

## Problem 1 Report

Part 1 of the homework creates hybrid images by manipulating the phase and magnitude components of the image. In order to implement a hybrid image, the pictures are first transformed into the frequency domain, manipulated, and then converted back into the spatial domain.

The problem is split into three separate programs focusing completing the task:

- I. Hybrid images
  - a. Fish magnitude with the motorcycle phase
  - b. Motorcycle magnitude with the fish phase
- II. Neutralizing the magnitude component
- III. Removing the phase component

Each sections code models this technique and is split into four main sections: Image load, Frequency domain, Reconstruction, and Display.

### Image load

Image load prepares the images for processing by loading the image into the workspace, resizing it, and then converting the image into grayscale.

### Frequency domain

The frequency domain section is responsible for converting the image grayscale matrix into the frequency domain, and then separating the magnitude and phase of each value. To transform the image into the frequency domain the code uses the command `fft2` and then shifts the zero-frequency component to center of spectrum using `fftshift`. From the frequency domain the magnitude and phase can then be calculated from the complex form  $z = a + bi$ .

The magnitude is calculated by taking the absolute value of the frequency domain and the phase is calculated from the complex portion using the code `exp(1i*Angle(Frequency_Domain))`.

For part ii, to neutralize the magnitude the, the magnitude of the image is set to 1 and in part iii, to remove the phase, the phase the frequency complex portion is set to 0 by the equation `exp(1i*0)`.

### Reconstruction

Once the image is split into magnitude and phase the components can be manipulated and then reconstructed back into the spatial domain. To manipulate the data, the magnitude and phase matrices where multiplied together according to the hybrid image instructions. Once the data is properly manipulated the zero-frequency shift is inverted using `ifftshift`, and then converted back into the spatial domain using `ifft2`. The data is then converted into a logarithmic scale to better visualize the spatial domain.

### Display

The display sections takes the reconverted image and displayed all the required images using subplots in order to visualize the output.

### Runtime Analysis

After running timing analysis for each problem, the manipulation and reconstruction portion of the code was found to be consistently the longest. This confirms our initial guess in that the reconstruction section not only has to perform the matrix multiplication for the images in addition to converting the hybrid image back into the spatial domain. This hunch is also confirmed after using the profiler to complete runtime analysis. Both parts ii and ii had the ifft2 command with a longer run time than fft2. The results of fft2 and ifft2 can be seen in the table below.

Command	Part i	Part ii	Part iii
fft2	0.051 s	0.041 s	0.057 s
ifft2	0.050 s	0.071 s	0.061 s

The runtime analysis and timing for the rest of the code can be viewed in the attachments.

### Runtime Improvement

In order to improve the critical portion of our code we could implement techniques such as reducing the size of our images or reducing the unnecessary data in the frequency domain similar to wavelets. Since you can cut out large portion of the frequency domain and still retain a good output image, this would allow us to speed up the time of ifft2 and shifting back into the spatial domain.

---

## Table of Contents

.....	1
Images .....	1
Fequency Domain .....	1
Reconstructin .....	2
Display reconstructed images .....	2

```
% Create a Hybrid Image between the (fish) as input#1 and the
(motorcycle) as input#2.
% 1. First hybrid is the (fish) magnitude with the (motorcycle) phase
% 2. Second hybrid is the (motorcycle) magnitude and the (fish) phase
clear; close all; clc;
```

## Images

```
disp('Open Images');
tic;
im1 = imread('./data/motorcycle.bmp');
im1 = imresize(im1,[360, 410]);
im1 = double(im1)/255;
im1 = rgb2gray(im1);
toc;
tic;
im2 = imread('./data/fish.bmp');
im2 = imresize(im2,[360, 410]);
im2 = double(im2)/255;
im2 = rgb2gray(im2);
toc;
disp('End open images');
```

```
Open Images
Elapsed time is 0.010452 seconds.
Elapsed time is 0.011241 seconds.
End open images
```

## Fequency Domain

```
disp(' ');
tic;
disp('fftshifts');
F1 = fftshift(fft2(im1));
F2 = fftshift(fft2(im2));
toc;

disp(' ');
tic;
disp('Neutralize Magnitude');
```

---

```

% Neutralize Magnitude
F1_Mag = abs(F1);
F2_Mag = abs(F2);

toc;
disp(' ');

tic;
disp('Phase');
% Phase
F1_Phase = exp(1i*angle(F1));
F2_Phase = exp(1i*angle(F2));
toc;
disp(' ');

fftshifts
Elapsed time is 0.008460 seconds.

Neutralize Magnitude
Elapsed time is 0.002088 seconds.

Phase
Elapsed time is 0.007451 seconds.

