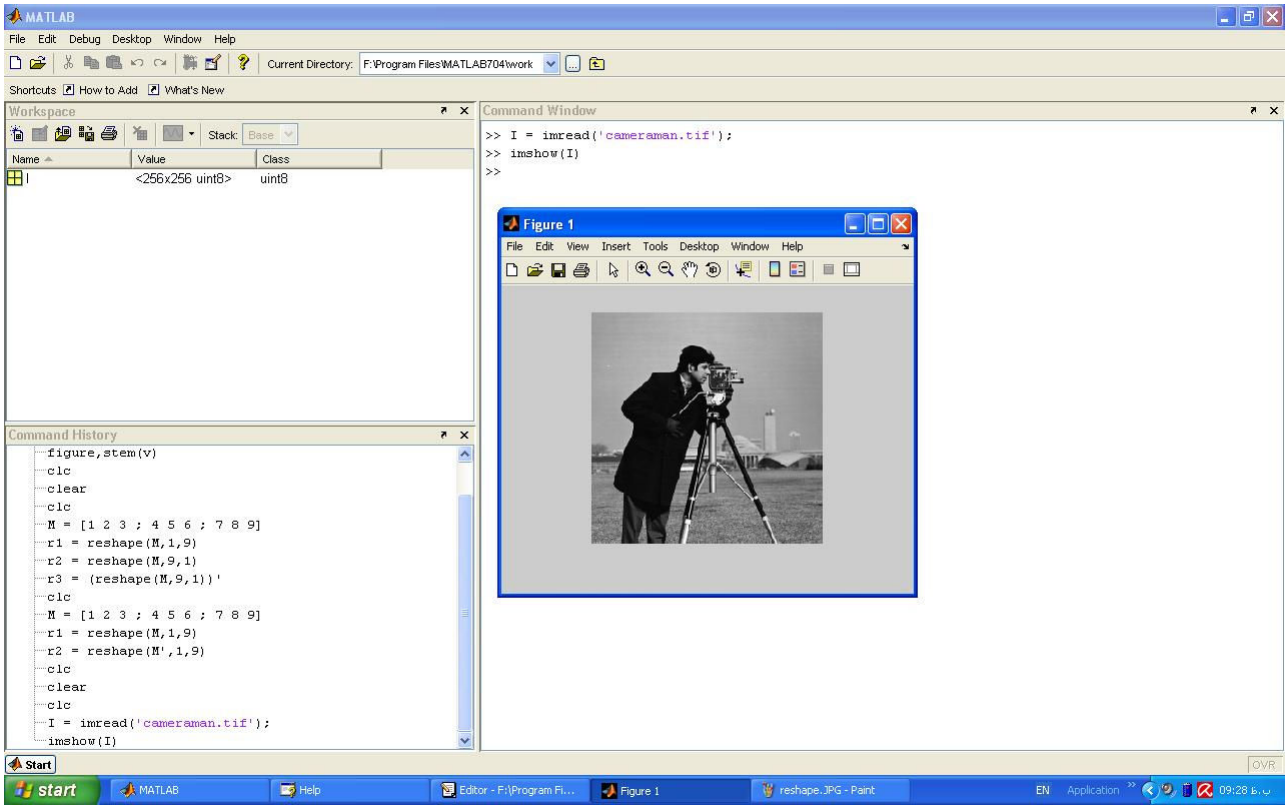
بخش دوم – کاربرد *MATLAB* در پردازش تصویر

پردازش سیگنال تصویر یکی از مهمترین قابلیت‌های نرم‌افزار MATLAB است. جعبه‌ابزار Image Processing امکانات فراوانی برای پردازش تصاویر سیاه و سفید و رنگی در اختیار ما می‌گذارد. در ادامه به توابع پایه برای پردازش تصاویر سیاه و سفید خواهیم پرداخت.

