Question 5:

Read in the images 'goi1.jpg' and 'goi2.jpg' from the homework folder using the MATLAB imread function and cast them as double. These are images of the Gateway of India acquired from two different viewpoints. As such, no motion model we have studied in class is really adequate for representing the motion between these images, but it turns out that an affine model is a reasonably good approximation, and you will see this. We will estimate the affine transformation between these two images in the following manner:

a) **Display both images using imshow(im1) and imshow(im2) in MATLAB. Use the ginput function of MATLAB to manually select (via an easy graphical user interface) and store n = 12 pairs of physically corresponding salient feature points from both the images. For this, you can do the following:**

   for i=1:12, figure(1); imshow(im1/255); [x1(i), y1(i)] = ginput(1);

   figure(2); imshow(im2/255); [x2(i), y2(i)] = ginput(1);

   Solution:

   The points are selected from Figure 1 as well as Figure 2 using the Matlab code and selected points coordinated are given below. Points are selected from the common region of two images and edge places are preferred. The points are quite the same place correspondence of both images but the coordinates are somehow different as image 2 is obtained by applying affine transformation on image 1.

   Figure 1 goi1 selected points are:
   Columns 1 through 12

   99.0600 426.7400 24.8200 127.2200 380.6600 454.9000 255.2200 564.9800 173.3000
   362.7400 163.0600 165.6200 168.9800 176.6600 171.5400 28.1800 43.5400 238.1000
   327.7000 192.0200 248.3400 243.2200 71.7000 48.6600

   Figure 2 goi2 selected points are:
   Columns 1 through 12

   132.3400 472.8200 37.6200 165.6200 419.0600 503.5400 303.8600 623.8600 209.1400
   403.7000 403.7000 204.0200 186.9000 181.7800 186.9000 43.5400 58.9000 256.0200
   343.0600 197.1400 266.2600 243.2200 102.4200 76.8200

b) **Write MATLAB code to determine the affine transformation which converts the first image ('goi1') into the second one ('goi2').**

Solution: After checking for all the affine transformations we get that goi2_downsampled is a scaling as well as shear of goi1.

c) **Using nearest neighbor interpolation that you should implement yourself, warp the first image with the affine transformation matrix determined in the previous step, so that it is now better aligned with the second image.**

Solution: After applying neighbor interpolation on gio1 we get Figure 3. Here we can compare these three images. The black strip in output images is the extra part of gio1 which we did not obtain after applying linear interpolation.

Figure 1: gio1	
Figure 2: gio2	

Figure 3: Obtained after applying Linear Interpolation on gio1.	

a) **Repeat the previous step with bilinear interpolation that you should implement yourself.**
   Solution: After applying neighbor interpolation on gio1 we get Figure 3. Here we can compare these three images. The black strip in output images is the extra part of gio1 which we did not obtain after applying bilinear interpolation.

Figure 1: gio1	

Figure 2: gio2	
Figure 3: Obtained after applying Bilinear Interpolation on gio1.	

b) **In the first step, suppose that the n points you chose in the first image happened to be collinear. Explain the effect on the estimation of the affine transformation matrix.**
Solution:

Suppose that the n points you chose in the first image happened to be collinear. Need to find the effect on the estimation of the affine transformation matrix. If they are collinear then it can have a significant effect on the estimation of the affine transformation matrix. Collinear points all lie on the same straight line and provide limited information about the transformation.

Here are the situations that can impact our estimation:

1. Sometimes it can give results that are overfitted. When we try to include collinear points in the estimation process, the algorithm might try to fit the transformation

matrix to these points, even though they do not represent the underlying transformation correctly.

2. Collinear points essentially reduce the number of independent data points available for estimating the affine transformation. We need at least three collinear points for unique affine transformation. These do not give additional information on the independent nature implying we have the loss of degrees of freedom. Hence, no contribution to the estimation of the transformation matrix.

3. Information about other transformation components may be missing as collinear points are primarily useful for estimating linear transformations along the direction of the collinearity.

4. Collinear points can make the estimation process more sensitive to noise in the data.

5. The algorithm may become unstable because of transformation estimation by collinear points.

6. This instability can lead to inaccurate and unpredictable results as small perturbations in the position of these places can lead to diverse inappropriate effects on the estimated transformation.