Namrata Vivek Raghavan
nraghav@purdue.edu

Interest point extraction and establish correspondence

In the following homework, given two photos taken from slightly different view points we have been asked to first extract the interest points from the two pictures and two establish correspondence between the interest points in the two images.

Task 1

1.1 Extraction of interest points using the Harris Corner Detector

The Harris corners are pixels in the images around which there is a significant variation in the grayscale values in atleast two different directions. Thus, making the Harris corner invariant to rotation. The Harris corner detects the corner points using the x and y gradient of the pixels in the image. The x and y pixels in the image can be calculated using the Sobel operator which uses two 3x3 matrices to find the x and y derivates of the image.

The x derivative is calculated by convolving the grayscale values in the image with the matrix

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

The y derivative is calculated by convolving the grayscale values in the image with the matrix

the x derivates, $D_x = \ I \ * S_x$ and the y derivative is , $D_y = \ I \ * S_y$

Once the x and y derivates are calculated, calculate the squares of the x derivates, the squares of the y derivates and the product of the x and y derivates

$$D_{x^2} = D_x^2$$

$$D_{y^2} = D_y^2$$

$$D_{xy} = D_x \times D_y$$

Once the above calculations are completed the corner points are calculated by looping through the image and creating a search window over every pixel (x,y) and creating the following matrix H(x,y)

$$H_{x,y} = \begin{bmatrix} \Sigma D_{x^2} & \Sigma D_{xy} \\ \Sigma D_{xy} & \Sigma D_{y^2} \end{bmatrix}$$

Once the H matrix is created calculate the determinant and the trace of the matrix. The pixel at (x,y)

is considered as a corner point if R is greater then 10000, where R is given by

$$R = Det\ (H) - k \times \pi(Trace(H))^2,\ 0.04 < k < 0.06$$

Once all the possible corner points are detected, they are then sorted based on descending order of R to stores the corner points from the strongest to the least strong corner point. Next, we go through the first 10000 strongest points and delete the points that are in a 30 x 30-pixel window close to one another. Lastly we select the first 200 strongest points corner points that are clustered together.

Namrata Vivek Raghavan
nraghav@purdue.edu

1.2 Feature Mapping between the two Images:

1.2.1 SSD : Sum of Squared Differences

The first method for establishing correspondence between the corner points is the sum of squared difference method. In this method we create a (M+1) x (M+1) matrix around the selected corner points. Once the matrix is created SSD is calculate as,

$$SSD = \Sigma_i \Sigma_j |f_1(i,j) - f_2(i,j)|^2$$

, where f_1 is the window around the point in the 1st image anf_2 is the window around the point in the 2nd image. For each point in the first image find the corresponding minimum SSD value in the second image. Once the minimum SSD values have been calculated for each of the selected corner points the we can threshold the SSD values to find the point correspondence. In this homework I have thresholded the SSD values to 5 × min(SSD).

1.2.2 NCC : Normalized Cross Correlation

The second method to establish correspondence between the points is using the normalized cross correlation method. In this method we once again create a (M+1) x (M+1) matrix around the selected corner points. Once the matrix is created NCC is calculate as,

$$NCC = \frac{\Sigma_i \Sigma_j (f_1(i,j) - m_1)(f_2(i,j) - m_2)}{\sqrt{(\Sigma_i \Sigma_j (f_1(i,j) - m_1)^2 \times \Sigma_i \Sigma_j (f_1(i,j) - m_1)^2)}}$$

where f_1 is the window around the point in the 1st image anf_2 is the window around the point in the 2nd image. For each point in the first image find the corresponding maximum NCC value in the second image. Once the maximum NCC values have been calculated for each of the selected corner points the we can threshold the NCC values to find the point correspondence. In this homework I have thresholded the NCC values approximately between 0.95 and 1.5, since this is that range that has given me the most accurate results.

Task 2:

2. Extraction of interest point using SIFT

Another method for extraction of interest points is the Scale Invariant Feature Transform. For this task I have used the in-built SIFT function in OpenCV to implement the SIFT method to extract the interest points from the image.

SIFT is based on scale space analysis to find interest points. In SIFT the interest points are defined as the local extrema of the Difference of Gaussian (DoG) pyramid. The DoG is defined as,

$$D(x, y, \sigma) = ff(x, y, \sigma_1) - ff(x, y, \sigma_2) ,$$

where $ff(x, y, \sigma)$ is the image convolved with a gaussian filter. The DoG pyramid extends through multiple octaves located at i* σ, where i = 1,2,3… with multiple DoGs calculated between the different octave levels. Each local extrema in the pyramid is found by a comparison in a three dimensional space (x, y, σ). The Taylor series is then used to estimate the second order derivative of the sampling points in the DoG pyramid to localize the extremum with sub pixel accuracy.

Namrata Vivek Raghavan
nraghav@purdue.edu

$$D(\vec{X}) = D(\vec{X}_o) + J^T(\vec{X}_o)(\vec{X}) + 0.5 * \vec{X}^T H(\vec{X}_o)\vec{X},$$

Where $\vec{X}$, is the incremental deviation from $\vec{X}_o$ and where J is the gradient vector estimated at $\vec{X}_o$, $J(\vec{X}_o) = \left(\frac{\delta D}{\delta x}\frac{\delta D}{\delta y}, \frac{\delta D}{\delta z}\right)^T_{\vec{X}_o}$. H is the Hessian defined as

$$H(\vec{X}_o) = \begin{bmatrix} \frac{\delta^2 D}{\delta x^2} & \frac{\delta^2 D}{\delta x \delta y} & \frac{\delta^2 D}{\delta x \delta \sigma} \\ \frac{\delta^2 D}{\delta y \delta x} & \frac{\delta^2 D}{\delta y^2} & \frac{\delta^2 D}{\delta y \delta \sigma} \\ \frac{\delta^2 D}{\delta \sigma \delta x} & \frac{\delta^2 D}{\delta \sigma \delta y} & \frac{\delta^2 D}{\delta \sigma^2} \end{bmatrix}$$

Thus, this approximation for D allows to localize the SIFT keypoints with sub-pixel precision. Weak extrema are then removed by thresholding $D(\vec{X}) > 0.03$.

2.2 Feature Mapping between the two Images:

2.2.1 Calculating the descriptor vectors

Once the key point between the two images were calculated using the OpenCV SIFT function we then established correspondence between the points by calculating the Euclidean distance between the descriptor vectors in the two images. The descriptor vectors for a SIFT key point is a 128-dimensional vector that is calculated based off the pixels in a window around the keypoint in the smooth image. The dominant local orientation is determined from the gradient vector of the Gaussian smoothed image $ff(x, y, \sigma)$ at scale σ of the extremum. To do this, we first need to determine the orientation of the keypoint from the orientation and magnitudes of the greyscale value in a K x K window around it. The orientation of the keypoint is given by

$$\theta(x, y) = arctan \frac{ff(x, y+1, \sigma) - ff(x, y, \sigma)}{ff(x+1, y, \sigma) - ff(x, y, \sigma)}$$

The magnitude is given by,

$$M(x,y) = \sqrt{(ff(x+1, y, \sigma) - ff(x, y, \sigma))^2 + (ff(x, y+1, \sigma) - ff(x, y, \sigma))^2}$$

Next θ is weighted with m and a histogram is constructed using 36 bins spanning a full 360° range. The bins with the greatest peak give the dominant local orientation. Next the descriptor is created using a 16 × 16 window of pixels that is divided into 4 x 4 cells. The magnitude of the gradients in the 16 x 16 window are weighted by a gaussian whose σ is half the width of the window. Thus, from every 4 x 4 cell we get one 8 bin histogram from the gradient magnitude weighted values of $\theta(x, y)$ at the 16 pixels in the cell. These histograms together form the 128-dimensional descriptor for each key point. In this task I have used the OpenCV inbuilt SIFT function to calculate the key points and their respective descriptor values.