سیگنال تصویر

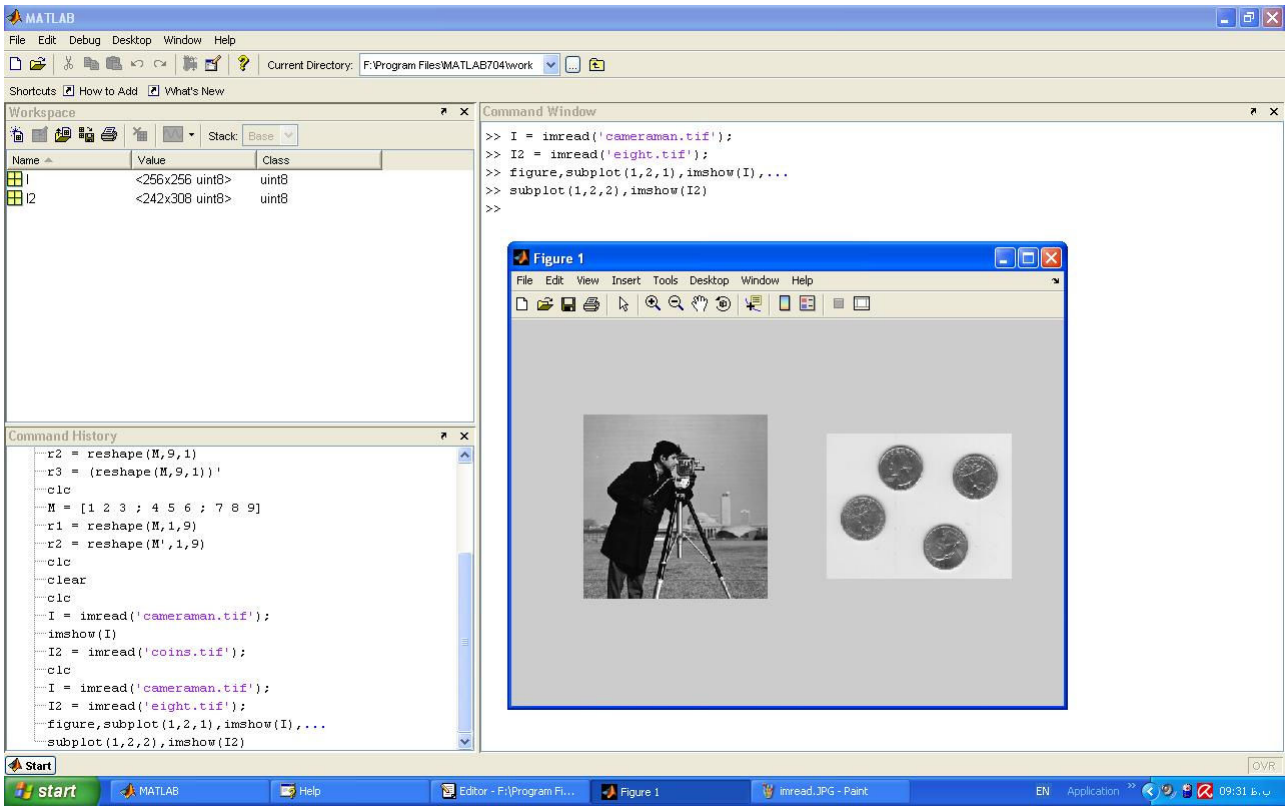
در یک تصویر سیاه و سفید، هر نقطه یا پیکسل تصویر(Pixel) یک سطح روشنایی بین صفر تا ۲۵۵ دارد که سطح صفر مربوط به رنگ سیاه و سطح ۲۵۵ مربوط به رنگ سفید است. بنابراین می‌توان سطح روشنایی هر پیکسل تصویر را با یک عدد ۸ بیتی نمایش داد.

