می دهد. در تصویر رنگی RGB که متداولترین نوع تصویر است بردار خروجی سه بعد دارد که هر بعد نشان دهنده شدت نور یکی از رنگ‌های اصلی قرمز سبز و آبی است.

از دیدگاه دیگر می توان تصویر دو بعدی را یک ماتریس دانست که درایه‌های آن بردارهای n بعدی هستند. هر درایه دارای اندیس i و j است این اندیس نشان‌دهنده مکان پیکسل بصورت سطر و ستون است و مقدار درایه یک بردار است که مولفه‌های آن می‌تواند مقادیر رنگ آن پیکسل را مشخص کند. علاوه بر مقادیر رنگ مولفه‌های دیگری نظیر مقدار شفافیت پیکسل را هم می‌توان ذخیره کرد. اگر درایه‌های ماتریس به جای بردار یک عدد یک بایتی باشد و آن عدد نشان‌دهنده شدت روشنایی پیکسل باشد تصویر از نوع مقیاس خاکستری (grayscale) خواهد بود. درایه‌های ماتریس مورد نظر می‌توانند خود اندیس ماتریس دیگری بنام ماتریس پالت باشند در این صورت کل رنگ‌های تصویر در یک ماتریس بنام پالت ذخیره می‌شود و درایه‌های ماتریس اصلی فقط نشان‌دهنده اندیس رنگ مورد نظر هستند.

- گرافیک برداری (Vector Graphics) و گرافیک شطرنجی (Raster Graphics)

تصاویر رایانه‌ای در هنگام نمایش از تعدادی نقطه بنام پیکسل تشکیل شده‌اند هرگاه اطلاعات تک تک پیکسل‌ها ذخیره و بازیابی شوند گرافیک مورد نظر از نوع شطرنجی یا raster است. اما نوع دیگری از گرافیک رایانه‌ای وجود دارد که در آن بجای ذخیره‌سازی اطلاعات پیکسل‌ها، اطلاعات اجزاء تصویر بصورت فرمول‌ها و معادلات ریاضی ذخیره و بازیابی می‌شوند. بکارگیری اجزاء ریاضی مانند خط، دایره، چند ضلعی و منحنی که با معادلات ریاضی قابل توصیف هستند اساس گرافیک برداری (vector graphics) را تشکیل می‌دهد. گرافیک برداری برای نمایش اشکال هندسی ساده بسیار مناسب‌تر و کارآمدتر از گرافیک شطرنجی (raster) می‌باشد. مثلا برای ترسیم یک دایره بجای ذخیره‌سازی تمام پیکسل‌های محیط آن دایره کافی است مقدار عدد شعاع، رنگ و ضخامت قلم و مختصات مرکز دایره ذخیره شوند. با داشتن این اطلاعات می توان دایره را ترسیم کرد. در واقع مختصات تمام نقاط محیط دایره در معادله ریاضی دایره ذخیره شده است و در هنگام ترسیم این نقاط مشخص می‌شوند. گرافیک برداری برای تصاویری که از اجزاء هندسی ساده تشکیل شده است بسیار مناسب می‌باشد اما برای تصاویر طبیعی مناسب نیست. تصویر شطرنجی (raster) در اثر بزرگنمایی (zoom in) کیفیت خود را از دست می‌دهند اما تصاویر برداری در هر اندازه و مقیاسی کیفیت مشابه دارند. حجم تصاویر برداری نسبت به تصاویر شطرنجی raster بسیار کمتر است.

- تفکیک پذیری مکانی (spatial resolution)

هر تصویر از تعدادی نقطه (پیکسل) تشکیل شده است. تعداد پیکسل‌ها در امتداد محور x طول تصویر را تشکیل می‌دهد. تعداد پیکسل‌ها در امتداد محور y عرض تصویر را تشکیل می‌دهد. حاصلضرب طول در عرض تصویر «*تفکیک‌پذیری مکانی*» تصویر را بدست می‌دهدکه حتما بصورت دو عدد جدا از هم با یک نماد ضرب در وسط آن دو نوشته می‌شود زیرا باید طول و عرض به

تفکیک مشخص باشند. بطور مثال تصویری که طول آن ۴۸۰ و عرض آن ۳۲۰ است دارای تفکیک پذیری مکانی 480 x 320 می‌باشد. از دیدگاه ماتریسی عدد اول از سمت چپ (۴۸۰) تعداد ستونهای ماتریس است و عدد دوم (۳۲۰) تعداد سطرهای ماتریس را نشان می دهد. در برخی منابع بجای تفکیک پذیری مکانی از واژه «تفکیک پذیری پیکسل» استفاده می‌شود و بجای آن تعریف متفاوتی از تفکیک پذیری مکانی ارائه می دهند که عبارتست از تعداد پیکسل های متفاوت در واحد طول.