2.2.2 Calculating the Euclidean distance between the descriptor vectors

Namrata Vivek Raghavan
nraghav@purdue.edu

The correspondence between two key points is established by calculating the Euclidean distance between their descriptor vectors. The Euclidean distance between the Euclidean distance is calculated using the following formula

$$D = \sqrt{\Sigma_i^{128} (f_i - g_i)^2},$$

Where f is the descriptor vector from one image and g is the descriptor vector from another image. The smaller the Euclidean distance between the descriptors more likely they are a match. Thus to determine point correspondence I place the threshold of the Euclidean distances to around 10 times the minimum value.

3. Observation and Conclusions

Between the Harris Corner Detector and the SIFT, we see that in general the SIFT method is a much more robust approach to mapping interest points between the two images. This is mainly since the Harris Corner detector depends on the parameters used for scaling and the matching algorithms. In the case of the Harris corner detector we also see that as the scale increases the mapping between the interest points is seen to improve as the number of interest points located along the edges is seen to reduce.

Between the SSD and the NCC method, the SSD is seen to be more robust than the NCC since the NCC is highly depended on the size of the window around the corner as well as the parameter used to threshold the values

4. Assignment Results

4.1 Pair 1

Input Images

Figure 1: Input image 1 of pair 1

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 2: Input image 2 of pair 1

Figure 3: Point correspondence found using NCC in pair 1 at sigma = 1.5

Namrata Vivek Raghavan
nraghav@purdue.edu

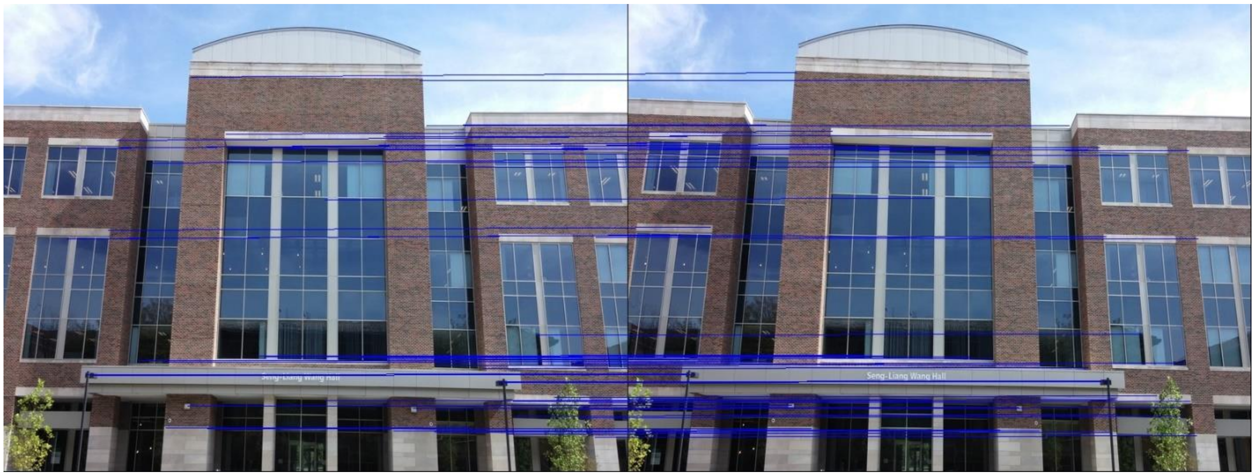

Figure 4: Point correspondence found using SSD in pair 1 at sigma = 1.5

Figure 5: Point correspondence found using NCC in pair 1 at sigma = 2

Figure 6: Point correspondence found using SSD in pair 1 at sigma = 2

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 7: Point correspondence found using NCC in pair 1 at sigma = 2.5

Figure 8: Point correspondence found using SSD in pair 1 at sigma = 2.5

Figure 9: Point correspondence found using NCC in pair 1 at sigma = 3

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 10: Point correspondence found using SSD in pair 1 at sigma = 3

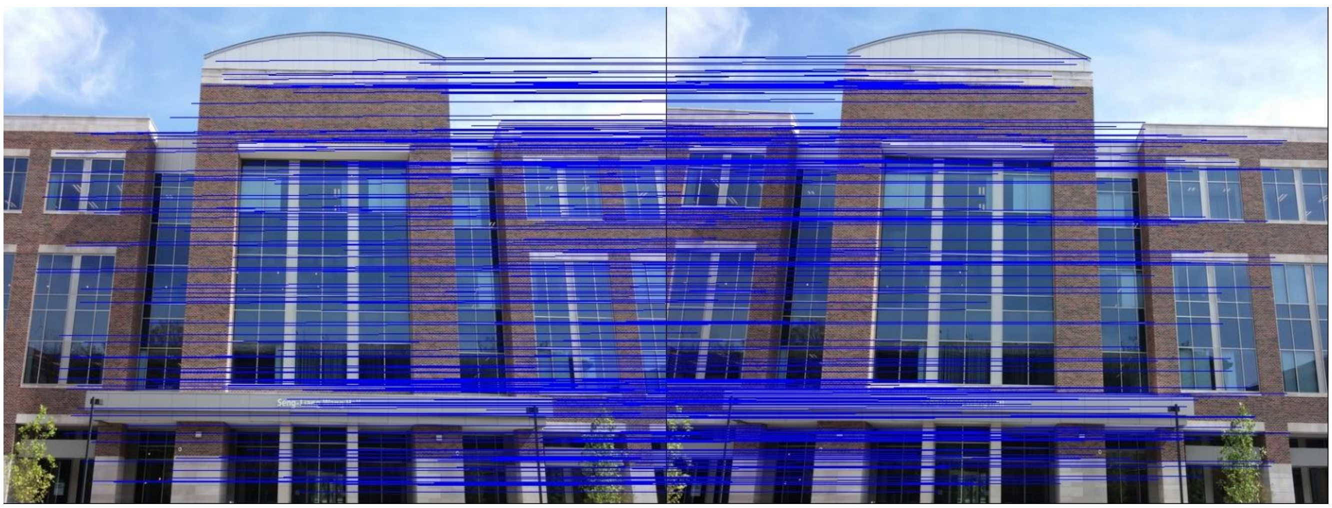

Figure 11: Point correspondence found using SIFT in pair 1

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 12: Input image 1 of pair 2

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 13: Input image 2 of pair 2

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 14: Point correspondence found using SSD in pair 2 at sigma = 1.5

Figure 15: Point correspondence found using NCC in pair 2 at sigma = 1.5

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 16: Point correspondence found using SSD in pair 2 at sigma = 2

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 17: Point correspondence found using NCC in pair 2 at sigma = 2

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 18: Point correspondence found using SSD in pair 2 at sigma = 2.5

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 19: Point correspondence found using NCC in pair 2 at sigma = 2.5

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 20: Point correspondence found using SSD in pair 2 at sigma = 3

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 20: Point correspondence found using NCC in pair 2 at sigma = 3

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 21: Point correspondence found using SIFT in pair 2