```

## Reconstructin

```

tic;
disp('Reconstruct');
Reconstruct1 = log(abs(iff2(iff2shift(F2_Mag.*F1_Phase)))+1);
Reconstruct2 = log(abs(iff2(iff2shift(F1_Mag.*F2_Phase)))+1);
toc;
disp(' ');

Reconstruct
Elapsed time is 0.020154 seconds.

```

## Display reconstructed images

```

tic;
disp('Display Images');
figure;
subplot(1,2,1), imagesc(Reconstruct1), colormap gray, axis off,
    title({'Fish magnitude with', 'motorcycle phase'});
subplot(1,2,2), imagesc(Reconstruct2), colormap gray, axis off,
    title({'Motorcycle magnitude', 'with fish phase'});
toc;

Display Images
Elapsed time is 0.084241 seconds.

```

---

**Fish magnitude with  
motorcycle phase**



**Motorcycle magnitude  
with fish phase**



*Published with MATLAB® R2017a*

---

## Table of Contents

.....	1
Images .....	1
Fequency Domain .....	1
Reconstructin .....	2
Display reconstructed images .....	3

```
% Create and show the new image resulting from neutralizing the
    magnitude
% components for each the following input images separately: (dog),
    (Einstein), and
% fish).
clear; close all; clc;
```

## Images

```
disp('Display Images');
tic;
im1 = imread('./data/dog.bmp');
im1 = imresize(im1,[360, 410]);
im1 = double(im1)/255;
im1 = rgb2gray(im1);
toc;
im2 = imread('./data/einstein.bmp');
im2 = imresize(im2,[360, 410]);
im2 = double(im2)/255;
im2 = rgb2gray(im2);
toc;
```

```
im3 = imread('./data/fish.bmp');
im3 = imresize(im3,[360, 410]);
im3 = double(im3)/255;
im3 = rgb2gray(im3);
toc;
disp(' ');
```

```
Display Images
Elapsed time is 0.016428 seconds.
Elapsed time is 0.024544 seconds.
Elapsed time is 0.033712 seconds.
```

## Fequency Domain

```
tic;
```

---

```

disp('fftshifts');
F1 = fftshift(fft2(im1));
F2 = fftshift(fft2(im2));
F3 = fftshift(fft2(im3));
toc;
disp(' ');
% Neutralize Magnitude
tic;
disp('Neutralize Magnitudes');
F1_Mag = 1;
F2_Mag = 1;
F3_Mag = 1;
toc;
disp(' ');

tic;
disp('Phases');
% Phase
F1_Phase = exp(1i*angle(F1));
F2_Phase = exp(1i*angle(F2));
F3_Phase = exp(1i*angle(F3));

toc;

disp(' ');

fftshifts
Elapsed time is 0.011513 seconds.

Neutralize Magnitudes
Elapsed time is 0.000113 seconds.

Phases
Elapsed time is 0.008760 seconds.

```

## Reconstructin

```

tic;
disp('Recontruction');
Reconstruct1 = log(abs(ifft2(ifftshift(F1_Mag.*F1_Phase)))+1);
Reconstruct2 = log(abs(ifft2(ifftshift(F2_Mag.*F2_Phase)))+1);
Reconstruct3 = log(abs(ifft2(ifftshift(F3_Mag.*F3_Phase)))+1);

toc;

disp(' ');

Recontruction
Elapsed time is 0.025091 seconds.

```

---

---

## Display reconstructed images

```
tic;
disp('Display Images');
figure;
subplot (2,2,1), imagesc(Reconstruct1), colormap gray, axis off,
    title('Dog Magnitude Neutralize');
toc;
subplot (2,2,2), imagesc(Reconstruct2), colormap gray, axis off,
    title('Einstein Magnitude Neutralize');
toc;
subplot (2,2,3), imagesc(Reconstruct3), colormap gray, axis off,
    title('Fish Magnitude Neutralize');
toc;
```

*Display Images*

*Elapsed time is 0.073588 seconds.*

*Elapsed time is 0.093497 seconds.*

*Elapsed time is 0.111950 seconds.*

**Dog Magnitude Neutralize**



**Einstein Magnitude Neutralize**



**Fish Magnitude Neutralize**



*Published with MATLAB® R2017a*



---

## Table of Contents

.....	1
Images .....	1
Frequency Domain .....	1
Neutralize Phase .....	2
Reconstructin .....	2
Display reconstructed images .....	2

```
% Create and show the new image resulting from removing the phase
% components for
% each the following input images separately: (dog), (motorcycle), and
% (submarine).
clear; close all; clc;
```

## Images

```
tic;
disp('Display images');
im1 = imread('./data/dog.bmp');
im1 = imresize(im1,[360, 410]);
im1 = double(im1)/255;
im1 = rgb2gray(im1);
toc;

im2 = imread('./data/motorcycle.bmp');
im2 = imresize(im2,[360, 410]);
im2 = double(im2)/255;
im2 = rgb2gray(im2);
toc;

im3 = imread('./data/submarine.bmp');
im3 = imresize(im3,[360, 410]);
im3 = double(im3)/255;
im3 = rgb2gray(im3);
toc;
disp(' ');

Display images
Elapsed time is 0.011072 seconds.
Elapsed time is 0.020707 seconds.
Elapsed time is 0.042132 seconds.
```

