

BE5300-009

IMAGE PROCESSING WITH MATLAB

Fall 2015

Final Project



MRI Image Processing

Dhineshvikram Krishnamurthy (1001237384)

Due Date: 07-Dec-2015

Introduction:

Main aim of this project is to process the given MRI data set by sorting the MRI slices according to the slice location, segmenting the normal brain tissue & diseased tissue for better visualization, plotting 2D contour view & 3-D view of brain slices using MATLAB & GUI.

1: Flow Chart

Attached PDF for Flowchart.

2: To read the MRI Dicom slices.....	3
3 : To sort MRI slices using slice_location.....	4
4: Selecting one slice to show ROI	5
6: using colormap jet or colormap hot to show the color contrast of damaged.....	9
7: To show Contour View	9
8 & 9: Generating 3-D of brain slices and publish.....	11
Function -mri_transform.....	13
Function-dilation.....	14
Function- mri_segment.....	14
Function- mri_sigmoid	15

```
clc
clear all
close all

% Reading one dicom file
```

```
fname=['00000' num2str(1) '.dcm'];
info=dicominfo(fname);
```

2: To read the MRI Dicom slices

```
% Storing the dicom slices in variable 'fname' after concatenating '0000%'
% with ',dcm'

for i=1:52
    if i<=9
        fname=['00000' num2str(i) '.dcm'];
    else
        fname=['0000' num2str(i) '.dcm'];
    end

    % Transformation using ITF by calling user-defined func mri_transform

    trans = mri_transform(uint16(dicomread(fname)));

    % Storing the transformed images in mri(:, :, i)

    mri(:, :, i) = trans;
    unsort_img(i, :) = [fname];

    % dicominfo to obtain slice location

    info = dicominfo(fname);

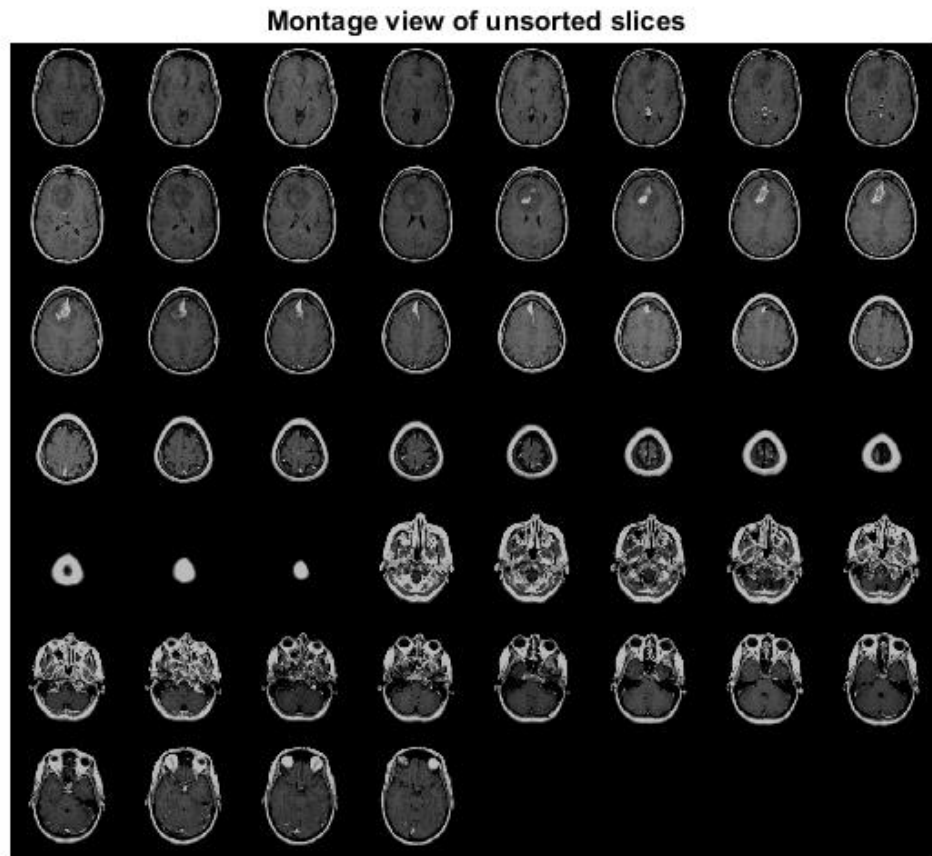
    % getting voxel size to plot 3-D-using sliceThickness and pixel spacing

    voxel_size=[info. PixelSpacing ;info. SliceThickness]';
    sliceloc(i) = info. SliceLocation;
end
whos mri
figure(1)
% creating montage view for the unsorted transformed slices

montage(reshape(double(mri), [size(mri,1) size(mri,2) 1 size(mri,3)]));
set(gca, 'clim', [0 350]);
title('Montage view of unsorted slices')
colormap(gray(256))
```