Namrata Vivek Raghavan
nraghav@purdue.edu

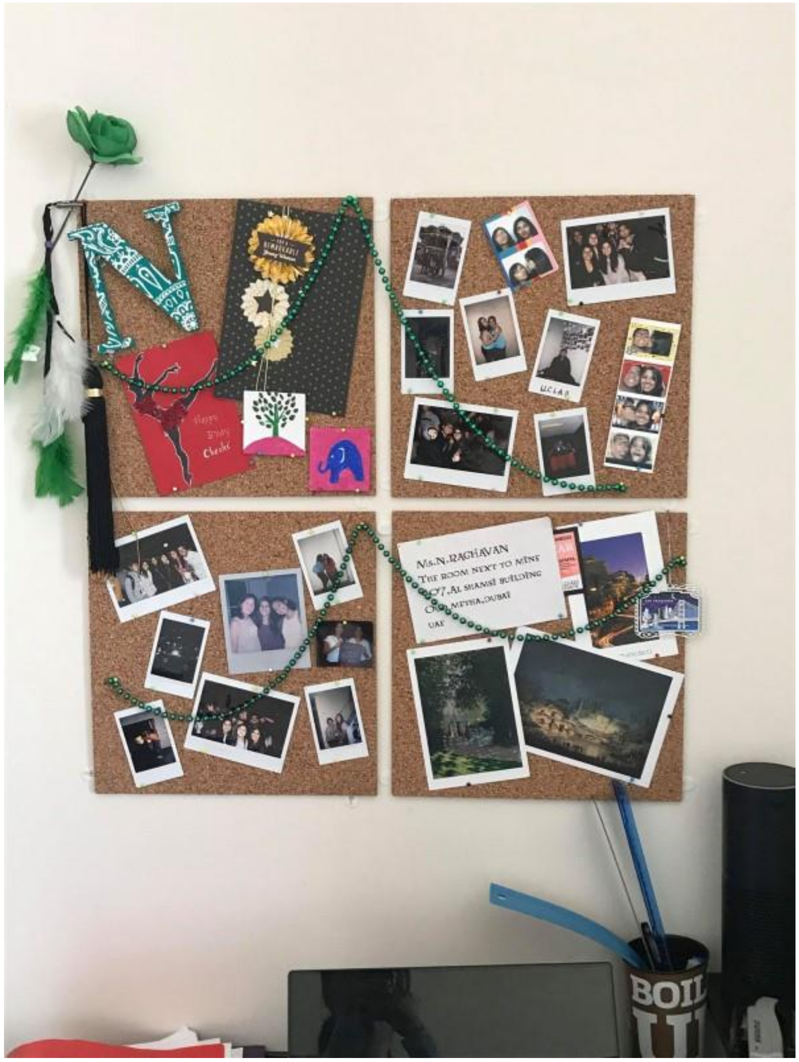

Figure 22: Input image 1 of pair 3

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 23: Input image 2 of pair 3

