

---

```
clear;clc;close all;
```

## Question 1

a)

```
t1 = imread('T1.png');
t2_template = imread('T2_template.png');

SSD = SearchTranslations(t2_template,t1,'SSD');
NCC = SearchTranslations(t2_template,t1,'NCC');
MI = SearchTranslations(t2_template,t1,'MI');

% b)
figure;
subplot(1,3,1)
imagesc(SSD)
title('1b SSD')

subplot(1,3,2)
imagesc(NCC)
title('1b NCC')

subplot(1,3,3)
imagesc(MI)
title('1b MI')

% c)
% MI shows a clear maximum at [30 30]

% d)
% Calculate locations of maximums
[max_SSDx, max_SSDy] = find(SSD == max(SSD(:)));
[max_NCCx, max_NCCy] = find(NCC == max(NCC(:)));
[max_MIx, max_MIy] = find(MI == max(MI(:)));

fprintf('Max SSD: %.2d at [%.2d %.2d]\n', max(SSD(:)), max_SSDx,
    max_SSDy);
fprintf('Max NCC: %.2d at [%.2d %.2d]\n', max(NCC(:)), max_NCCx,
    max_NCCy);
fprintf('Max MI: %.2d at [%.2d %.2d]\n', max(MI(:)), max_MIx,
    max_MIy);

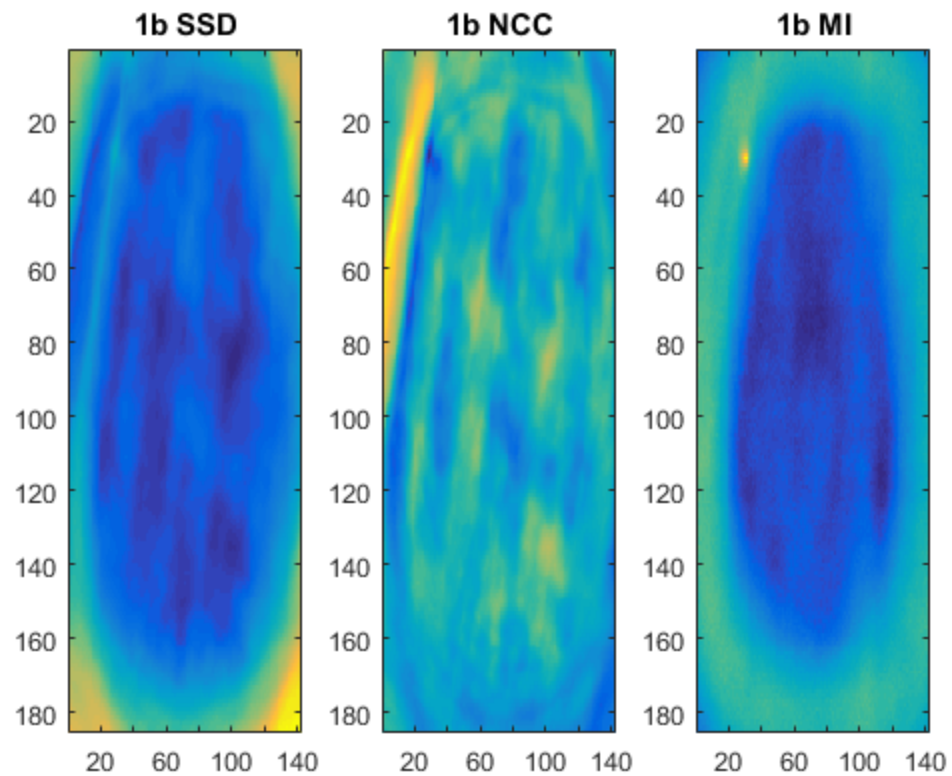
% e)
% calculate display alignments

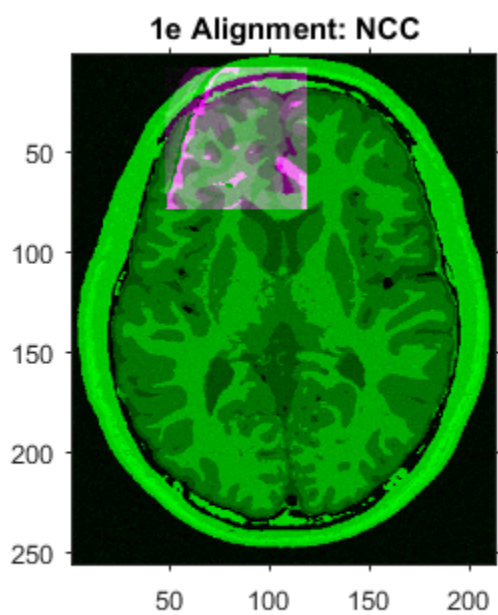
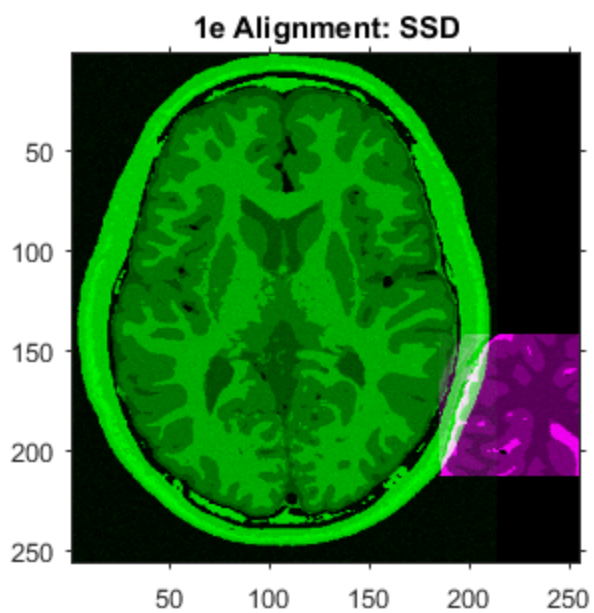
DisplayTranslatedImages(t1, t2_template, max_SSDx, max_SSDy);
title('1e Alignment: SSD');

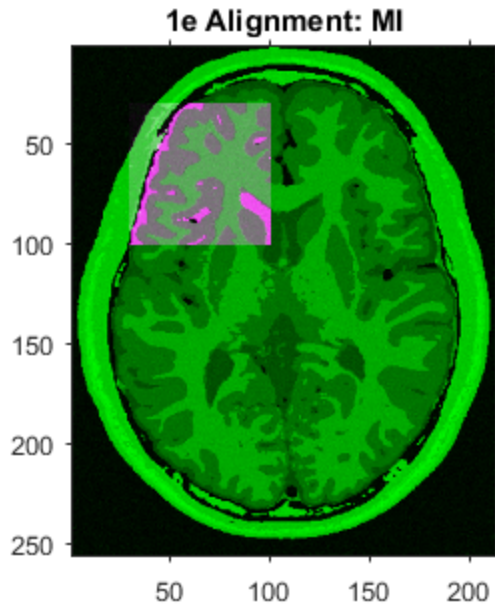
DisplayTranslatedImages(t1, t2_template, max_NCCx, max_NCCy);
title('1e Alignment: NCC');
```