Name	Size	Bytes	Class	Attributes
mri	256x256x52	13631488	single	

Warning: Image is too big to fit on screen; displaying at 25%



3 : To sort MRI slices using slice_location

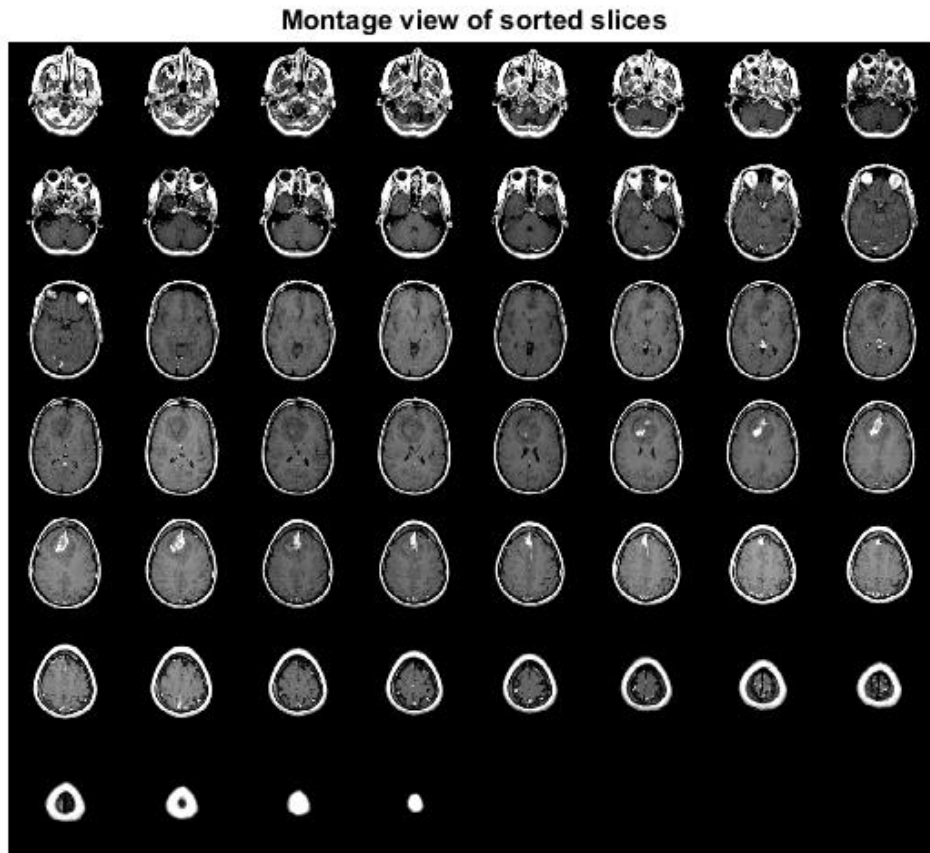
```
% sort the slices based on their slice locations
% Storing the sorted locations in a variable 'sorted_slice_loc'
% Iterating the loop till the sorted location is equal to unsorted loc
% Transforming the sorted slices using ITF and storing it in variable sorted_mri

sorted_slice_loc = sort(sliceloc);
for i = 1:length(sorted_slice_loc)
    for j = 1:length(sliceloc)
        if sorted_slice_loc(i) == sliceloc(j)
            sorted_mri(:,:,i) = mri_transform(uint16(dicomread(unsort_img(j,:))));
        end
    end
end

% creating montage view for sorted transformed slices
figure(2)
montage(reshape(double(sorted_mri), [size(sorted_mri,1) size(sorted_mri,2) 1
size(sorted_mri,3)]));
colormap(gray(256))
set(gca, 'c1im', [0 256]);
```

```
title('Montage view of sorted slices')
colormap(gray(256))
```