Namrata Vivek Raghavan
nraghav@purdue.edu

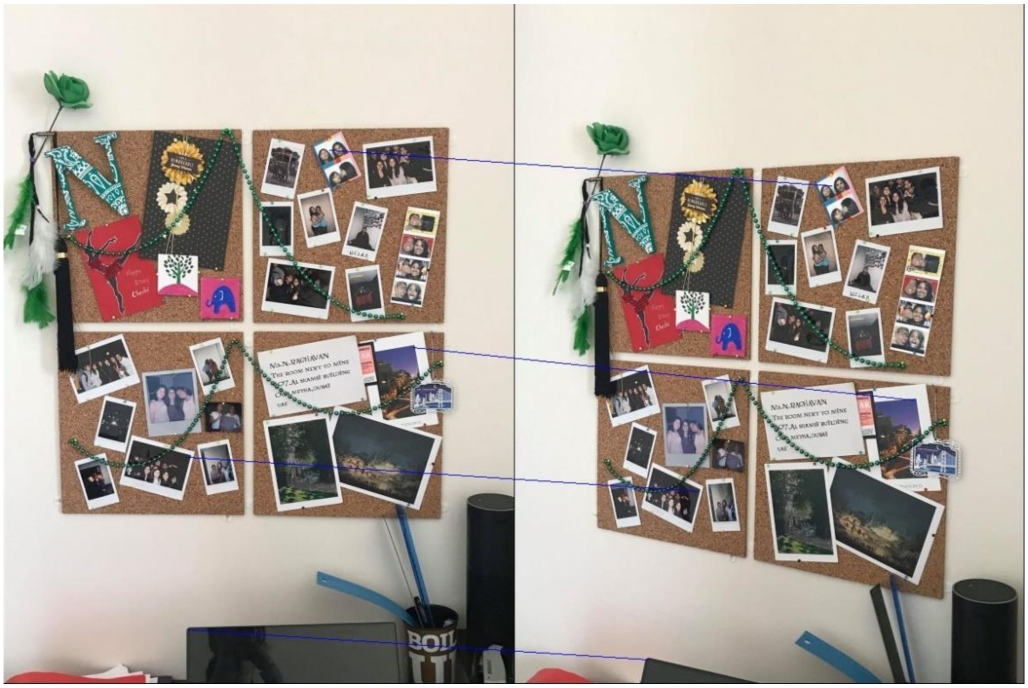

Figure 24: Point correspondence found using NCC in pair 3 at sigma = 1.5

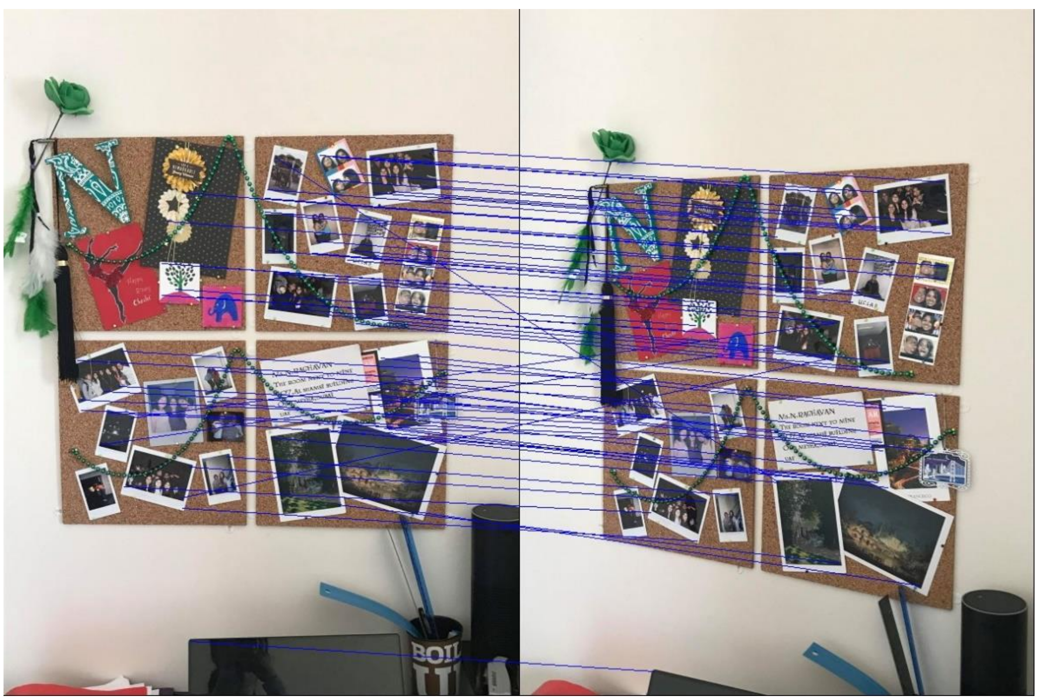

Figure 25: Point correspondence found using SSD in pair 3 at sigma = 1.5

Namrata Vivek Raghavan
nraghav@purdue.edu

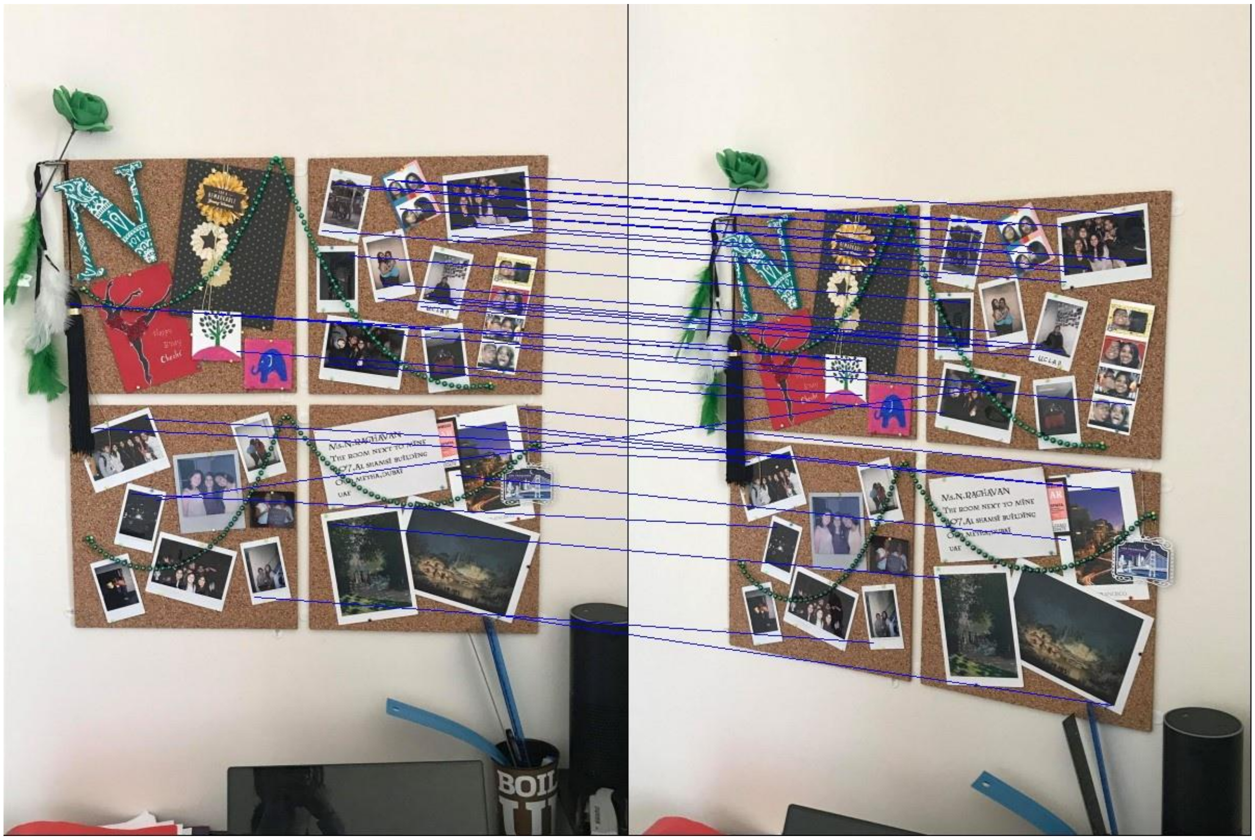

Figure 26: Point correspondence found using SSD in pair 3 at sigma = 2