## Frequency Domain

```
tic;
disp('Frequency Domain');
F1 = fftshift(fft2(double(im1)));
```

---

```
F2 = fftshift(fft2(double(im2)));
F3 = fftshift(fft2(double(im3)));
toc;
disp(' ');
```

```
% Magnitude
tic;
disp('Magnitude');
F1_Mag = abs(F1);
F2_Mag = abs(F2);
F3_Mag = abs(F3);
toc;
disp(' ');
```

```
Frequency Domain
Elapsed time is 0.012149 seconds.
```

```
Magnitude
Elapsed time is 0.002962 seconds.
```

## Neutralize Phase

```
tic;
disp('Neutralize Phase');
F1_Phase = exp(1i*0);
F2_Phase = exp(1i*0);
F3_Phase = exp(1i*0);
toc;
disp(' ');
```

```
Neutralize Phase
Elapsed time is 0.000275 seconds.
```

## Reconstructin

```
tic;
disp('Reconstruct');
Reconstruct1 = log(abs(ifft2(ifftshift(F1_Mag.*F1_Phase)))+1);
Reconstruct2 = log(abs(ifft2(ifftshift(F2_Mag.*F2_Phase)))+1);
Reconstruct3 = log(abs(ifft2(ifftshift(F3_Mag.*F3_Phase)))+1);
toc;
```

```
Reconstruct
Elapsed time is 0.022922 seconds.
```

## Display reconstructed images

```
tic;
disp('Display Final Images');
figure;
```

---

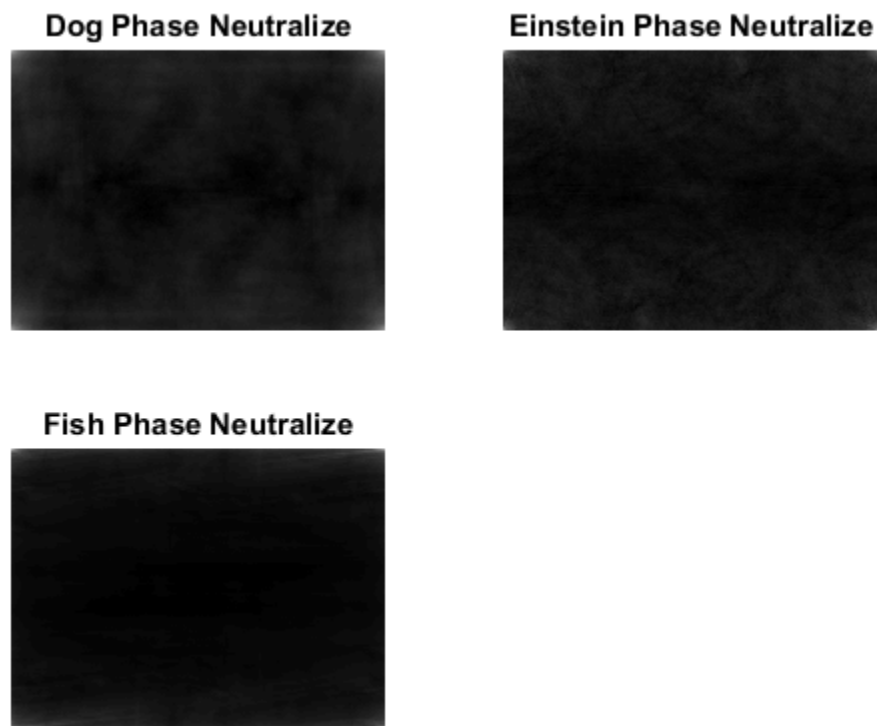
```
subplot (2,2,1), imagesc(Reconstruct1), colormap gray, axis off,  
    title('Dog Phase Neutralize');  
toc;  
subplot (2,2,2), imagesc(Reconstruct2), colormap gray, axis off,  
    title('Einstein Phase Neutralize');  
toc;  
subplot (2,2,3), imagesc(Reconstruct3), colormap gray, axis off,  
    title('Fish Phase Neutralize');  
toc;
```

*Display Final Images*

*Elapsed time is 0.061337 seconds.*

*Elapsed time is 0.077263 seconds.*

*Elapsed time is 0.094900 seconds.*



*Published with MATLAB® R2017a*

## Profile Summary

Generated 02-Mar-2018 18:11:44 using performance time.