---

```
DisplayTranslatedImages(t1, t2_template, max_MIx, max_MIy);  
title('1e Alignment: MI');  
  
% The MI similarity measure gives the best alignment of t1 and  
% t2_template  
  
Max SSD: 1.14e+04 at [185 142]  
Max NCC: 4.60e-01 at [48 08]  
Max MI: 2.65e+00 at [30 30]
```







## Quaetion 2

a)

```
t2 = imread('T2.png');

% I)
%create gaussian filter
gauss_filt = fspecial('gaussian',[3,3], 0.75);

% II)
% create sobol filter
sobel_filt = fspecial('sobel');

% III)
%
gaus_filt_brain = imfilter(double(t2), gauss_filt);
g_y = imfilter(gaus_filt_brain, sobel_filt);

% IV)
g_x = imfilter(gaus_filt_brain, sobel_filt');
figure;
subplot(1,3,1)
imshow(g_x,[]);title('2a grad x')
subplot(1,3,2);
imshow(g_y,[]);title('2a grad y')

% V)
M_gaus = sqrt(double(g_x).^2 + double(g_y).^2);
subplot(1,3,3);
```

---

```

imshow(M_gaus,[]);title('2a Magnitude')

% b)
% i)
alpha = atan(double(g_y)./double(g_x));

% ii)
gn = CannyNonMaximaSuppression(M_gaus, alpha);

figure; subplot(1,2,1);
imshow(M_gaus,[]); title('2b M gauss')
subplot(1,2,2);
imshow(gn,[]); title('2b canny suppressed')

% c)
% i)
% create new image of zeros and assigns indecies that correspond to
the
% values in gn that exceed the threshonld to the max value of 255.
[m, n] = size(gn);
strong_edges = zeros(m, n);
strong_edges (gn >70) = 255;

% ii)
% Duplicates the strong edges and then includes the threshold for the
weak
% edges.
weak_edges = zeros(m, n);
weak_edges (gn<70 & gn>15) = 255;

% iii)
% plots stong and weak edges
figure;
subplot(1,2,1); imshow(strong_edges,[]), title('2c Strong');
subplot(1,2,2); imshow(weak_edges,[]), title('2c Weak');

% iv)
ind = find(strong_edges);
% finds the locations of the non zero elements in the Strong edges
image.
[ind_r,ind_c]=ind2sub(size(strong_edges),ind);
% returns the subscripts corrsponding to a single index in an array.
edges = bwselect(strong_edges + weak_edges,ind_c,ind_r,8);
% This will find the weak edges that are not connected to he strong
edges
% and sets them to zero.

figure;
imshow(edges,[]);
title('2c Edges')

```

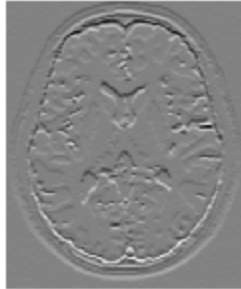
---

---

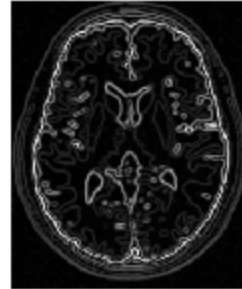
**2a grad x**



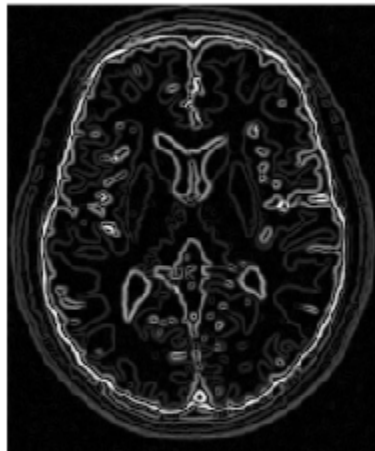
**2a grad y**



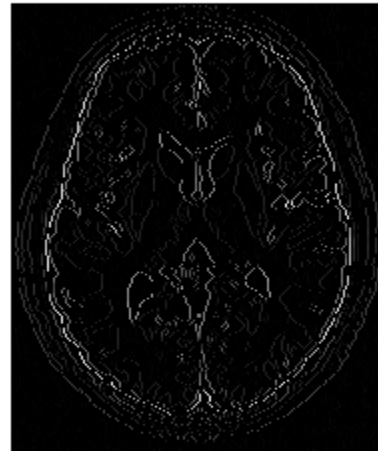
**2a Magnitude**



**2b M gauss**



**2b canny suppressed**

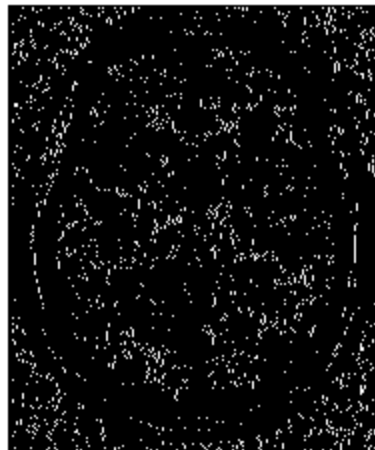


---

**2c Strong**



**2c Weak**



**2c Edges**