Warning: Image is too big to fit on screen; displaying at 25%



4: Selecting one slice to show ROI

```
% Selecting 17th slice from unsorted slice list to show ROI
```

```
figure(3)
selected_slice = mri(:, :, 17);
subplot(2, 2, 1)
image(selected_slice)
title('mri(:, :, 17)')
```

```
% Que 5: 1.Segmenting the damaged tissue by calling user-defined func mri_segment by
% -selecting appropriate seed
% 2.Selecting seed 87,108 to show ROI from selected_slice
% 3.performing dilation on the damaged segmented tissue to make the area
% -uniform by calling user defined func mri_dilation
```

```

tumor_reg = mri_segment(87,108,selected_slice);
dilated_tumor_img=dilation(tumor_reg);
subplot(2,2,2)
image(dilated_tumor_img)
title('Segmented damaged tissue')

% 1.Segmenting the normal tissue by calling user-defined func mri_segment to
% -remove skull and CSF by selecting appropriate seed.
% 2.Selecting seed 114,111 to show normal tissue from selected_slice
% 3.Adding the segmented damaged tissue and normal tissue to get the
% -segmented area deprived of skull.

brain_reg = mri_segment(114,111,selected_slice);
subplot(2,2,3)
image(brain_reg)
title('Segmented healthy tissue')

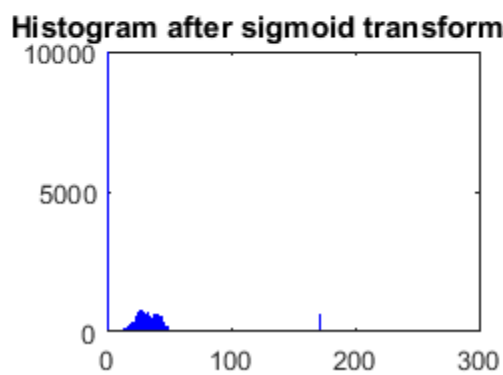
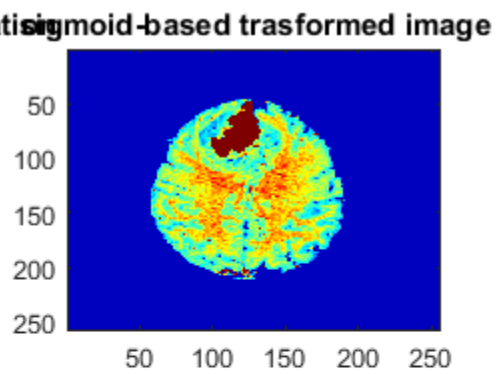
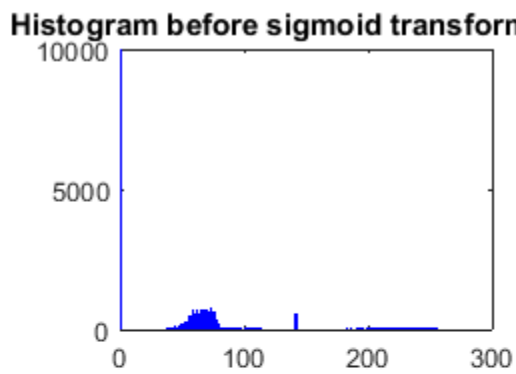
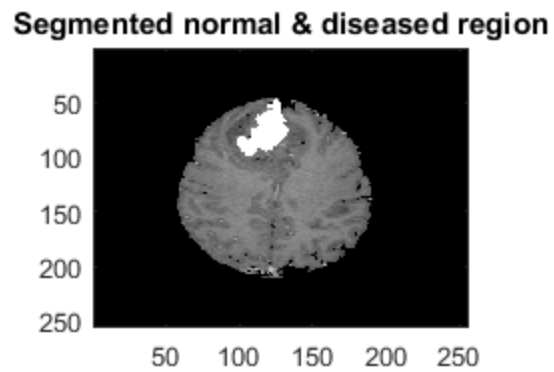
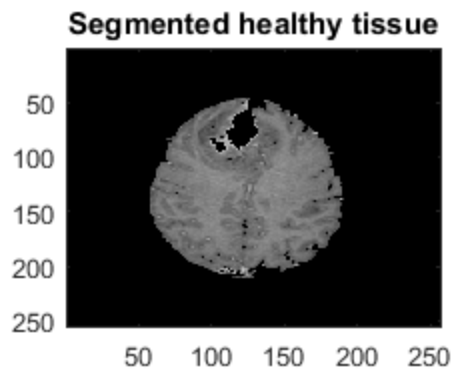
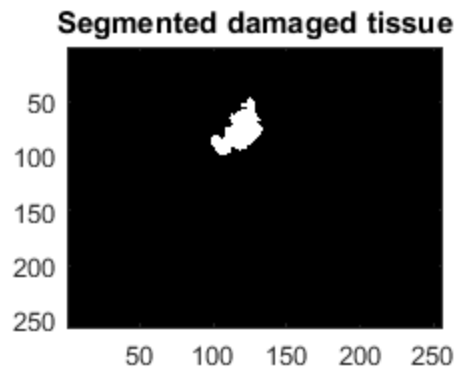
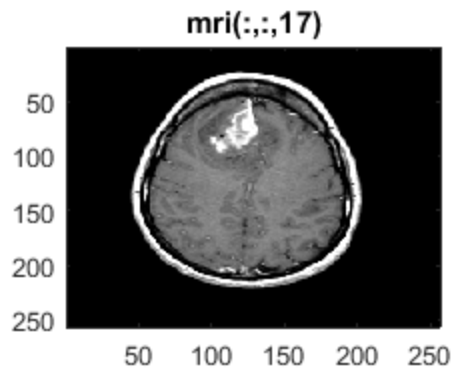
segmented_img = brain_reg+dilated_tumor_img;
subplot(2,2,4)
colormap(gray(256))
image(segmented_img)
title('Segmented normal & diseased region')

% Transforming the segmented image by calling user defined func mri_sigmoid
% -& plotting histograms

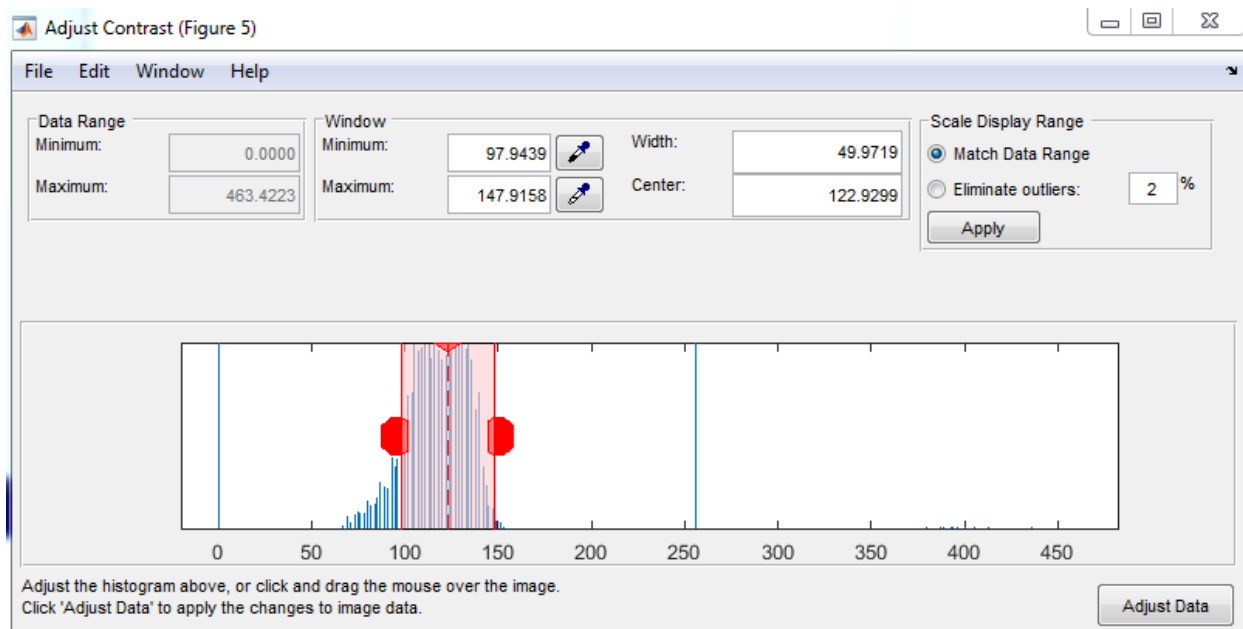
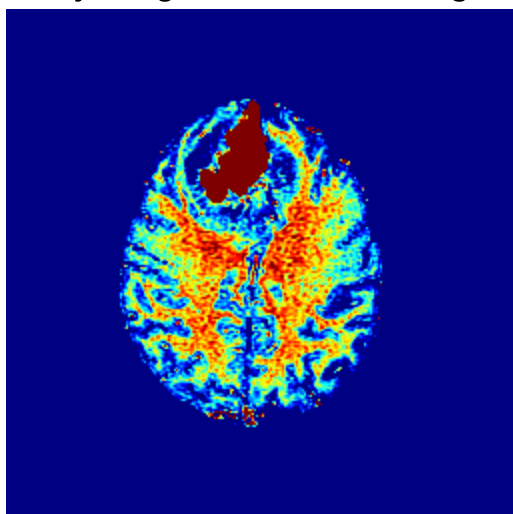
figure(4)
sigmoid_trans_image=mri_sigmoid(segmented_img);
colormap jet

figure(5)
b=imshow(segmented_img);
title('Adjusting contrast of the image')
colormap jet
imcontrast(b)