<a href="#">Function Name</a>	<a href="#">Calls</a>	<a href="#">Total Time</a>	<a href="#">Self Time*</a>	Total Time Plot (dark band = self time)
<a href="#">part1i</a>	1	1.365 s	0.143 s	
<a href="#">close</a>	1	0.658 s	0.005 s	
<a href="#">close&gt;safegetchildren</a>	1	0.532 s	0.225 s	
<a href="#">setdiff</a>	2	0.292 s	0.024 s	
<a href="#">imread</a>	2	0.268 s	0.035 s	
<a href="#">setdiff&gt;setdiffR2012a</a>	2	0.268 s	0.183 s	
<a href="#">imread&gt;call_format_specific_reader</a>	2	0.205 s	0.007 s	
<a href="#">imagesci\private\readbmp</a>	2	0.198 s	0.004 s	
<a href="#">imagesci\private\imbmpinfo</a>	2	0.153 s	0.012 s	
<a href="#">imresize</a>	2	0.144 s	0.024 s	
<a href="#">close&gt;request_close</a>	1	0.119 s	0.011 s	
<a href="#">closereq</a>	1	0.097 s	0.090 s	
<a href="#">...\imbmpinfo&gt;initializeBMPInfoStruct</a>	2	0.091 s	0.003 s	
<a href="#">ima...\private\initializeMetadataStruct</a>	2	0.088 s	0.009 s	
<a href="#">datestr</a>	2	0.079 s	0.009 s	

<a href="#">timefun\private\dateformverify</a>	2	0.068 s	0.007 s	
<a href="#">timefun\private\formatdate</a>	2	0.061 s	0.038 s	
<a href="#">unique</a>	2	0.053 s	0.009 s	
<a href="#">fft2</a>	2	0.051 s	0.051 s	
<a href="#">ifft2</a>	2	0.050 s	0.050 s	
<a href="#">imresize&gt;parseInputs</a>	2	0.049 s	0.014 s	
<a href="#">unique&gt;uniqueR2012a</a>	2	0.044 s	0.044 s	
<a href="#">imagesci\private\imbmpinfo&gt;readBMPInfo</a>	2	0.041 s	0.007 s	
<a href="#">imagesci\private\readbmpdata</a>	2	0.041 s	0.005 s	
<a href="#">ima...ivate\readbmpdata&gt;bmpReadData24</a>	2	0.036 s	0.020 s	
<a href="#">imresize&gt;resizeAlongDim</a>	4	0.036 s	0.005 s	
<a href="#">imagesci\private\imbmpinfo&gt;readWin3xInfo</a>	2	0.034 s	0.010 s	
<a href="#">ismember</a>	3	0.034 s	0.003 s	
<a href="#">ismember&gt;ismemberR2012a</a>	3	0.031 s	0.008 s	
<a href="#">images\private\imresizemex</a> (MEX-file)	4	0.030 s	0.030 s	
<a href="#">contributions</a>	4	0.027 s	0.018 s	
<a href="#">angle</a>	2	0.024 s	0.024 s	

<a href="#">ismember&gt;ismemberClassTypes</a>	3	0.023 s	0.023 s	
<a href="#">cnv2icudf</a>	2	0.022 s	0.022 s	
<a href="#">...te\imbmpinfo&gt;readWin3xBitmapHeader</a>	2	0.015 s	0.014 s	
<a href="#">imresize&gt;fixupSizeAndScale</a>	2	0.014 s	0.008 s	
<a href="#">allchild</a>	1	0.013 s	0.006 s	
<a href="#">rgb2gray</a>	2	0.012 s	0.009 s	
<a href="#">close&gt;request_close_helper</a>	2	0.011 s	0.007 s	
<a href="#">imresize&gt;parsePreMethodArgs</a>	2	0.011 s	0.007 s	
<a href="#">fileparts</a>	2	0.010 s	0.009 s	
<a href="#">ima...ivate\readbmpdata&gt;readFromFile</a>	2	0.010 s	0.008 s	
<a href="#">fftshift</a>	2	0.009 s	0.009 s	
<a href="#">onCleanup&gt;onCleanup.delete</a>	5	0.009 s	0.006 s	
<a href="#">rot90</a>	2	0.007 s	0.007 s	
<a href="#">cubic</a>	4	0.007 s	0.007 s	
<a href="#">ifftshift</a>	2	0.006 s	0.006 s	
<a href="#">ima...ate\imbmpinfo&gt;readBMPFileHeader</a>	2	0.006 s	0.006 s	
<a href="#">imread&gt;get_full_filename</a>	2	0.005 s	0.005 s	