Namrata Vivek Raghavan
nraghav@purdue.edu

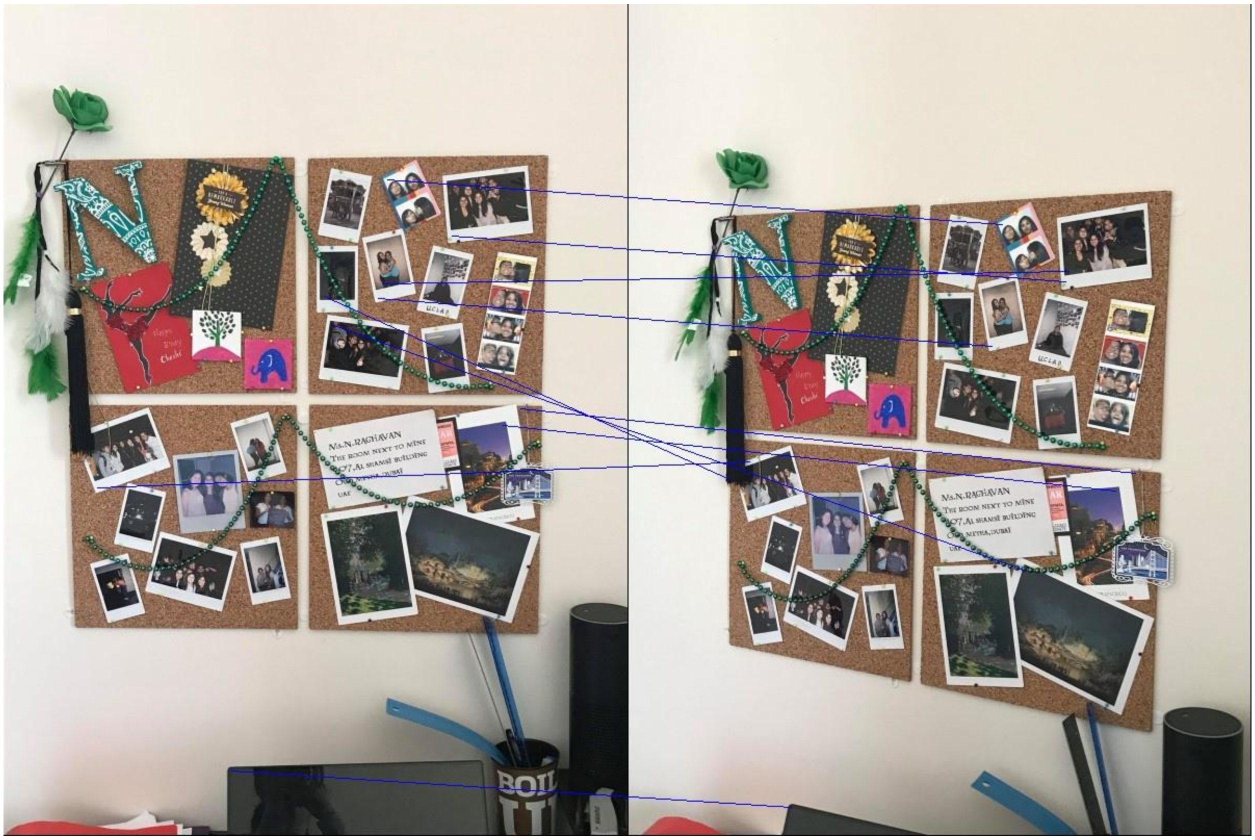

Figure 27: Point correspondence found using NCC in pair 3 at sigma = 2

Namrata Vivek Raghavan
nraghav@purdue.edu

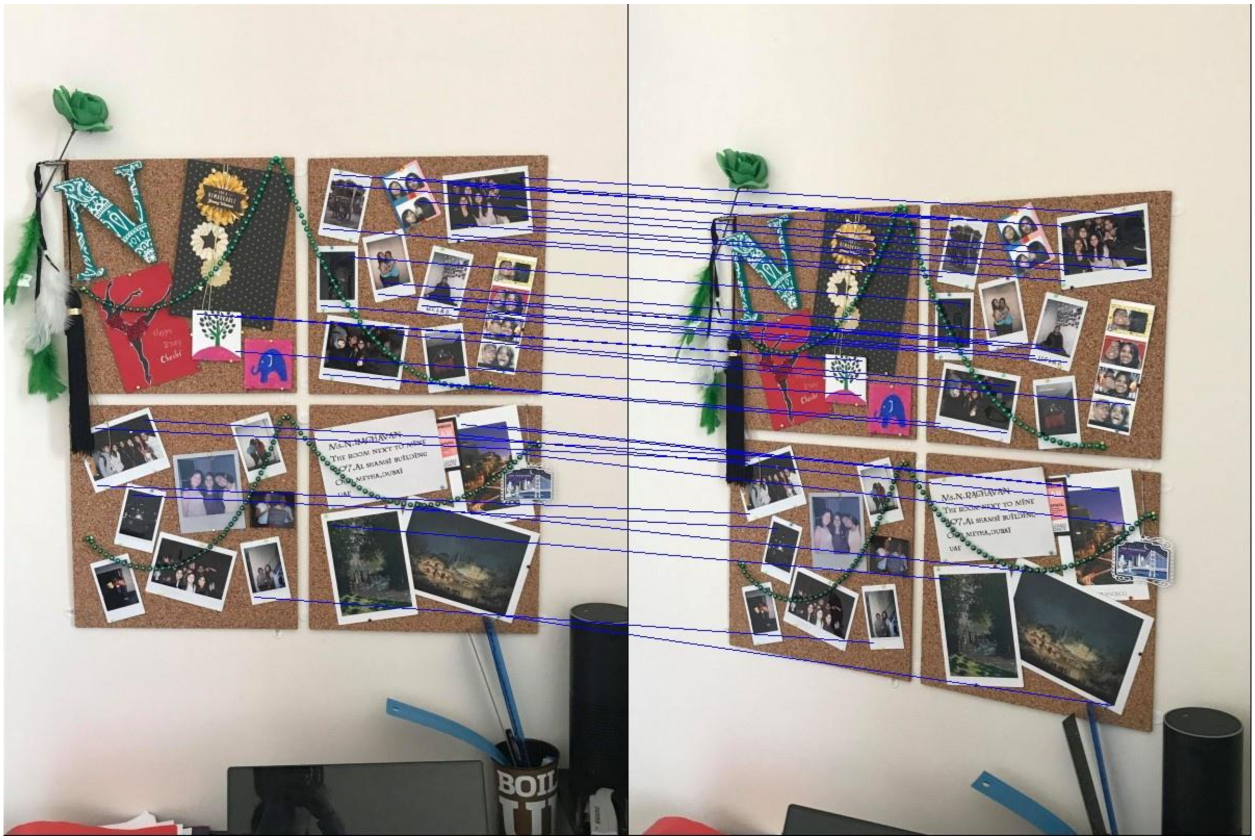

Figure 28: Point correspondence found using SSD in pair 3 at sigma = 2.5

Namrata Vivek Raghavan
nraghav@purdue.edu

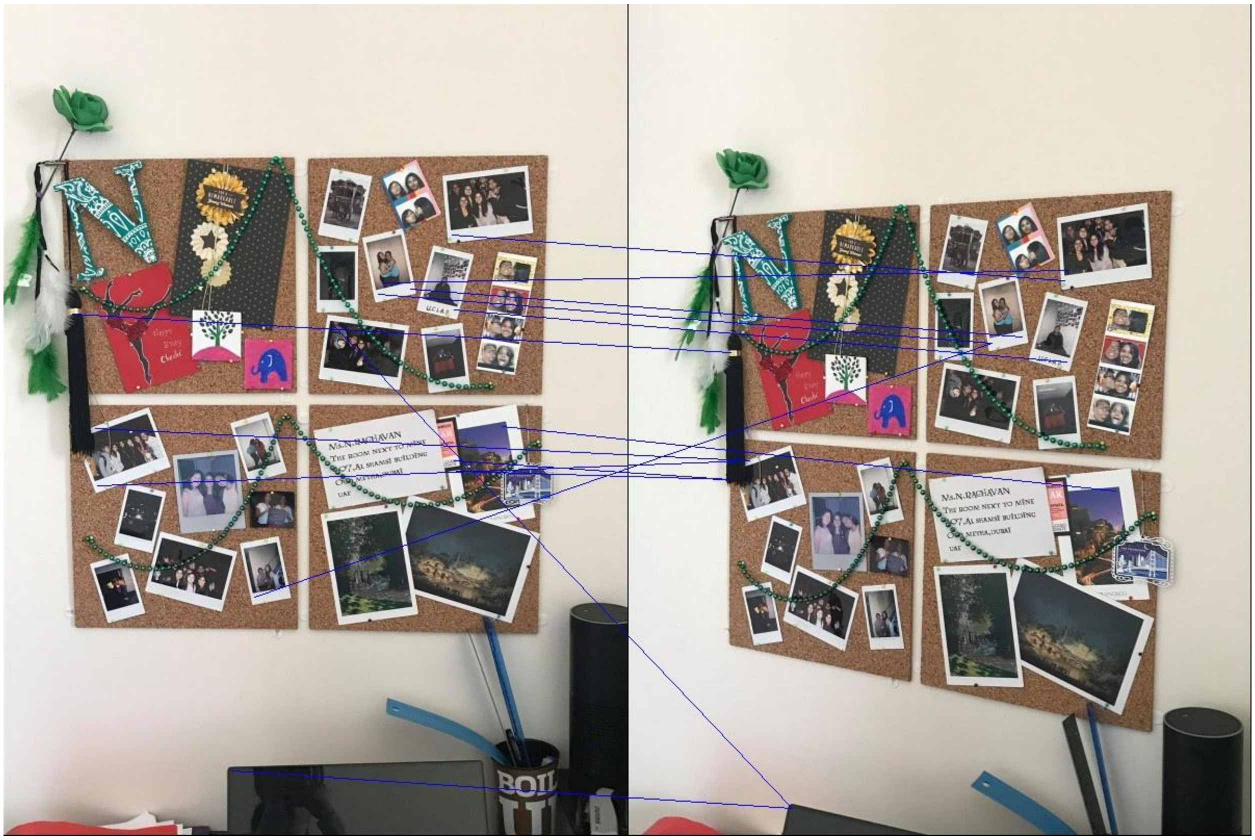