```



Adjusting contrast of the image



6: using colormap jet or colormap hot to show the color contrast of damaged and healthy tissue.

```
colormap jet
```

Que 7: To show Contour View

```
% For generating 3-D contour, taking slices from 30 to 34 in which we can  
% - see injury clearly  
% Segmenting the ROI from skull for all the selected slices  
% Generating the 3-D contour view by using contourslice
```

```
% to plot 2-D contour using contourf  
% image should be flipped from top to bottom using flipdim
```

```
figure(6)  
contourf(flipdim(segmented_img,1));  
title('2-D contour view')
```

```
% Selecting the Tumour based Slices
```

```
for range = 10:29;  
    brain_reg = mri_segment(114,111,sorted_mri(:,:,range));  
    segmented_img = brain_reg;  
    selected_set(:,:,range) = segmented_img;  
end
```

```
% Seed-87,108
```

```
for range = 30:34;  
    tumor_reg = mri_segment(87,108,sorted_mri(:,:,range));  
    dilated_tumor_img=dilation(tumor_reg);  
    brain_reg = mri_segment(114,111,sorted_mri(:,:,range));  
    segmented_img = brain_reg+dilated_tumor_img;  
    dilated_tumor_set(:,:,range) = dilated_tumor_img;  
    selected_set(:,:,range) = segmented_img;  
end
```

```
% Seed-55,126
```

```
for range = 34:36;  
    tumor_reg = mri_segment(55,126,sorted_mri(:,:,range));  
    dilated_tumor_img=dilation(tumor_reg);  
    brain_reg = mri_segment(114,111,sorted_mri(:,:,range));  
    segmented_img = brain_reg+dilated_tumor_img;  
    dilated_tumor_set(:,:,range) = dilated_tumor_img;  
    selected_set(:,:,range) = segmented_img;  
end
```

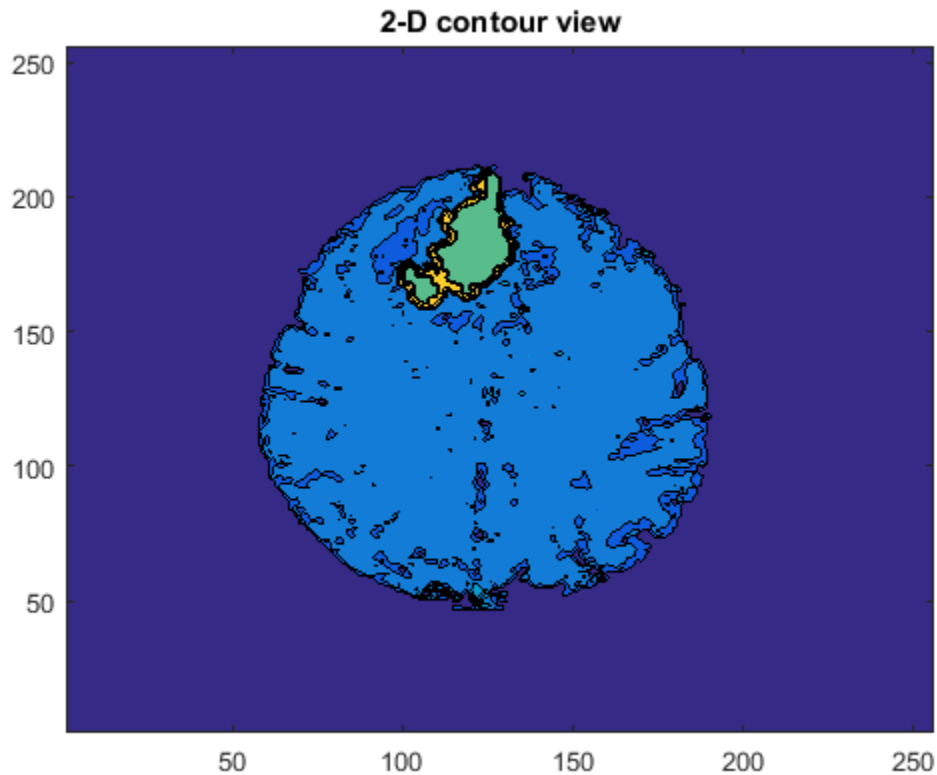
```

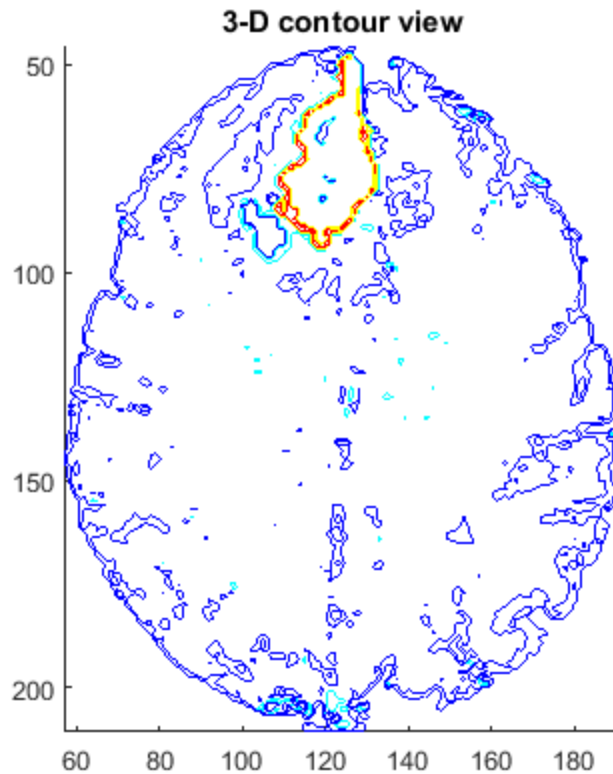
end
% Seed-67,123
for range = 38:40;
    tumor_reg = mri_segment(67,123,sorted_mri(:,:,range));
    dilated_tumor_img=dilation(tumor_reg);
    brain_reg = mri_segment(114,111,sorted_mri(:,:,range));
    segmented_img = brain_reg+dilated_tumor_img;
    dilated_tumor_set(:,:,range) = dilated_tumor_img;
    selected_set(:,:,range) = segmented_img;
end
for range = 40:47;
    brain_reg = mri_segment(114,111,sorted_mri(:,:,range));
    segmented_img = brain_reg;
    selected_set(:,:,range) = segmented_img;
end

% selecting 34th slice

image_num = 34;
cm = brighten(jet(100),1);
figure('Colormap', cm)
title('3-D contour view')
contourslice(selected_set,[],[],image_num);
axis ij tight
daspect([1,1,3])