<a href="#">imread&gt;parse_inputs</a>	2	0.005 s	0.005 s	
<a href="#">cell.ismember</a>	2	0.004 s	0.004 s	
<a href="#">...te\imbmpinfo&gt;readVersion3xColormap</a>	2	0.004 s	0.004 s	
<a href="#">...graphics.internal.axesDestroyed(o.e)</a>	2	0.004 s	0.002 s	
<a href="#">close&gt;getEmptyHandleList</a>	3	0.003 s	0.003 s	
<a href="#">imresize&gt;fixupSize</a>	2	0.003 s	0.003 s	
<a href="#">findFirstParamString</a>	2	0.003 s	0.003 s	
<a href="#">imresize&gt;findMethodArg</a>	2	0.003 s	0.003 s	
<a href="#">deriveScaleFromSize</a>	2	0.003 s	0.003 s	
<a href="#">imresize&gt;scaleOrSize</a>	2	0.003 s	0.003 s	
<a href="#">imagesci\private\imbmpinfo&gt;postProcess</a>	2	0.003 s	0.003 s	
<a href="#">rgb2gray&gt;parse_inputs</a>	2	0.003 s	0.003 s	
<a href="#">contributions&gt;@(x)scale*kernel(scale*x)</a>	1	0.003 s	0.002 s	
<a href="#">imagesci\private\imbmpinfo&gt;getSignature</a>	2	0.003 s	0.003 s	
<a href="#">imagesci\private\getFileFromURL</a>	2	0.003 s	0.003 s	
<a href="#">imresize&gt;checkForMissingOutputArgument</a>	2	0.003 s	0.002 s	
<a href="#">imresize&gt;postprocessImage</a>	2	0.002 s	0.002 s	

<a href="#">...ger&gt;SubplotListenersManager.delete</a>	1	0.002 s	0.002 s	
<a href="#">imresize&gt;warnIfPostMethodArgs</a>	2	0.002 s	0.002 s	
<a href="#">imresize&gt;preprocessImage</a>	2	0.002 s	0.002 s	
<a href="#">onCleanup&gt;onCleanup.onCleanup</a>	5	0.002 s	0.002 s	
<a href="#">imresize&gt;parseParamValuePairs</a>	2	0.002 s	0.002 s	
<a href="#">axesDestroyed</a>	2	0.002 s	0.002 s	
<a href="#">dimensionOrder</a>	2	0.001 s	0.001 s	
<a href="#">...ddenHandles',oldUDDShowHiddenHandles)</a>	1	0.001 s	0.001 s	
<a href="#">gcbf</a>	2	0.001 s	0.001 s	
<a href="#">ispc</a>	2	0.001 s	0.001 s	
<a href="#">ima...ivate\imbmpinfo&gt;@()fclose(fid)</a>	2	0.001 s	0.001 s	
<a href="#">ima...ate\imbmpinfo&gt;decodeCompression</a>	2	0.001 s	0.001 s	
<a href="#">isPureNearestNeighborComputation</a>	8	0.001 s	0.001 s	
<a href="#">datestr&gt;getdateform</a>	2	0.001 s	0.001 s	
<a href="#">ima...vate\readbmpdata&gt;@()fclose(fid)</a>	2	0.001 s	0.001 s	
<a href="#">imresize&gt;isInputIndexed</a>	6	0.001 s	0.001 s	
<a href="#">close&gt;checkfigs</a>	2	0.001 s	0.001 s	



<a href="#">allchild&gt;getchildren</a>	1	0.001 s	0.001 s	
<a href="#">...set(rootobj,'ShowHiddenHandles'.Temp)</a>	1	0.001 s	0.001 s	
<a href="#">uitools\private\allchildRootHelper</a>	1	0.000 s	0.000 s	

**Self time** is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

## Profile Summary

Generated 02-Mar-2018 19:03:25 using performance time.

<a href="#">Function Name</a>	<a href="#">Calls</a>	<a href="#">Total Time</a>	<a href="#">Self Time*</a>	Total Time Plot (dark band = self time)
<a href="#">part1ii</a>	1	1.342 s	0.159 s	
<a href="#">close</a>	1	0.548 s	0.031 s	
<a href="#">close&gt;request_close</a>	1	0.386 s	0.023 s	
<a href="#">closereq</a>	1	0.343 s	0.327 s	
<a href="#">imread</a>	3	0.326 s	0.029 s	
<a href="#">imread&gt;call_format_specific_reader</a>	3	0.267 s	0.007 s	
<a href="#">imagesci\private\readbmp</a>	3	0.259 s	0.004 s	
<a href="#">imagesci\private\imbmpinfo</a>	3	0.225 s	0.015 s	
<a href="#">imresize</a>	3	0.139 s	0.024 s	
<a href="#">close&gt;safegetchildren</a>	1	0.126 s	0.021 s	
<a href="#">...\imbmpinfo&gt;initializeBMPInfoStruct</a>	3	0.117 s	0.004 s	
<a href="#">ima...\private\initializeMetadataStruct</a>	3	0.113 s	0.012 s	
<a href="#">datestr</a>	3	0.101 s	0.015 s	
<a href="#">timefun\private\dateformverify</a>	3	0.084 s	0.008 s	
<a href="#">timefun\private\formatdate</a>	3	0.076 s	0.044 s	