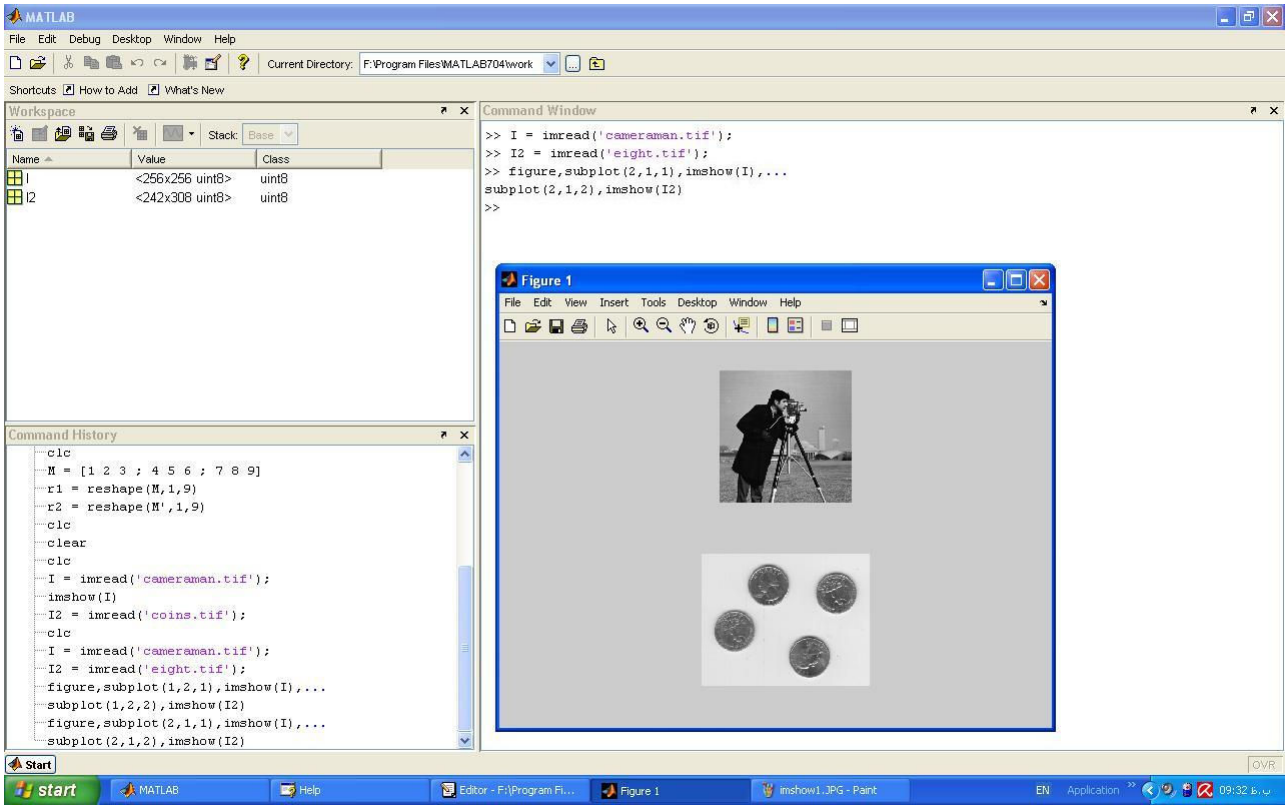
یک تصویر سیاه و سفید با اندازه m×n (مثلاً ۴۸۰×۶۴۰)، در واقع یک ماتریس از پیکسلها با m سطر و n ستون است که مقدار هر خانه ماتریس (یک بایت) نشان‌دهنده سطح روشنایی آن پیکسل تصویر می‌باشد. به کمک دستور imread (image read) می‌توان یک تصویر را به صورت یک ماتریس وارد MATLAB کرد. تابع imshow این ماتریس را به صورت تصویر نمایش می‌دهد:

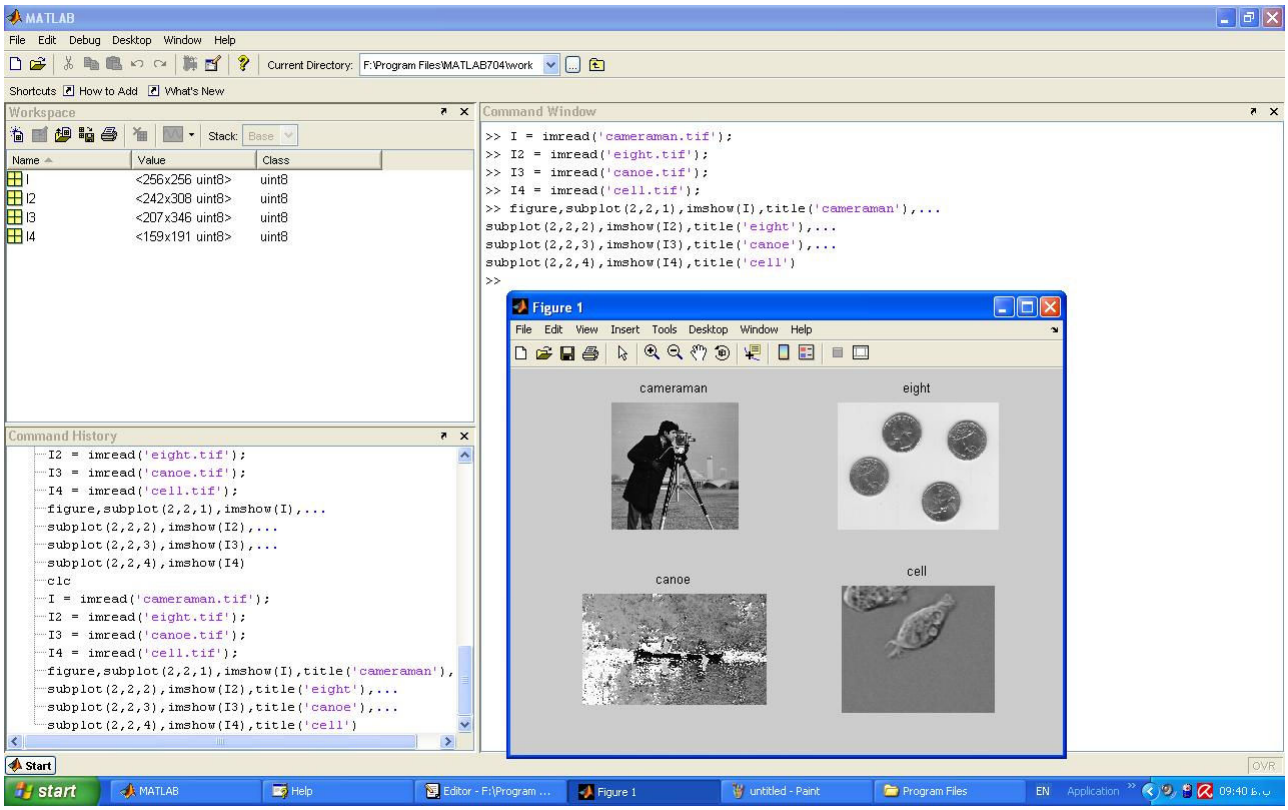


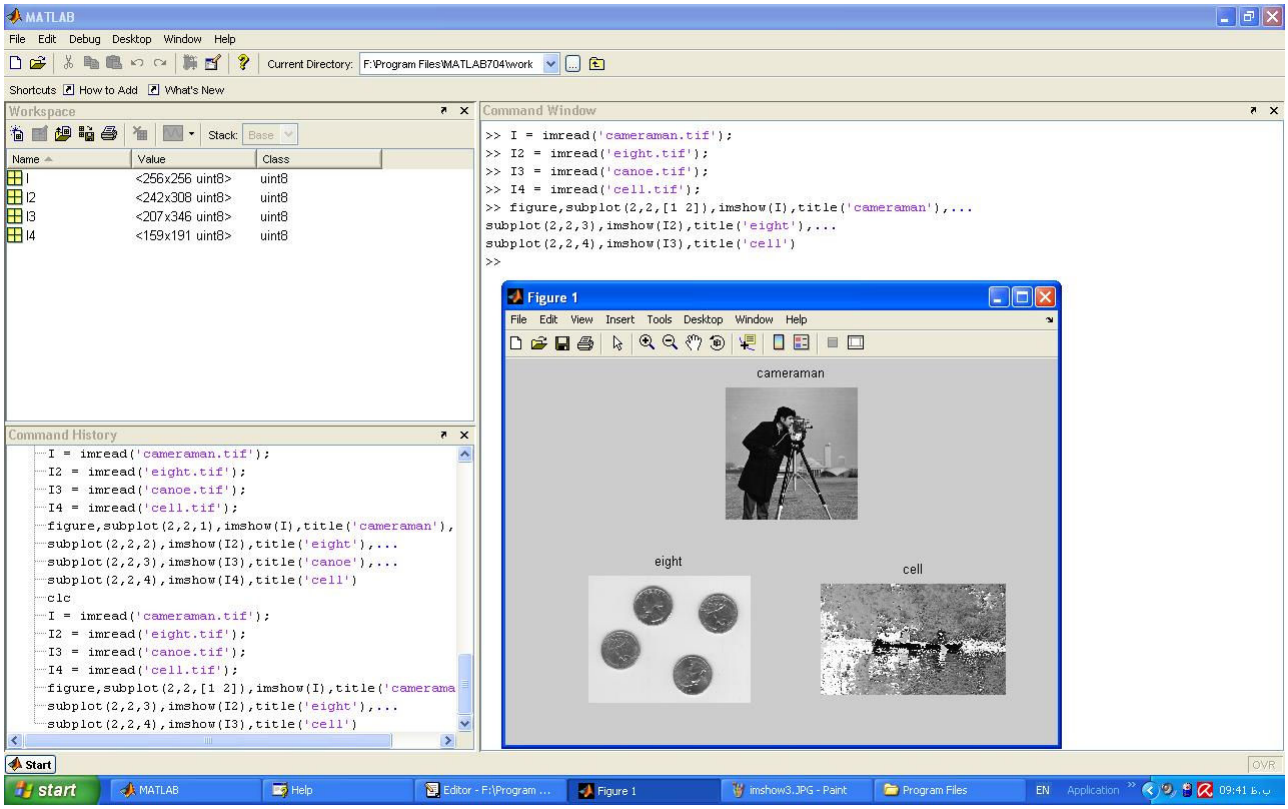
عناصر یک تصویر می‌تواند از نوع uint8 (یک بایت که می‌تواند بین صفر تا ۲۵۵ باشد) یا از نوع double (عدد حقیقی بین صفر تا یک) باشد.