```





8 & 9: Generating 3-D of brain slices and publish

% 1. Resizing the selected set
 % 2. Flipping the dimensions 1-for top to bottom and 2 for left to right
 % 3. Generating 3-D using isosurface and patch

```
Ds = imresize(selected_set,1,'nearest');
Ds = flipdim(Ds,1);
Ds = flipdim(Ds,2);
Ds = permute(Ds,[3 2 1]);
voxel_size2=voxel_size([1 3 2]).*[4 1 4];

white_vol = isosurface(Ds,80);
gray_vol = isosurface(Ds,200);
h=figure('visible','off','outerposition',[0 0 600 600]);
title('3-D view of selected slices')

patch(white_vol,'FaceColor','y','EdgeColor','none',...
      'FaceAlpha',.3);
patch(gray_vol,'FaceColor','r','EdgeColor','none');
view(150,0);
daspect(1./voxel_size2);
axis tight;
axis off;
camlight;
camlight(-80,-10);
```

```

lighting phong;
movegui(h,'center');
set(h,'visible','on');

% generating 3-D for segmented damaged area

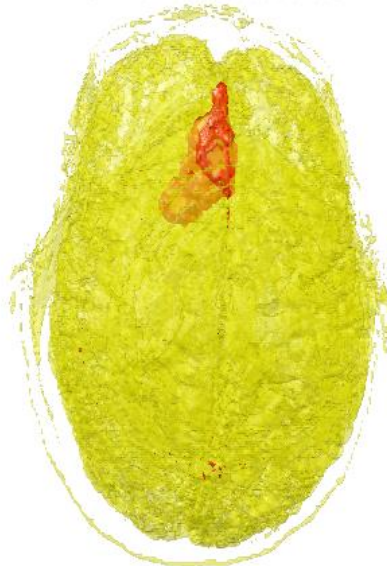
Ds = dilated_tumor_set;
Ds = flipdim(Ds,1);
Ds = flipdim(Ds,2);
Ds = permute(Ds,[3 2 1]);
voxel_size2=voxel_size([1 3 2]).*[4 1 4];

white_vol = isosurface(Ds,120);
gray_vol = isosurface(Ds,200);
h=figure('visible','off','outerposition',[0 0 600 600]);
title('3-D of damaged tissue')

patch(white_vol,'FaceColor','r','EdgeColor','none');
patch(gray_vol,'FaceColor','r','EdgeColor','none');
view(150,0);
daspect(1./voxel_size2);
axis tight;
axis off;
camlight;
camlight(-80,-10);
lighting phong;
movegui(h,'center');
set(h,'visible','on');

```

3-D view of selected slices



3-D of damaged tissue



Function -mri_transform

```
% function transform = mri_transform(selected_slice)
% AA=single(selected_slice);
% Amin=min(AA(:));
% Amax=max(AA(:));
% CC=256*(AA-Amin)/(Amax-Amin);
%
% % This part is to perform thresholding/windowing.
%
% thre_low=20;
% thre_high=120;
%
% DD = CC < thre_high & CC > thre_low; % select/mask the region within thresholds.
% DD1=CC > thre_high;
% DD1=256.*DD1;
% DD2 = (CC-thre_low).*(DD*256)/(thre_high-thre_low);% multiply the thresholded mask.
% transform=DD2+DD1;
```

Function-dilation

```
% function [dilated_img]=dilation(segmented_img)
% modified_img=zeros(size(segmented_img));
% for i=2:size(segmented_img,1)-1
%     for j=2:size(segmented_img,2)-1
%         if (segmented_img((i-1),j) ~= 0) || (segmented_img((i+1),j) ~= 0) ||
(segmented_img(i,(j-1)) ~= 0) || (segmented_img(i,(j+1)) ~= 0)
%             modified_img(i,j)=256;
%         end
%     end
% end
% dilated_img=modified_img;
```