Figure 29: Point correspondence found using NCC in pair 3 at sigma = 2.5

Namrata Vivek Raghavan
nraghav@purdue.edu

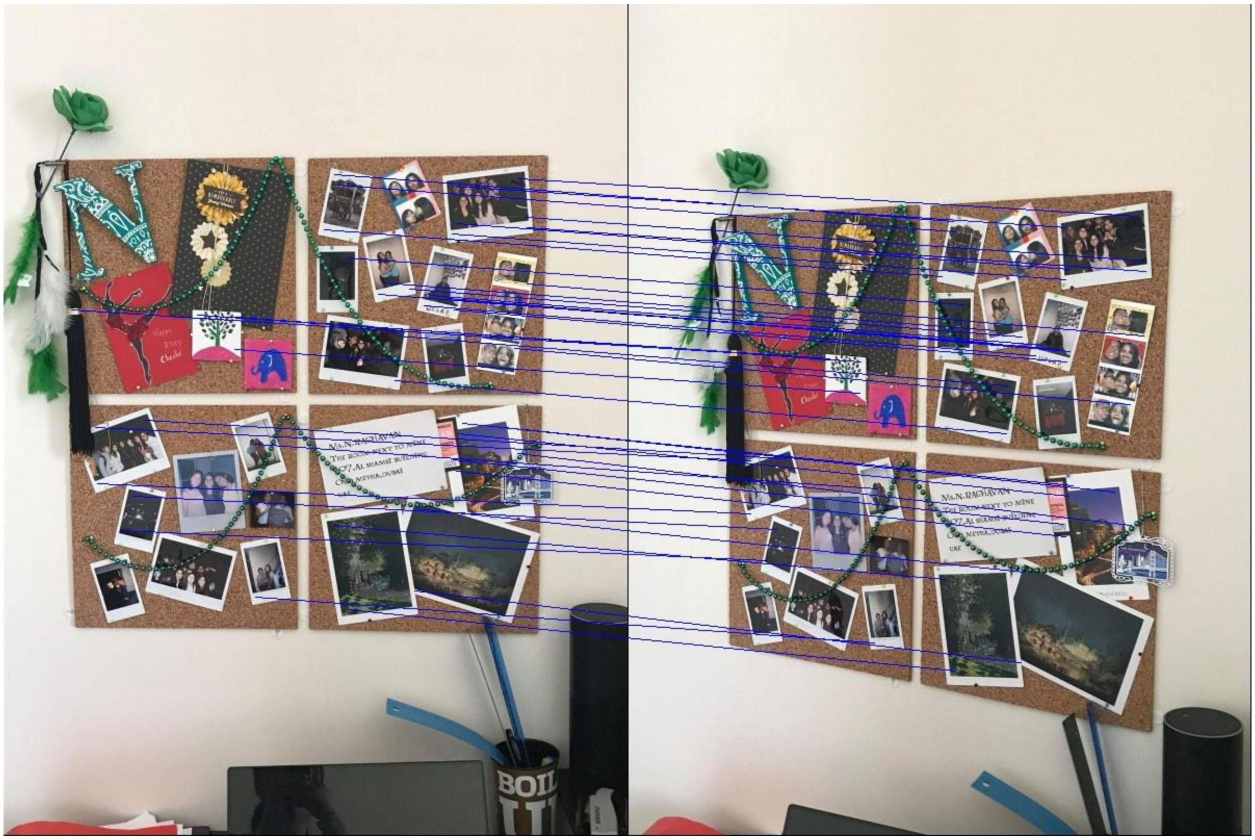

Figure 30: Point correspondence found using SSD in pair 3 at sigma = 3

Namrata Vivek Raghavan
nraghav@purdue.edu

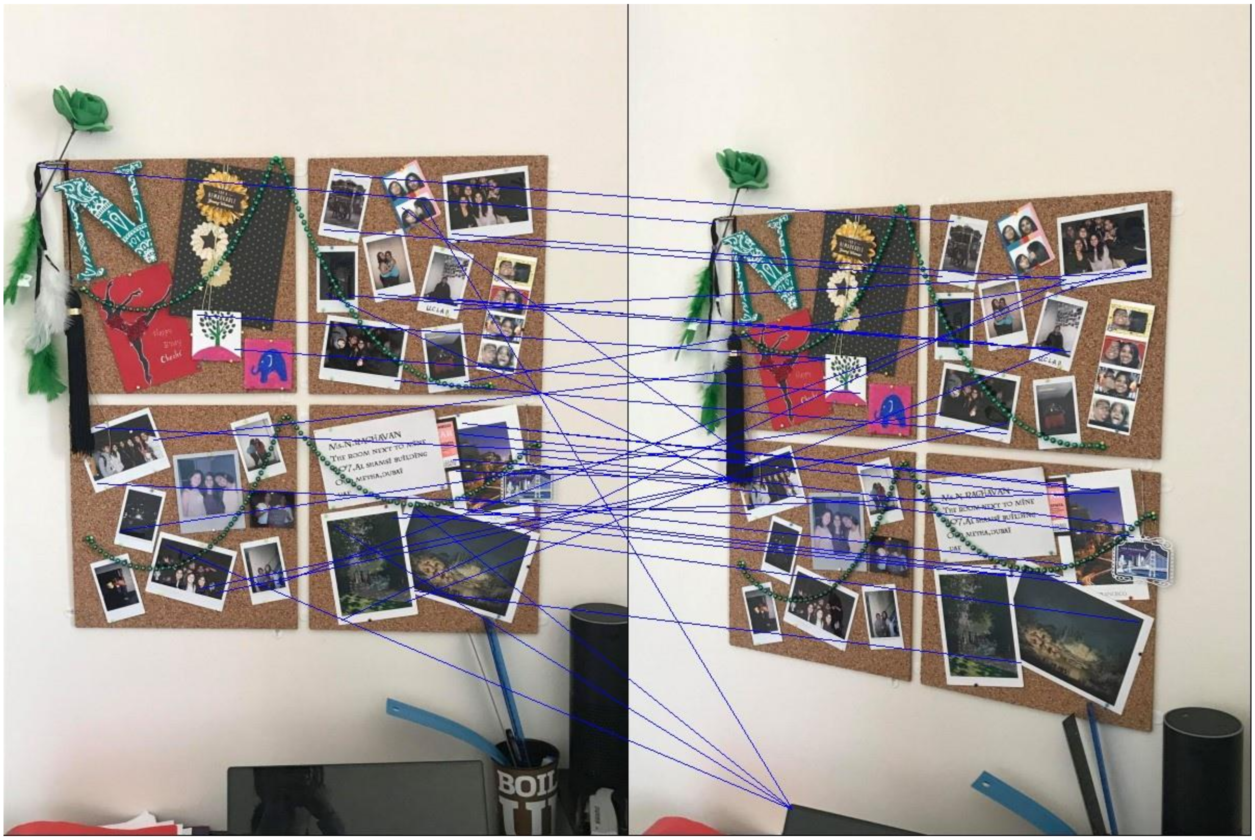

Figure 31: Point correspondence found using NCC in pair 3 at sigma = 3

Namrata Vivek Raghavan
nraghav@purdue.edu