به کمک تابع subplot می‌توان چند ترسیم را روی یک شکل انجام داد:


MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
I <256x256 uint8> uint8
I2 <242x308 uint8> uint8
Command Window
>> I = imread('cameraman.tif');
>> I2 = imread('eight.tif');
>> figure,subplot(1,2,1),imshow(I),...
>> subplot(1,2,2),imshow(I2)
>>
Figure 1
File Edit View Insert Tools Desktop Window Help
Command History
r2 = reshape(M,9,1)
r3 = (reshape(M,9,1))'
clc
M = [1 2 3 ; 4 5 6 ; 7 8 9]
r1 = reshape(M,1,9)
r2 = reshape(M',1,9)
clc
clear
clc
I = imread('cameraman.tif');
imshow(I)
I2 = imread('coins.tif');
clc
I = imread('cameraman.tif');
I2 = imread('eight.tif');
figure,subplot(1,2,1),imshow(I),...
subplot(1,2,2),imshow(I2)
Start
start
MATLAB
Help
Editor - F:\Program Fi...
Figure 1
imread.JPG - Paint
EN
Application
09:31


MATLAB
File Edit Debug Desktop Window Help
Current Directory: F:\Program Files\MATLAB704\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
I <256x256 uint8> uint8
I2 <242x308 uint8> uint8
Command Window
>> I = imread('cameraman.tif');
>> I2 = imread('eight.tif');
>> figure,subplot(2,1,1),imshow(I),...
subplot(2,1,2),imshow(I2)
>>
Figure 1
File Edit View Insert Tools Desktop Window Help
Command History
clc
M = [1 2 3 ; 4 5 6 ; 7 8 9]
r1 = reshape(M,1,9)
r2 = reshape(M',1,9)
clc
clear
clc
I = imread('cameraman.tif');
imshow(I)
I2 = imread('coins.tif');
clc
I = imread('cameraman.tif');
I2 = imread('eight.tif');
figure,subplot(1,2,1),imshow(I),...
subplot(1,2,2),imshow(I2)
figure,subplot(2,1,1),imshow(I),...
subplot(2,1,2),imshow(I2)
Start
start
MATLAB
Help
Editor - F:\Program Fi...
Figure 1
imshow1.JPG - Paint
EN
Application
09:32





پردازش سیگنالها و سیستم‌های دوبعدی نیاز به معلومات بیشتری دارد. بنابراین به کمک تابع reshape تصویر را به یک سیگنال یک‌بعدی تبدیل می‌کنیم و پس از پردازش به کمک همین تابع دوباره تصویر را به اندازه اصلی برمی‌گردانیم.

دستور conv

به کمک این دستور می‌توان یک فیلتر را روی یک سیگنال (مثلاً یک تصویر) اعمال کرد. این کار به معنی کانوالو کردن (convolution) سیگنال تصویر با پاسخ ضربه فیلتر است. قطعه کد زیر یک تصویر ۲۵۶×۲۵۶ به نام cameraman.tif که همراه با MATLAB ارائه می‌شود را خوانده و یک فیلتر متوسط‌گیر با پاسخ ضربه $h=[\frac{1}{3},\frac{1}{3},\frac{1}{3}]$ را روی آن اعمال کرده و نتیجه را نمایش می‌دهد. به نحوه reshape کردن توجه کنید:

```
Im = imread ( ' cameraman.tif ' );
I = im2double ( Im );

% size of this image is 256*256
Vector = reshape ( I , 1 , 256*256 ); %Image as a vector

h1 = [1/3 1/3 1/3]; % or "h1 = ones(1,3)/3"

Filt = conv ( Vector , h1 );

F = Filt ( 2 : end-1 );

Out = reshape ( F , 256 , 256 );  vector as image

imshow ( Out );
```

چون دستور conv فقط می‌تواند روی داده‌های از نوع double عمل کند، به کمک تابع im2double تصویر خوانده شده را به این نوع داده تبدیل می‌کنیم.

کانوالو کردن یک فیلتر با پاسخ ضربه به طول n، باعث می‌شود n-1 واحد به نتیجه کانولوشن اضافه شود. بنابراین برای اینکه اندازه تصویر خروجی با تصویر اصلی یکی باشد، به کمک دستور F = Filt (2 : end-1) یک پیکسل از ابتدا و یک پیکسل از انتهای بردار خروجی کم می‌کنیم.

تابع reshape اول برای تبدیل تصویر دوبعدی به یک بردار یک‌بعدی و reshape دوم برای تبدیل نتیجه پردازش که یک بردار یک‌بعدی است به ماتریس دوبعدی تصویر به کار می‌رود. با استفاده از قطعه کد بالا، ستونهای تصویر به دنبال هم چیده شده و پردازش می‌شوند. می‌توانید به جای این کار، سطرهای تصویر را به دنبال هم بچینید (به کمک آنچه قبلاً در قسمت reshape دیدیم) و پردازش کنید. این دو تکنیک را در کانولوشن تصویر با یک فیلتر بالاگذر با پاسخ ضربه $h=[-1,2,-1]$ به کار ببرید و نتایج را با هم مقایسه کنید.