Function- mri_segment

```
% function seedmask = mri_segment(i,j,transform)
% seedmask=zeros(size(transform));
% seedmask(i,j)=64;
% seedintensity=transform(i,j);
% seedrangemin=seedintensity-70;
% if seedrangemin < 0
%     seedrangemin = 0
% end
% seedrangemax=seedintensity+70;
% % if seedrangemax > 255
% % seedrangemax = 255;
% % end
% oldseeds = 1;
% newseeds = 0;
% while newseeds ~= oldseeds;
%     oldseeds = newseeds;
%     newseeds = 0;
%     for i = 2:size(transform,1)-1
%         for j = 2:size(transform,2)-1
%             if seedmask(i,j) > 0
%                 intens=transform((i-1),j);
%                 if (intens >= seedrangemin) & (intens <= seedrangemax)
%                     newseeds = newseeds + 1;
%                     seedmask((i-1),j) = transform((i-1),j);
%                 end
%                 intens=transform((i+1),j);
%                 if (intens >= seedrangemin) & (intens <= seedrangemax)
%                     newseeds = newseeds + 1;
%                     seedmask((i+1),j) = transform((i+1),j);
%                 end
%                 intens=transform(i,(j-1));
%                 if (intens >= seedrangemin) & (intens <= seedrangemax)
%                     newseeds = newseeds + 1;
%                     seedmask(i,(j-1)) = transform(i,(j-1));
%                 end
%                 intens=transform(i,(j+1));
```

```

%             if (intens >= seedrangemin) & (intens <= seedrangemax)
%                 newseeds = newseeds + 1;
%                 seedmask(i,(j+1)) = transform(i,(j+1));
%             end
%         end
%     end
% end
% % seedmask for injured tissue

```

Function- mri_sigmoid

```

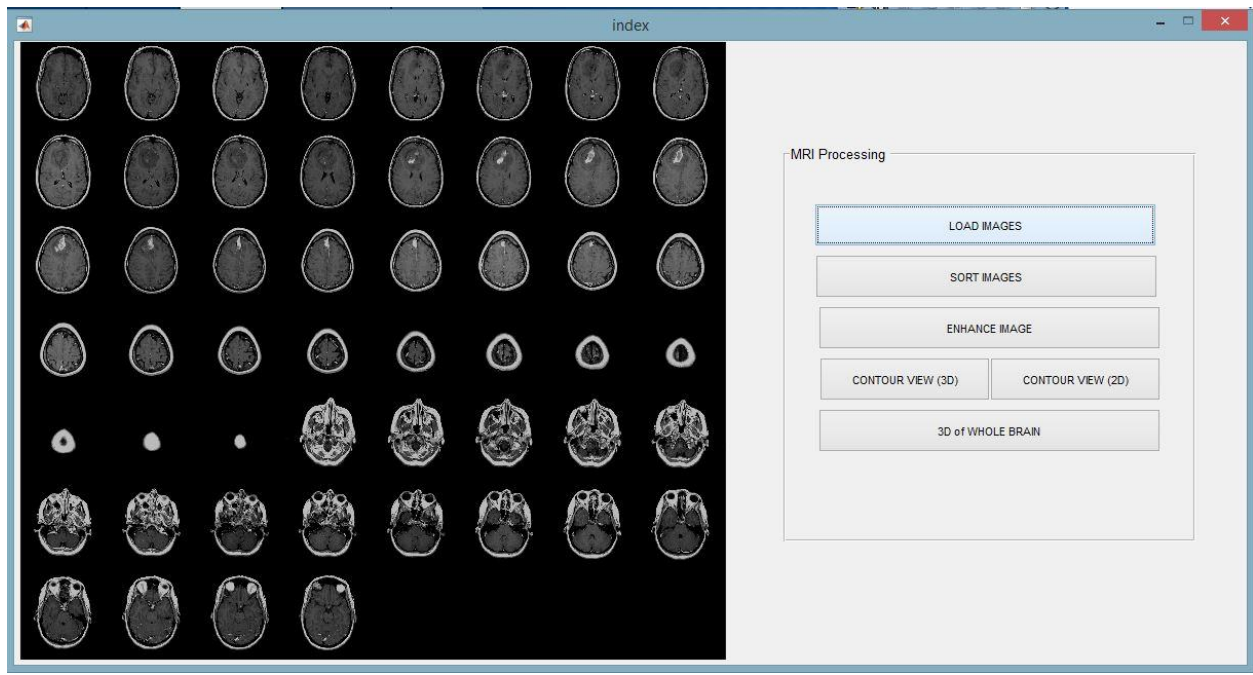
% function sigmoid_trans_image = mri_sigmoid(segmented_img)
% linear_trans_image=255*(segmented_img-min(segmented_img(:)))/(max(segmented_img(:))-
min(segmented_img(:)));
% hist_linear_img=hist(linear_trans_image(:),256)
% subplot(2,2,1)
% bar(hist_linear_img,'b')
% ylim([0 10^4])
% title('Histogram before sigmoid transformation')
% omega=120;
% sigma=30;
% sigmoid_trans_image=255./(1+exp(-1*(linear_trans_image-omega)/sigma));
% subplot(2,2,2)
% image(sigmoid_trans_image)
% title('sigmoid-based trasformed image')
% hist_sigmoid_img=hist(sigmoid_trans_image(:),256);
% subplot(2,2,3)
% bar(hist_sigmoid_img,'b');
% ylim([0 10^4])
% title('Histogram after sigmoid transformation')