<a href="#">setdiff</a>	2	0.072 s	0.004 s	
<a href="#">ifft2</a>	3	0.071 s	0.071 s	
<a href="#">setdiff&gt;setdiffR2012a</a>	2	0.068 s	0.013 s	
<a href="#">imagesci\private\imbmpinfo&gt;readBMPInfo</a>	3	0.052 s	0.009 s	
<a href="#">ismember</a>	3	0.050 s	0.008 s	
<a href="#">imagesci\private\imbmpinfo&gt;readWin3xInfo</a>	3	0.043 s	0.010 s	
<a href="#">imresize&gt;parseInputs</a>	3	0.042 s	0.013 s	
<a href="#">ismember&gt;ismemberR2012a</a>	3	0.042 s	0.021 s	
<a href="#">fft2</a>	3	0.041 s	0.041 s	
<a href="#">imresize&gt;resizeAlongDim</a>	6	0.038 s	0.007 s	
<a href="#">imagesci\private\imbmpinfo&gt;getSignature</a>	3	0.036 s	0.036 s	
<a href="#">angle</a>	3	0.035 s	0.035 s	
<a href="#">cnv2icudf</a>	3	0.033 s	0.033 s	
<a href="#">imagesci\private\readbmpdata</a>	3	0.031 s	0.002 s	
<a href="#">allchild</a>	1	0.031 s	0.018 s	
<a href="#">images\private\imresizemex</a> (MEX-file)	5	0.029 s	0.029 s	
<a href="#">ima...ivate\readbmpdata&gt;bmpReadData24</a>	3	0.029 s	0.014 s	

<a href="#">contributions</a>	6	0.026 s	0.017 s	
<a href="#">ismember&gt;ismemberClassTypes</a>	3	0.020 s	0.020 s	
<a href="#">close&gt;request_close_helper</a>	2	0.019 s	0.006 s	
<a href="#">unique</a>	2	0.017 s	0.008 s	
<a href="#">...te\imbmpinfo&gt;readWin3xBitmapHeader</a>	3	0.017 s	0.016 s	
<a href="#">ima...ate\imbmpinfo&gt;readBMPFileHeader</a>	3	0.014 s	0.014 s	
<a href="#">fileparts</a>	3	0.014 s	0.012 s	
<a href="#">imresize&gt;fixupSizeAndScale</a>	3	0.012 s	0.006 s	
<a href="#">onCleanup&gt;onCleanup.delete</a>	7	0.009 s	0.005 s	
<a href="#">rgb2gray</a>	3	0.009 s	0.007 s	
<a href="#">...graphics.internal.axesDestroyed(o.e)</a>	3	0.009 s	0.004 s	
<a href="#">unique&gt;uniqueR2012a</a>	2	0.009 s	0.009 s	
<a href="#">imresize&gt;parsePreMethodArgs</a>	3	0.009 s	0.006 s	
<a href="#">ima...ivate\readbmpdata&gt;readFromFile</a>	3	0.008 s	0.007 s	
<a href="#">ifftshift</a>	3	0.008 s	0.008 s	
<a href="#">contributions&gt;@(x)scale*kernel(scale*x)</a>	1	0.006 s	0.003 s	
<a href="#">rot90</a>	3	0.006 s	0.006 s	

<a href="#">cubic</a>	6	0.006 s	0.006 s	
<a href="#">fftshift</a>	3	0.006 s	0.006 s	
<a href="#">cell.ismember</a>	3	0.005 s	0.005 s	
<a href="#">imread&gt;get_full_filename</a>	3	0.005 s	0.005 s	
<a href="#">axesDestroyed</a>	3	0.005 s	0.005 s	
<a href="#">close&gt;getEmptyHandleList</a>	3	0.005 s	0.005 s	
<a href="#">...ger&gt;SubplotListenersManager.delete</a>	1	0.004 s	0.004 s	
<a href="#">imread&gt;parse_inputs</a>	3	0.004 s	0.004 s	
<a href="#">findFirstParamString</a>	3	0.003 s	0.003 s	
<a href="#">close&gt;checkfigs</a>	2	0.003 s	0.003 s	
<a href="#">gcbf</a>	2	0.003 s	0.003 s	
<a href="#">imresize&gt;findMethodArg</a>	3	0.003 s	0.003 s	
<a href="#">allchild&gt;getchildren</a>	1	0.003 s	0.003 s	
<a href="#">isPureNearestNeighborComputation</a>	12	0.003 s	0.003 s	
<a href="#">imresize&gt;fixupSize</a>	3	0.003 s	0.003 s	
<a href="#">imagesci\private\getFileFromURL</a>	3	0.003 s	0.003 s	
<a href="#">deriveScaleFromSize</a>	3	0.003 s	0.003 s	