پردازش تصویر در حوزه فرکانس

برای پردازش تصویر در حوزه فرکانس، ابتدا آن را به یک سیگنال یک‌بعدی تبدیل می‌کنیم و به کمک تابع fft از آن تبدیل فوریه (Fast Fourier Transform) می‌گیریم. نتیجه تابع fft یک بردار مختلط است که به کمک توابع abs و angle می‌توان مقادیر دامنه و فاز آن را به دست آورد.

قطعه کد زیر اصول پردازش سیگنال تصویر به صورت یک بعدی در فضای فرکانسی را نشان می‌دهد:

```
Im = imread('eight.tif');
I = im2double(Im);

[row col] = size(I);
% size of this image is row*col
x = reshape(I,1,row*col); %Image as a vector

% Fourier Transform
X = fft(x);
X = fftshift(X);
```

```
% Amplitude of Fourier Transform
absX = abs(X);

% Phase of Fourier Transform
angX = angle(X);
% High-pass filter
H = zeros(1,row*col);
H(1 : row*col/3) = 1;
H(end - row*col/3 : end) = 1;

absY = H .* absX;

Y = absY .* exp(j*angX);
Y = fftshift(Y);

y = ifft(Y);
y = real(y);

Out = reshape(y,row,col);

imshow(Out),title('High-pass Filter');
```

تابع fftshift مولفه‌های کم‌فرکانس فوریه را به مرکز طیف منتقل می‌کند تا تحلیل آن ساده‌تر باشد. همان‌گونه که در قطعه کد بالا می‌بینید، H اندازه پاسخ فرکانسی یک فیلتر بالاگذر ایده‌آل است که در اندازه تبدیل فوریه تصویر ضرب شده است. Y تبدیل فوریه‌ای است که از ترکیب اندازه جدید با فاز تبدیل فوریه اصلی به دست می‌آید ($Y = absY.e^{j.angX}$). با اعمال مجدد تابع fftshift مؤلفه‌های فرکانسی را به مکان اصلیشان برمی‌گردانیم و به کمک تابع ifft (Inverse FFT) تبدیل فوریه جدید را به حوزه زمان برگردانده و قسمت حقیقی آن که همان تصویر خروجی است را به کمک تابع reshape به ماتریس تصویر تبدیل می‌کنیم.

توابع جعبه‌ابزار پردازش تصویر MATLAB

Image Display

colorbar Display color bar (MATLAB function)
image Create and display image object (MATLAB function)
imagesc Scale data and display as image (MATLAB function)
immovie Make movie from multiframe indexed image
imshow Display image in a MATLAB figure window
imtool Display image in the Image Viewer
montage Display multiple image frames as rectangular montage
subimage Display multiple images in single figure
warp Display image as texture-mapped surface

Image File I/O

dicomanon Anonymize a DICOM file
dicomdict Specify which DICOM data dictionary to use
dicominfo Read metadata from a DICOM message
dicomread Read a DICOM image
dicomuid Generate DICOM unique identifier
dicomwrite Write a DICOM image
dicom-dict.txt Text file containing DICOM data dictionary
imfinfo Return information about image file (MATLAB function)
imread Read image file (MATLAB function)
imwrite Write image file (MATLAB function)

Image Types and Type Conversions
dither Convert image using dithering
double Convert data to double precision (MATLAB function)
gray2ind Convert intensity image to indexed image
grayslice Create indexed image from intensity image by thresholding
graythresh Compute global image threshold using Otsu's method
im2bw Convert image to binary image by thresholding
im2double Convert image array to double precision
im2int16 Convert image array to 16-bit signed integer
im2java Convert image to instance of Java image object (MATLAB function)
im2java2d Convert image to instance of Java buffered image object
im2single Convert image array to single precision
im2uint16 Convert image array to 16-bit unsigned integers
im2uint8 Convert image array to 8-bit unsigned integers
ind2gray Convert indexed image to intensity image
ind2rgb Convert indexed image to RGB image
int16 Convert data to signed 16-bit integers (MATLAB function)
label2rgb Convert a label matrix to an RGB image
mat2gray Convert matrix to intensity image
rgb2gray Convert RGB image or colormap to grayscale
rgb2ind Convert RGB image to indexed image
uint16 Convert data to unsigned 16-bit integers (MATLAB function)
uint8 Convert data to unsigned 8-bit integers (MATLAB function)

Modular Tool Creation Functions
imageinfo Image Information tool
imcontrast Adjust Contrast tool
imdisplayrange Display range tool
impixelinfo Pixel Information tool
impixelinfoval Pixel Information tool, without text label
impixelregion Pixel Region tool
impixelregionpanel Pixel Region scroll panel

Navigational tools
immagbox Image Information tool
imoverview Adjust Contrast tool
imoverviewpanel Display range tool
imscrollpanel Pixel Information tool