```

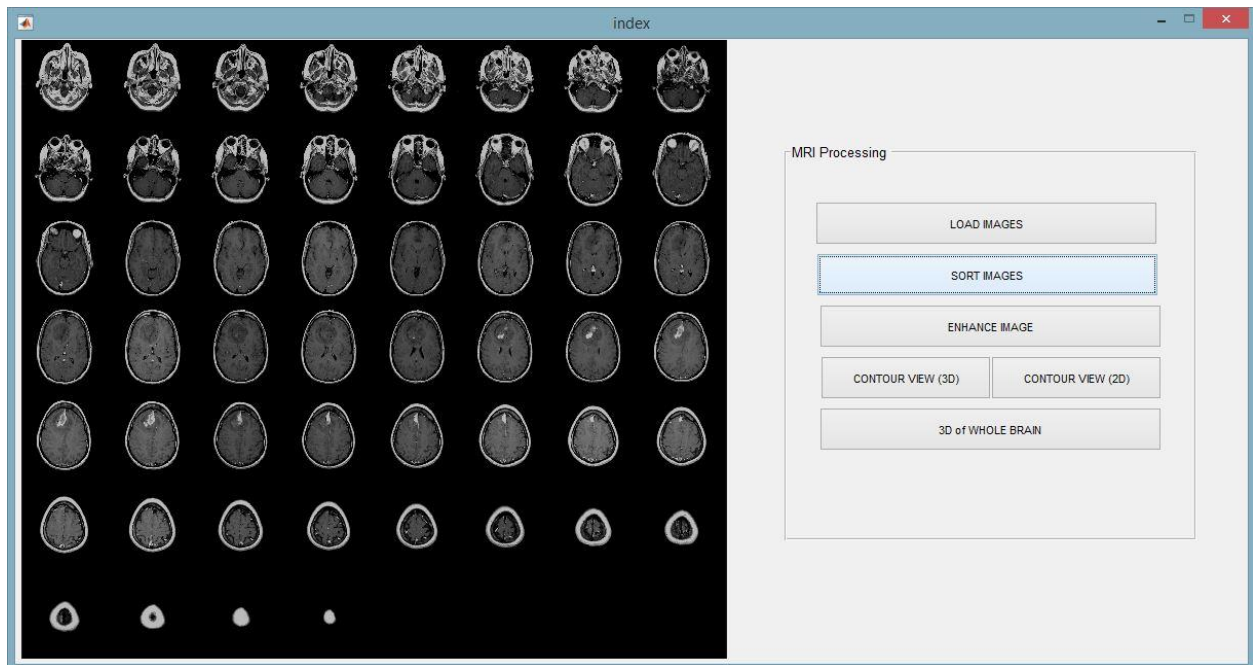
Published with MATLAB® R2015a

GUI: Simple GUI using push buttons. We will try to develop it further.