- تفکیک پذیری زمانی (temporal resolution)

هرگاه تصویر متحرک (ویدئو) را در نظر بگیریم به تعداد فریم که در هر ثانیه گرفته، ذخیره یا پخش می‌شود «*تفکیک‌پذیری زمانی*» می‌گویند. واحد آن «*فریم بر ثانیه*» و نماد آن fps می‌باشد.حساسیت سیستم بینایی انسان به نحوی است که اگر تصویر متحرکی را با نرخ بیشتر از ۲۵ فریم بر ثانیه مشاهده نماید آن را پیوسته می‌بیند. از این نرخ پایین‌تر تصویر منقطع دیده می‌شود.

- تفکیک پذیری بیت (bit resolution)

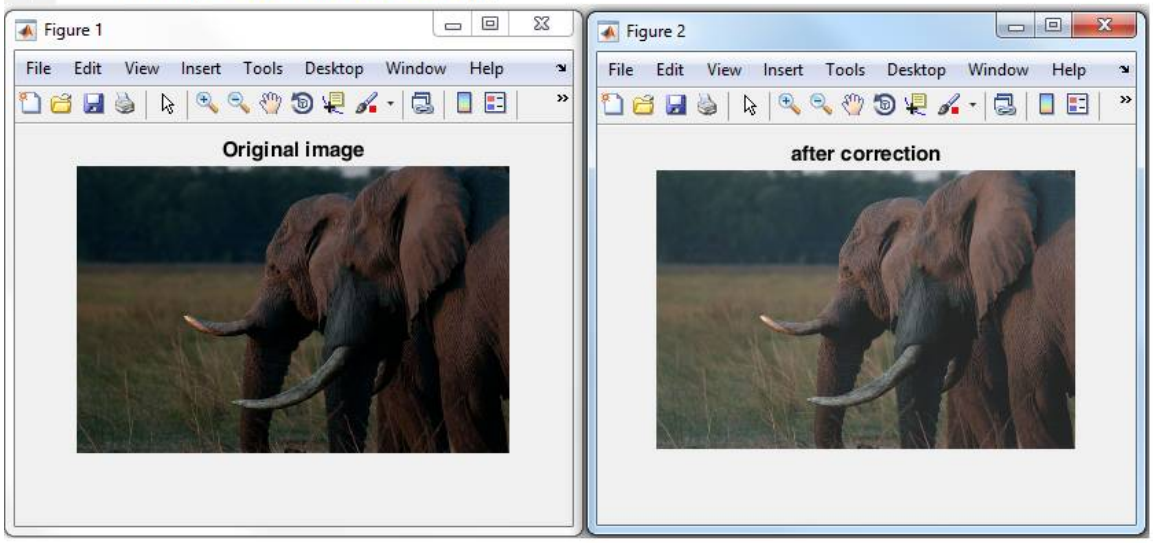
تعداد بیت‌های مورد نیاز برای ذخیره‌سازی اطلاعات تصویری یک پیکسل «*تفکیک‌پذیری بیت*» نامیده می‌شود. بطور مثال در تصاویر مقیاس خاکستری (grayscale) بطور معمول برای نگهداری اطلاعات روشنایی یک بیت عددی بین ۰ تا ۲۵۵ ذخیره می گردد. لذا برای ذخیره‌سازی آن به یک بایت یا هشت بیت نیاز است. بنابراین تفکیک پذیری بیت آن ۸ می باشد در خصوص تصاویر رنگی RGB که هر پیکسل سه مولفه رنگ دارد و هر مولفه با یک بایت نگهداری می‌گردد تفکیک پذیری آن ۲۴ = ۳ x ۸ بیت است. در برخی از منابع از واژه color depth یا عمق رنگ استفاده می‌گردد و در برخی نمادگذاری‌ها آن را با bpp که مخفف (bits per pixel) است نشان می دهند. در سایر منابع از radiometric resolution نیز استفاده شده است.