Utility Functions
axes2pix Convert axes coordinate to pixel coordinate
getimage Get image data from axes
getimagemodel Retrieve imagemodel objects from image handles
imattributes Return information about image attributes
imgca Get handle to current image axes
imgcf Get handle to current image figure
imgetfile Image Open File dialog box
imhandles Get all image handles
impositionrect Create position rectangle
iptaddcallback Add function handle to callback list
iptcheckhandle Check validity of handle
iptgetapi Get Application Programmer Interface from a handle
ipticondir Directories containing IPT and MATLAB icons
iptremovecallback Delete function handle from callback list
iptwindowalign Align figure windows
truesize Adjust display size of image

Spatial Transformations
checkerboard Create checkerboard image
findbounds Find output bounds for spatial transformation
fliptform Flip the input and output roles of a TFORM structure
imcrop Crop image

imresize Resize image
imrotate Rotate image
imtransform Apply 2-D spatial transformation to image
makeresampler Create resampling structure
maketform Create geometric transformation structure
tformarray Geometric transformation of a multidimensional array
tformfwd Apply forward geometric transformation
tforminv Apply inverse geometric transformation

Image Registration
cp2tform Infer geometric transformation from control point pairs
cpcorr Tune control point locations using cross-correlation
cpselect Control point selection tool
cpstruct2pairs Convert CPSTRUCT to valid pairs of control points
normxcorr2 Normalized two-dimensional cross-correlation

Image Analysis
bwboundaries Trace region boundaries in binary image
bwtraceboundary Trace object in binary image
edge Find edges in intensity image
hough Hough transform
houghlines Extract line segments based on the Hough transform
houghpeaks Identify peaks in the Hough transform
qtdecomp Perform quadtree decomposition
qtgetblk Get block values in quadtree decomposition
qtsetblk Set block values in quadtree decomposition

Texture Analysis
entropy Entropy of an intensity image
entropyfilt Local entropy of an intensity image
graycomatrix Gray-level co-occurrence matrix
graycoprops Properties of a gray-level co-occurrence matrix
rangefilt Local range of an image
stdfilt Local standard deviation of an image

Pixel Values and Statistics
corr2 Compute 2-D correlation coefficient
imcontour Create contour plot of image data
imhist Display histogram of image data
impixel Determine pixel color values
improfile Compute pixel-value cross-sections along line segments
mean2 Compute mean of matrix elements
pixval Display information about image pixels
regionprops Measure properties of image regions
std2 Compute standard deviation of matrix elements

Image Arithmetic
imabsdiff Compute absolute difference of two images
imadd Add two images, or add constant to image
imcomplement Complement image
imdivide Divide two images, or divide image by constant
imlincomb Compute linear combination of images
immultiply Multiply two images, or multiply image by constant
imsubtract Subtract two images, or subtract constant from image

Image Enhancement
adapthisteq Perform adaptive histogram equalization using CLAHE
decorrstretch Apply a decorrelation stretch to a multi-channel image
histeq Enhance contrast using histogram equalization
imadjust Adjust image intensity values or colormap
imnoise Add noise to an image
intlut Compute new array values based on lookup table (LUT)

medfilt2 Perform 2-D median filtering
ordfilt2 Perform 2-D order-statistic filtering
stretchlim Return a pair of intensities that can be used to increase the contrast of an image
wiener2 Perform 2-D adaptive noise-removal filtering

Image Restoration (Deblurring)
deconvblind Restore image using blind deconvolution
deconvlucy Restore image using accelerated Richardson-Lucy algorithm
deconvreg Restore image using regularized filter
deconvwnr Restore image using Wiener filter
edgetaper Taper the discontinuities along the image edges
otf2psf Convert optical transfer function to point spread function
psf2otf Convert point spread function to optical transfer function

Linear Filtering
conv2 Perform 2-D convolution (MATLAB function)
convmtx2 Compute 2-D convolution matrix
convn Perform N-D convolution (MATLAB function)
filter2 Perform 2-D filtering (MATLAB function)
fspecial Create predefined filters
imfilter Multidimensional image filtering

Linear 2-D Filter Design
freqspace Determine 2-D frequency response spacing (MATLAB function.)
freqz2 Compute 2-D frequency response
fsamp2 Design 2-D FIR filter using frequency sampling
ftrans2 Design 2-D FIR filter using frequency transformation
fwind1 Design 2-D FIR filter using 1-D window method
fwind2 Design 2-D FIR filter using 2-D window method

