Course: Introduction to Image Processing
Homework: #3
Due date: 4/06, 11:59pm

This is individual work. However, peer discussion is encouraged. For every programming assignment, clearly document your code so that the TA can follow your algorithms easily. Include the output images of your functions in the report.

Question 1. Using BrainProtonDensitySliceBorder20.png and BrainProtonDensitySliceR10X13Y17.png (see slide2, lecture7), do the following:

a. [10] Formulate the least-squares problem for similarity transformation with isotropic scaling. In other words, change the estimation model to similarity for formulation shown in slide 14 and 15 of lecture 7. (Hint: You may want to simply the problem by changing the variable to a and b, such that $a = s \cos \phi$ and $b = s \sin \phi$ to make the formulation linear.)

b. [20] Using your formulation, write a MATLAB program:

```matlab
function xformed_image=transform(fixed_pts, moving_pts, moving_image)
% fixed_pts is the point list from the fixed image.
% moving_pts is the corresponding point list from the moving image.
% moving_image is the image to be transformed to the space of the fixed image, % which is assumed the same size as the moving image.
% The result of transformation is stored in xformed_image
% Extra 5 points will be given if coded in the "matrix way"
```

Test your function with the moving image
BrainProtonDensitySliceR10X13Y17.png and the fixed image BrainProtonDensitySliceBorder20.png (and verify your result with slide2, lecture7). The corresponding point lists are:

<table>
<thead>
<tr>
<th>Point set for fixed_image ([x,y])</th>
<th>Point set of moving_image ([x,y])</th>
</tr>
</thead>
<tbody>
<tr>
<td>[95, 97]</td>
<td>[104,85]</td>
</tr>
<tr>
<td>[127,102]</td>
<td>[135,84]</td>
</tr>
<tr>
<td>[88,158]</td>
<td>[107,145]</td>
</tr>
<tr>
<td>[138,158]</td>
<td>[154,136]</td>
</tr>
<tr>
<td>[113,37]</td>
<td>[110,22]</td>
</tr>
<tr>
<td>[161,174]</td>
<td>[183,148]</td>
</tr>
<tr>
<td>[49,128]</td>
<td>[62,124]</td>
</tr>
<tr>
<td>[72,198]</td>
<td>[99,186]</td>
</tr>
</tbody>
</table>
Question 2.   [10] Using the cameraman image:

a. Compute the DFTs of the cameraman image and of the image after filtering with a 5x5 averaging filter. How does the filtering affect the output of a Fourier transform? Provide an explanation for the result. (Make sure the DFT is shifted for display)

b. What happen if the averaging filter increases in size? Discuss the result using filters of different sizes.

Question 3.   [15] To overcome some of the speed disadvantages of the median filter, some researchers proposed a variant called pseudomedian filter. For example, given a five-element sequence \{a,b,c,d,e\}, its pseudomedian is defined as

\[
\text{psmed}(a,b,c,d,e) = \frac{1}{2}\max[\min(a,b,c),\min(b,c,d),\min(c,d,e)] + \frac{1}{2}\min[\max(a,b,c),\max(b,c,d),\max(c,d,e)]
\]

In general, for an odd-length sequence \( L \) of length \( 2n+1 \), we take the maxima and minima of all subsequences of length \( n+1 \). We can apply the pseudomedian to 3x3 neighborhoods of an image, or any neighborhood with an odd number of pixels.

Write a MATLAB function to implement the pseudomedian and apply it to an image with the \texttt{nlfilter} function (testing is done Question 4).

Question 4.   [20] Using cameraman image and adding 5 % salt and pepper noise to the image. Attempt to remove the noise with a 3x3 kernel using:

a. Averaging filtering
b. Median filtering
c. Pseudomedian filtering

What do you observe and which method gives the best results? Repeat the same experiment, but with 10% and 20% noise.