توابع پردازش تصویر در متلب : تعدادی از توابع مقدماتی برای پردازش تصویر در کد زیر نشان داده شده است.

```
A=imread('test.png');  بارگزاری تصویر
figure,imshow(A);  نمایش تصویر

B = imadjust(A,stretchlim(A)); اصلاح کنتراست

G  = rgb2gray(B); تبدیل تصویر رنگی به مقیاس خاکستری

BW = im2bw(G,graythresh(G)); تبدیل تصویر مقیاس خاکستری به سیاه و سفید بر اساس آستانه مشخص

BW = ~BW;  معکوس کردن (نگاتیو کردن) تصویر سیاه و سفید

BW = imfill(BW,'holes');          پر کردن حفره ها
BW = bwareaopen(BW, 30); پر کردن نواحی باز
```

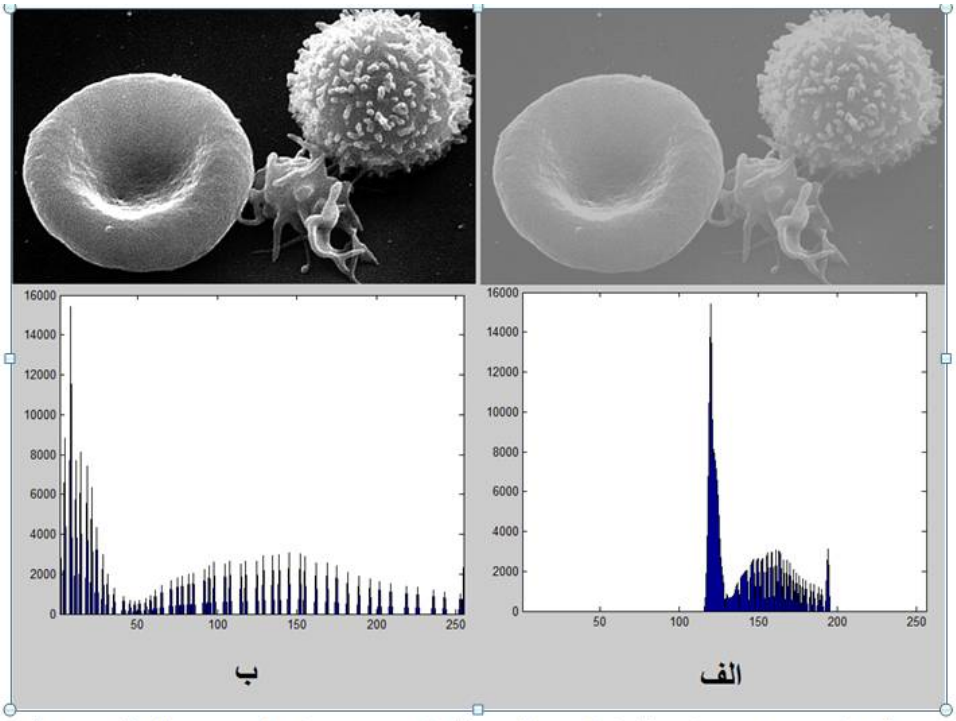
اصلاح نوردهی : اگر تصویر دارای نور زیاد باشد اصطلاحا overexposed شده یا اگر نور آن بسیار کم باشد underexposed شده است. برای تغییر نور می توان آفست مشخصی را به تصویر اضافه یا از آن کم کرد.

```
close all
clear all
A=imread('elephant.jpg');
figure,imshow(A);
title('Original image');
A = A + 35;
figure,imshow(A);
title('after correction');
```



اصلاح کنتراست : اگر تصویر دارای کنتراست نادرست باشد هیستوگرام آن در یک فضای کوچک فشرده شده است با گسترش کنتراست می‌توان آن را اصلاح کرد تا جزئیات به راحتی دیده شود.

```
close all
clear all
A=imread('bloodcell.jpg');
figure,imshow(A);
title('Original image');
B = imadjust(A,stretchlim(A));
figure,imshow(B);
title('after correction');
```



شکل (۱-۱۶) : تصویر (الف) از سمت چپ به راست گلوبول قرمز، پلاکت و گلوبول سفید خون را نشان می‌دهد. بافتنگار تصویر (الف) دارای گستردگی محدودی بین ۱۲۰ تا ۱۹۰ است پس از اعمال الگوریتم انبساط، تصویر (ب) بدست می‌آید که بافتنگار آن در گستره وسیعتری پراکنده شده است. تصویر (ب) دارای کنتراست بیشتری است و جزئیات آن راحت‌تر دیده می‌شود.

تغییر سایز تصاویر : می‌توان به راحتی سایز تصاویر را در متلب تغییر داد مطابق کد زیر.