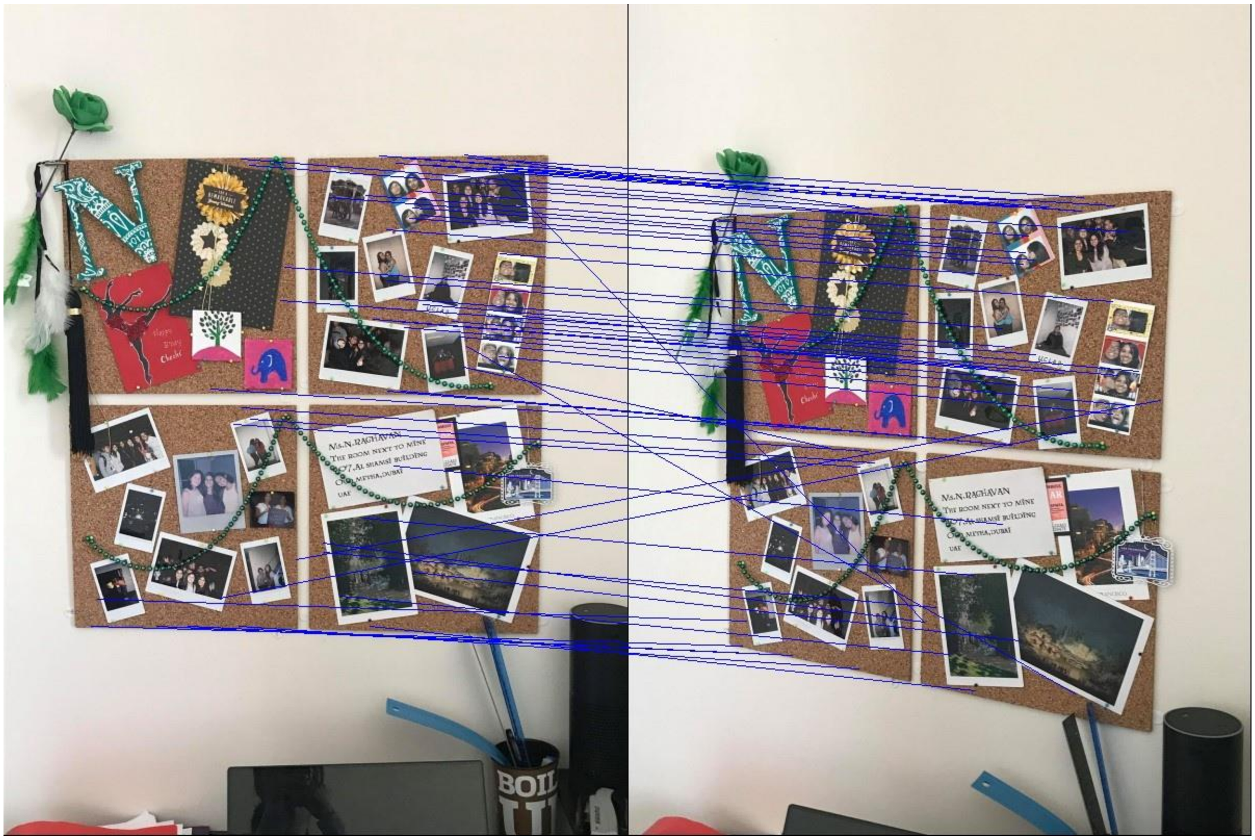

Figure 32: Point correspondence found using SIFT in pair 3

Namrata Vivek Raghavan
nraghav@purdue.edu

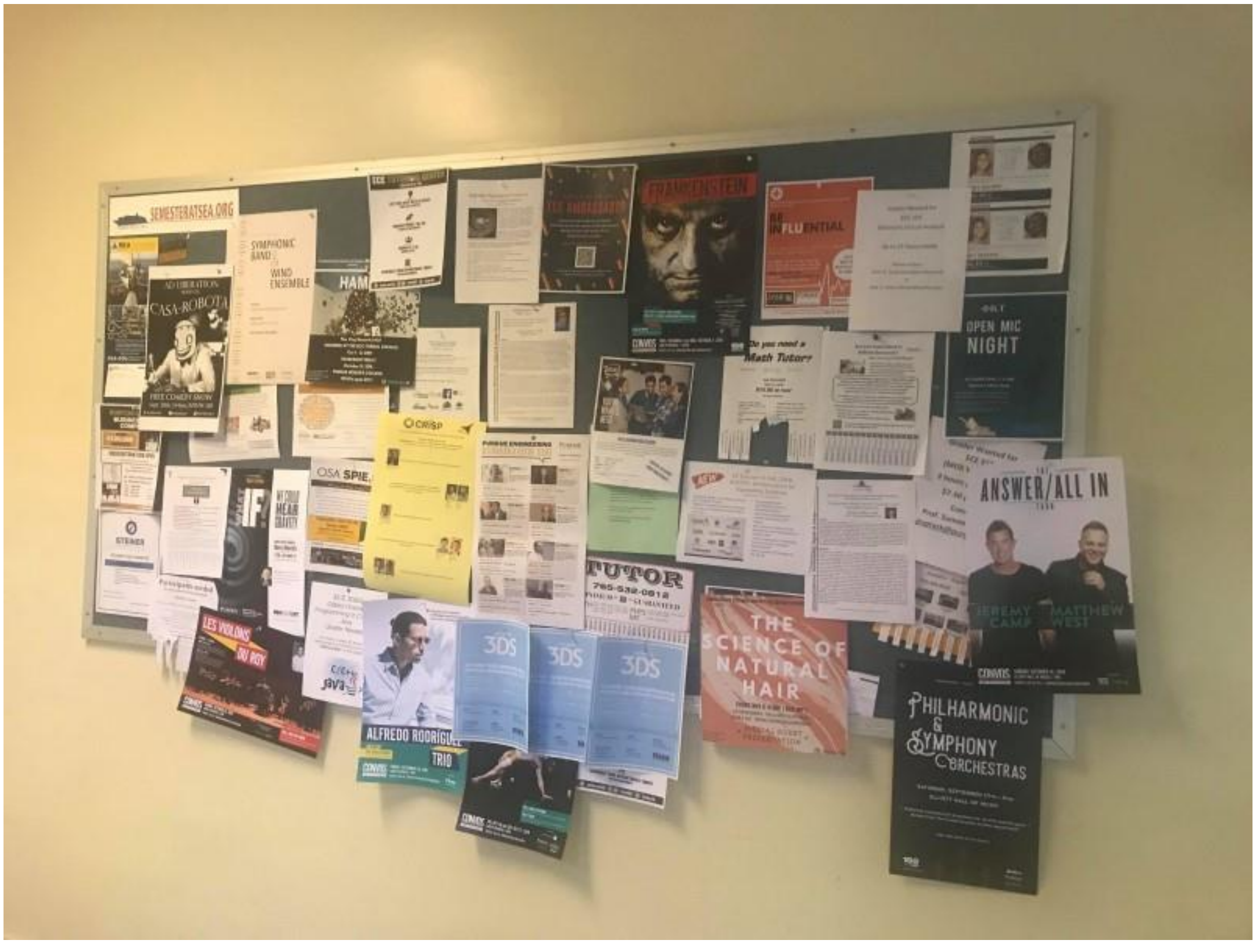

Figure 33: Input image 1 of pair 4

Namrata Vivek Raghavan
nraghav@purdue.edu

Figure 34: Input image 2 of pair 4

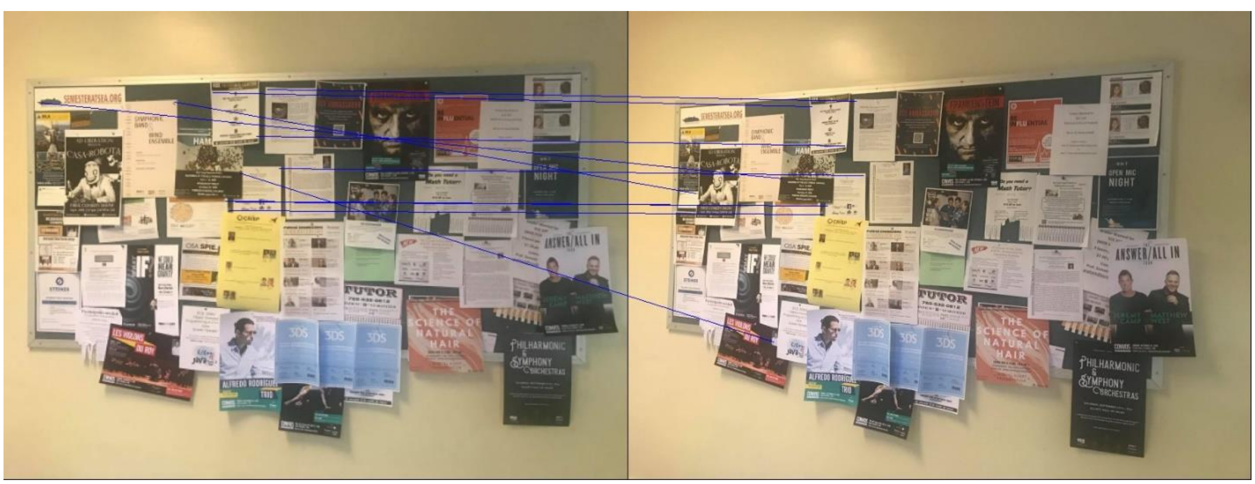

Figure 35: Point correspondence found using NCC in pair 4 at sigma = 1.5