Image Transforms
dct2 Compute 2-D discrete cosine transform
dctmtx Compute discrete cosine transform matrix
fan2para Convert fan-beam projection data to parallel-beam
fanbeam Compute fan-beam transform
fft2 Compute 2-D fast Fourier transform (MATLAB function)
fftn Compute N-D fast Fourier transform (MATLAB function)
fftshift Reverse quadrants of output of FFT (MATLAB function)
idct2 Compute 2-D inverse discrete cosine transform
ifft2 Compute 2-D inverse fast Fourier transform (MATLAB function)
ifftn Compute N-D inverse fast Fourier transform (MATLAB function)
ifanbeam Compute inverse fan-beam transform
iradon Compute inverse Radon transform
para2fan Convert parallel-beam projections to fan-beam
phantom Generate a head phantom image
radon Compute Radon transform

Intensity and Binary Images
conndef Default connectivity array
imbothat Perform bottom-hat filtering
imclearborder Suppress light structures connected to image border
imclose Close image
imdilate Dilate image
imerode Erode image
imextendedmax Find extended-maxima transform
imextendedmin Find extended-minima transform
imfill Fill image regions
imhmax Calculate H-maxima transform
imhmin Calculate H-minima transform
imimposemin Impose minima
imopen Open image
imreconstruct Perform morphological reconstruction

imregionalmax Find regional maxima of image
imregionalmin Find regional minima of image
imtophat Perform tophat filtering
watershed Find image watershed regions

Binary Images
applylut Perform neighborhood operations using lookup tables
bwarea Area of objects in binary image
bwareaopen
bwdist Distance transform
bweuler Euler number of binary image
bwhitmiss Binary hit-and-miss operation
bwlabel Label connected components in 2-D binary image
bwlabeln Label connected components in N-D binary image
bwmorph Perform morphological operations on binary image
bwpack Pack binary image
bwperim Find perimeter of objects in binary image
bwselect Select objects in binary image
bwulterode Ultimate erosion
bwunpack Unpack a packed binary image
imregionalmin Regional minima of image
imtophat Perform tophat filtering
makelut Construct lookup table for use with applylut

Structuring Element (STREL) Creation and Manipulation
getheight Get the height of a structuring element
getneighbors Get structuring element neighbor locations and heights
getnhood Get structuring element neighborhood
getsequence Extract sequence of decomposed structuring elements
isflat Return true for flat structuring element
reflect Reflect structuring element
strel Create morphological structuring element
translate Translate structuring element

Region-Based Processing
poly2mask Convert region-of-interest polygon to mask
roicolor Select region of interest, based on color
roifill Smoothly interpolate within arbitrary region
roifilt2 Filter a region of interest
roipoly Select polygonal region of interest

Neighborhood and Block Processing
bestblk Choose block size for block processing
blkproc Implement distinct block processing for image
col2im Rearrange matrix columns into blocks
colfilt Perform neighborhood operations using column-wise functions
im2col Rearrange image blocks into columns
nlfilter Perform general sliding-neighborhood operations

Colormap Manipulation
brighten Brighten or darken colormap (MATLAB function)
cmpermute Rearrange colors in colormap
cmunique Find unique colormap colors and corresponding image
colormap Set or get color lookup table (MATLAB function)
imapprox Approximate indexed image by one with fewer colors
rgbplot Plot RGB colormap components (MATLAB function)

Color Space Conversions
applycform Apply device-independent color space transformation
hsv2rgb Convert HSV values to RGB color space (MATLAB function)
iccread Read ICC color profile
iccwrite Write ICC color profile

isicc Determine if ICC color profile is valid
lab2double Convert color values to double
lab2uint16 Convert color values to uint16
lab2uint8 Convert color values to uint8
makecform Create device-independent color space transform structure
ntsc2rgb Convert NTSC values to RGB color space
rgb2hsv Convert RGB values to HSV color space (MATLAB function)
rgb2ntsc Convert RGB values to NTSC color space
rgb2ycbcr Convert RGB values to YCbCr color space
whitepoint Returns XYZ values of standard illuminants
xyz2double Convert XYZ color values to double
xyz2uint16 Convert XYZ color values to uint16
ycbcr2rgb Convert YCbCr values to RGB color space

Toolbox Preferences
iptgetpref Get value of Image Processing Toolbox preference
iptsetpref Set value of Image Processing Toolbox preference

Toolbox Utility Functions
getrangefromclass Get dynamic range of image based on its class
iptcheckconn Check validity of connectivity argument
iptcheckinput Check validity of array
iptcheckmap Check validity of colormap
iptchecknargin Check number of input arguments
iptcheckstrs Check validity of strings
iptnum2ordinal Convert positive integer to ordinal string

Interactive Mouse Utility Functions
getline Select polyline with mouse
getpts Select points with mouse
getrect Select rectangle with mouse

Array Operations
circshift Shift array circularly (MATLAB function)
padarray Pad an array

Demos
iptdemos Display index of Image Processing Toolbox demos

Performance
ippl Check for presence of Intel Performance Primitives Library (IPPL)