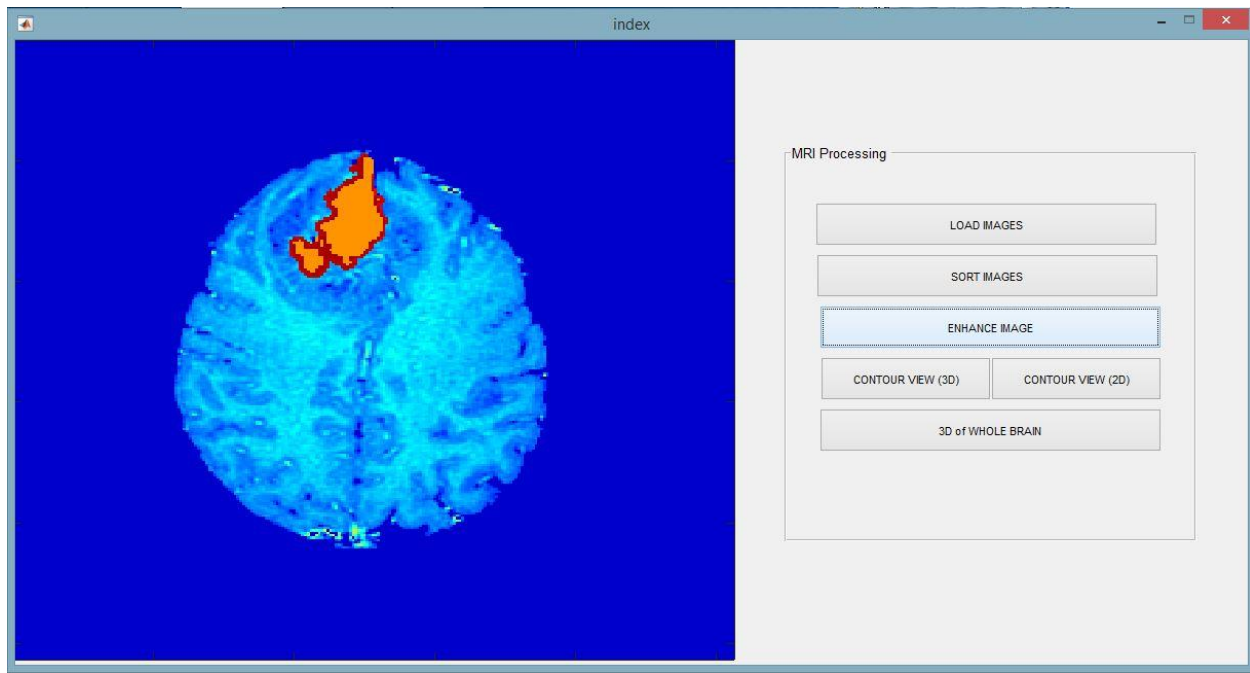
On Clicking 'Load Image' button



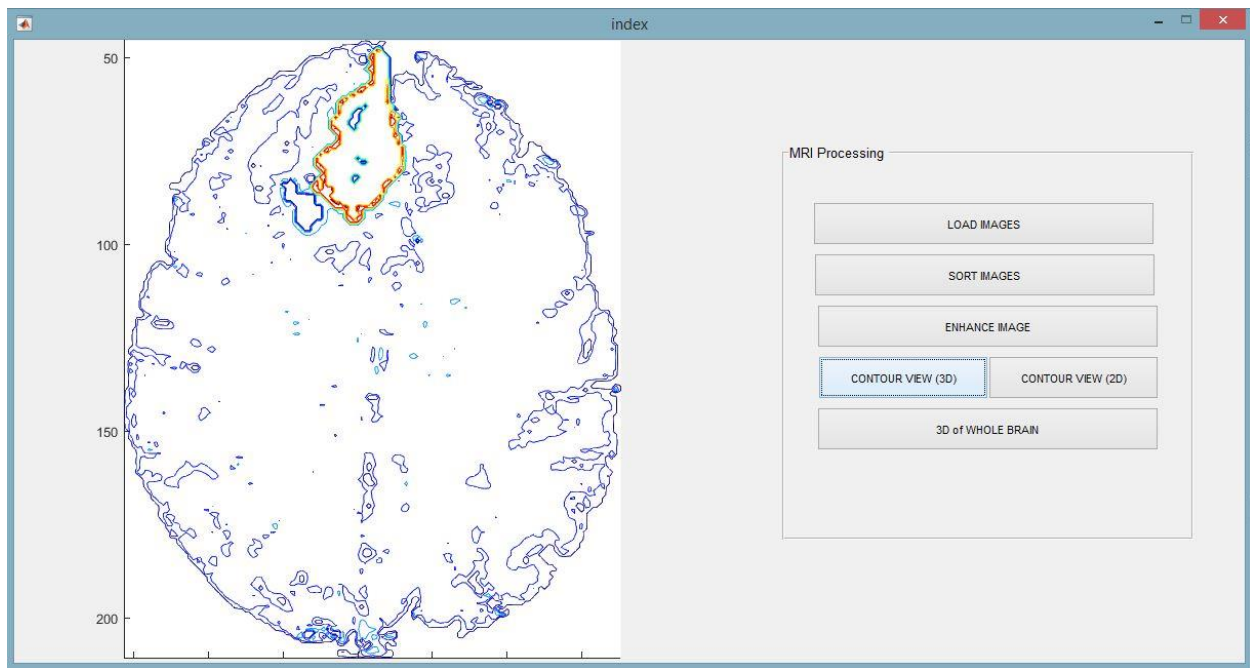
On Clicking 'Sort Images' button



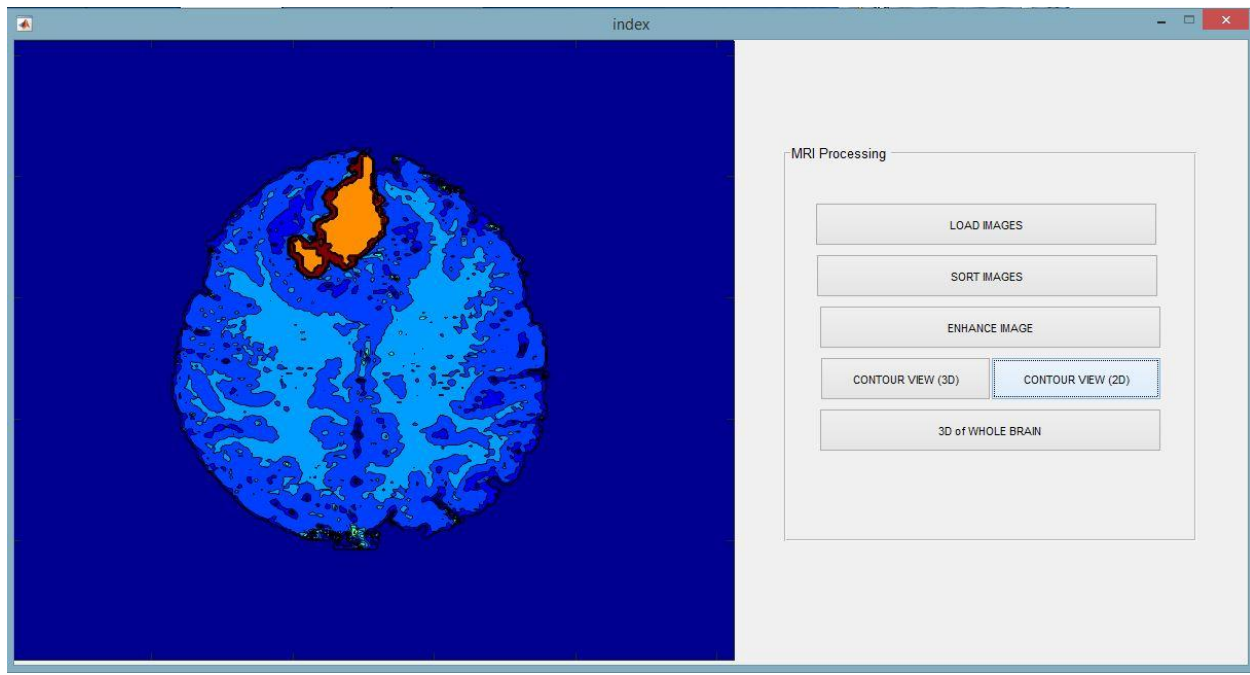
On clicking 'Enhance image'



On clicking 'Contour View-3D'



On clicking 'Contour view 2D'



On clicking 3-D of whole brain