<a href="#">imresize&gt;scaleOrSize</a>	3	0.003 s	0.003 s	
<a href="#">...set(rootobj,'ShowHiddenHandles'.Temp)</a>	1	0.002 s	0.002 s	
<a href="#">onCleanup&gt;onCleanup.onCleanup</a>	7	0.002 s	0.002 s	
<a href="#">uitools\private\allchildRootHelper</a>	1	0.002 s	0.002 s	
<a href="#">...te\imbmpinfo&gt;readVersion3xColormap</a>	3	0.002 s	0.002 s	
<a href="#">rgb2gray&gt;parse_inputs</a>	3	0.002 s	0.002 s	
<a href="#">resizeAlongDimUsingNearestNeighbor</a>	1	0.002 s	0.002 s	
<a href="#">imresize&gt;checkForMissingOutputArgument</a>	3	0.002 s	0.001 s	
<a href="#">imagesci\private\imbmpinfo&gt;postProcess</a>	3	0.002 s	0.002 s	
<a href="#">imresize&gt;postprocessImage</a>	3	0.002 s	0.002 s	
<a href="#">datestr&gt;getdateform</a>	3	0.002 s	0.002 s	
<a href="#">ispc</a>	3	0.002 s	0.002 s	
<a href="#">imresize&gt;warnIfPostMethodArgs</a>	3	0.002 s	0.002 s	
<a href="#">imresize&gt;preprocessImage</a>	3	0.001 s	0.001 s	
<a href="#">dimensionOrder</a>	3	0.001 s	0.001 s	
<a href="#">ima...ivate\imbmpinfo&gt;@()fclose(fid)</a>	3	0.001 s	0.001 s	
<a href="#">ima...ate\imbmpinfo&gt;decodeCompression</a>	3	0.001 s	0.001 s	

<a href="#">imresize&gt;parseParamValuePairs</a>	3	0.001 s	0.001 s	
<a href="#">ima...vate\readbmpdata&gt;@().fclose(fid)</a>	3	0.001 s	0.001 s	
<a href="#">imresize&gt;isInputIndexed</a>	9	0.001 s	0.001 s	
<a href="#">...ddenHandles'.oldUDDShowHiddenHandles)</a>	1	0.001 s	0.001 s	

**Self time** is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.

## Profile Summary

Generated 02-Mar-2018 19:03:56 using performance time.

<a href="#">Function Name</a>	<a href="#">Calls</a>	<a href="#">Total Time</a>	<a href="#">Self Time*</a>	Total Time Plot (dark band = self time)
<a href="#">part1iii</a>	1	0.907 s	0.113 s	
<a href="#">imread</a>	3	0.474 s	0.026 s	
<a href="#">imread&gt;call_format_specific_reader</a>	3	0.401 s	0.009 s	
<a href="#">imagesci\private\readbmp</a>	3	0.392 s	0.004 s	
<a href="#">imagesci\private\imbmpinfo</a>	3	0.347 s	0.013 s	
<a href="#">...\imbmpinfo&gt;initializeBMPInfoStruct</a>	3	0.270 s	0.003 s	
<a href="#">ima...\private\initializeMetadataStruct</a>	3	0.267 s	0.008 s	
<a href="#">datestr</a>	3	0.259 s	0.008 s	
<a href="#">timefun\private\dateformverify</a>	3	0.249 s	0.004 s	
<a href="#">timefun\private\formatdate</a>	3	0.245 s	0.220 s	
<a href="#">imresize</a>	3	0.146 s	0.022 s	
<a href="#">ifft2</a>	3	0.061 s	0.061 s	
<a href="#">fft2</a>	3	0.057 s	0.057 s	
<a href="#">imresize&gt;resizeAlongDim</a>	6	0.044 s	0.008 s	
<a href="#">imagesci\private\readbmpdata</a>	3	0.042 s	0.003 s	
<a href="#">imresize&gt;parseInputs</a>	3	0.040 s	0.011 s	
<a href="#">imagesci\private\imbmpinfo&gt;readBMPInfo</a>	3	0.039 s	0.012 s	
<a href="#">ima...\ivate\readbmpdata&gt;bmpReadData24</a>	3	0.038 s	0.018 s	
<a href="#">close</a>	1	0.035 s	0.010 s	
<a href="#">images\private\imresizemex (MEX-file)</a>	5	0.034 s	0.034 s	
<a href="#">contributions</a>	6	0.030 s	0.018 s	
<a href="#">imagesci\private\imbmpinfo&gt;readWin3xInfo</a>	3	0.026 s	0.006 s	
<a href="#">cnv2icudf</a>	3	0.025 s	0.025 s	



<a href="#">imread&gt;get_full_filename</a>	3	0.024 s	0.024 s	
<a href="#">close&gt;safegetchildren</a>	1	0.021 s	0.004 s	
<a href="#">imagesci\private\imbmpinfo&gt;getSignature</a>	3	0.020 s	0.020 s	
<a href="#">allchild</a>	1	0.017 s	0.008 s	
<a href="#">...te\imbmpinfo&gt;readWin3xBitmapHeader</a>	3	0.013 s	0.012 s	
<a href="#">ima...ivate\readbmpdata&gt;readFromFile</a>	3	0.013 s	0.012 s	
<a href="#">imresize&gt;fixupSizeAndScale</a>	3	0.012 s	0.006 s	
<a href="#">rgb2gray</a>	3	0.011 s	0.008 s	
<a href="#">fileparts</a>	3	0.010 s	0.009 s	
<a href="#">imresize&gt;parsePreMethodArgs</a>	3	0.009 s	0.006 s	
<a href="#">contributions&gt;@(x)scale*kernel(scale*x)</a>	2	0.009 s	0.004 s	
<a href="#">onCleanup&gt;onCleanup.delete</a>	7	0.009 s	0.004 s	
<a href="#">cubic</a>	6	0.008 s	0.008 s	
<a href="#">fftshift</a>	3	0.007 s	0.007 s	
<a href="#">rot90</a>	3	0.007 s	0.007 s	
<a href="#">cell.ismember</a>	3	0.006 s	0.006 s	
<a href="#">ifftshift</a>	3	0.005 s	0.005 s	
<a href="#">ima...ate\imbmpinfo&gt;readBMPFileHeader</a>	3	0.004 s	0.004 s	
<a href="#">imread&gt;parse_inputs</a>	3	0.004 s	0.004 s	
<a href="#">imresize&gt;fixupSize</a>	3	0.003 s	0.003 s	
<a href="#">imresize&gt;scaleOrSize</a>	3	0.003 s	0.003 s	
<a href="#">imagesci\private\imbmpinfo&gt;postProcess</a>	3	0.003 s	0.003 s	
<a href="#">isPureNearestNeighborComputation</a>	12	0.003 s	0.003 s	
<a href="#">imresize&gt;findMethodArg</a>	3	0.003 s	0.003 s	
<a href="#">deriveScaleFromSize</a>	3	0.003 s	0.003 s	
<a href="#">rgb2gray&gt;parse_inputs</a>	3	0.003 s	0.003 s	
<a href="#">findFirstParamString</a>	3	0.002 s	0.002 s	

<a href="#">...te\imbmpinfo&gt;readVersion3xColormap</a>	3	0.002 s	0.002 s	
<a href="#">...set(rootobj,'ShowHiddenHandles'.Temp)</a>	1	0.002 s	0.002 s	
<a href="#">imresize&gt;checkForMissingOutputArgument</a>	3	0.002 s	0.001 s	
<a href="#">imresize&gt;warnIfPostMethodArgs</a>	3	0.002 s	0.002 s	
<a href="#">imagesci\private\getFileFromURL</a>	3	0.002 s	0.002 s	
<a href="#">onCleanup&gt;onCleanup.onCleanup</a>	7	0.002 s	0.002 s	
<a href="#">imresize&gt;preprocessImage</a>	3	0.002 s	0.001 s	
<a href="#">imresize&gt;postprocessImage</a>	3	0.002 s	0.001 s	
<a href="#">resizeAlongDimUsingNearestNeighbor</a>	1	0.002 s	0.002 s	
<a href="#">close&gt;request_close</a>	1	0.001 s	0.001 s	
<a href="#">allchild&gt;getchildren</a>	1	0.001 s	0.001 s	
<a href="#">close&gt;getEmptyHandleList</a>	1	0.001 s	0.001 s	
<a href="#">ima...ate\imbmpinfo&gt;decodeCompression</a>	3	0.001 s	0.001 s	
<a href="#">uitools\private\allchildRootHelper</a>	1	0.001 s	0.001 s	
<a href="#">ima...ivate\imbmpinfo&gt;@()fclose(fid)</a>	3	0.001 s	0.001 s	
<a href="#">dimensionOrder</a>	3	0.001 s	0.001 s	
<a href="#">datestr&gt;getdateform</a>	3	0.001 s	0.001 s	
<a href="#">close&gt;checkfigs</a>	1	0.001 s	0.001 s	
<a href="#">ima...vate\readbmpdata&gt;@()fclose(fid)</a>	3	0.001 s	0.001 s	
<a href="#">imresize&gt;parseParamValuePairs</a>	3	0.001 s	0.001 s	
<a href="#">ispc</a>	3	0.001 s	0.001 s	
<a href="#">imresize&gt;isInputIndexed</a>	9	0.001 s	0.001 s	

**Self time** is the time spent in a function excluding the time spent in its child functions. Self time also includes overhead resulting from the process of profiling.