Namrata Vivek Raghavan
nraghav@purdue.edu

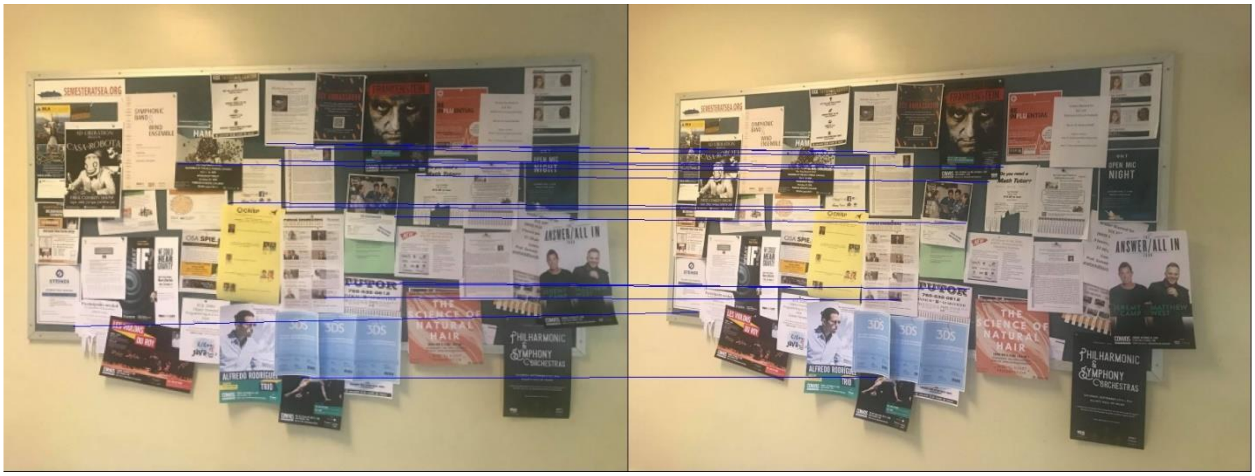

Figure 36: Point correspondence found using SSD in pair 4 at sigma = 1.5

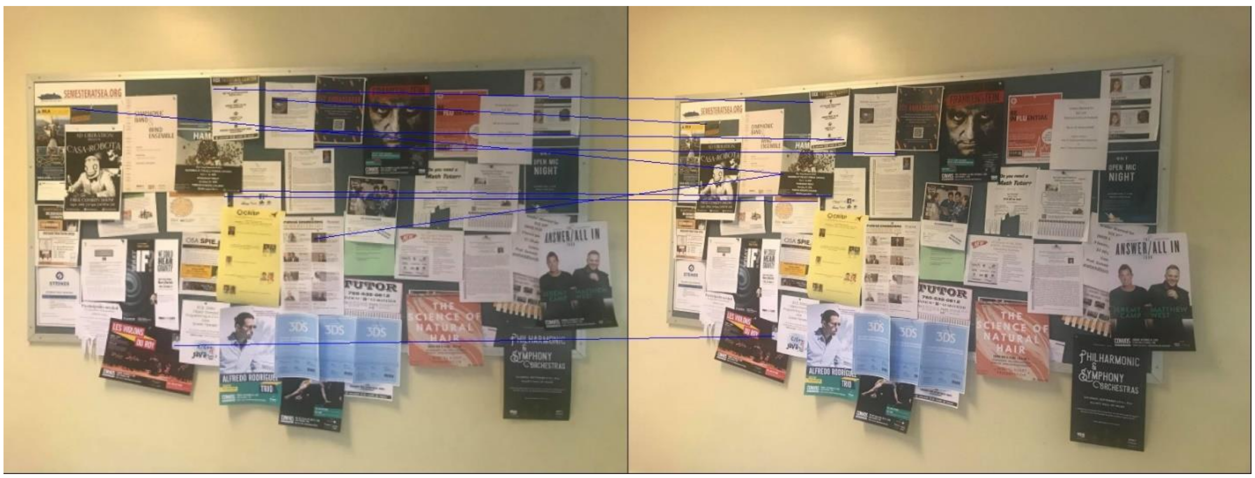

Figure 37: Point correspondence found using NCC in pair 4 at sigma = 2

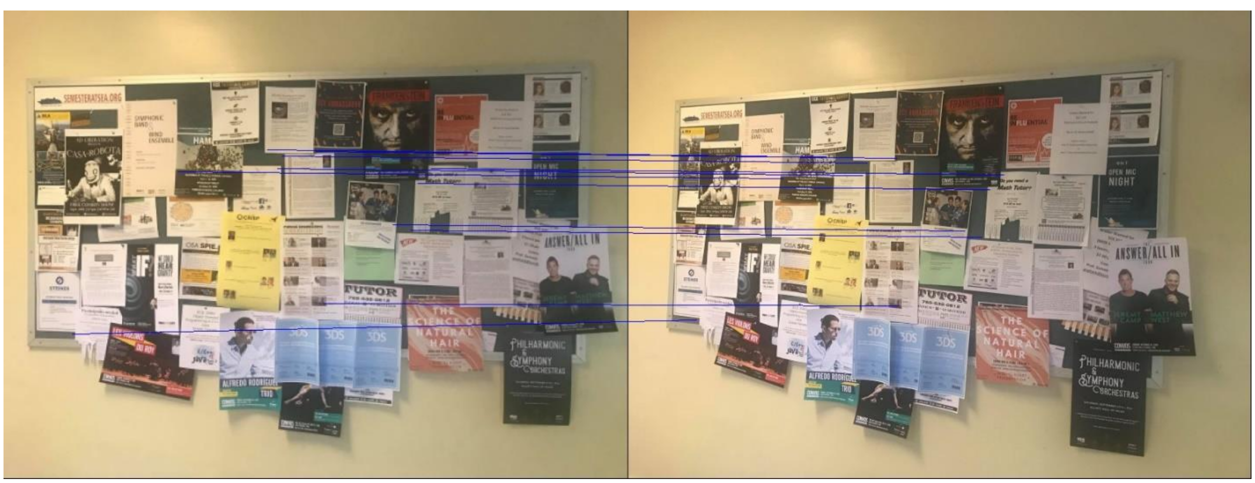

Figure 38: Point correspondence found using SSD in pair 4 at sigma = 2

Namrata Vivek Raghavan
nraghav@purdue.edu

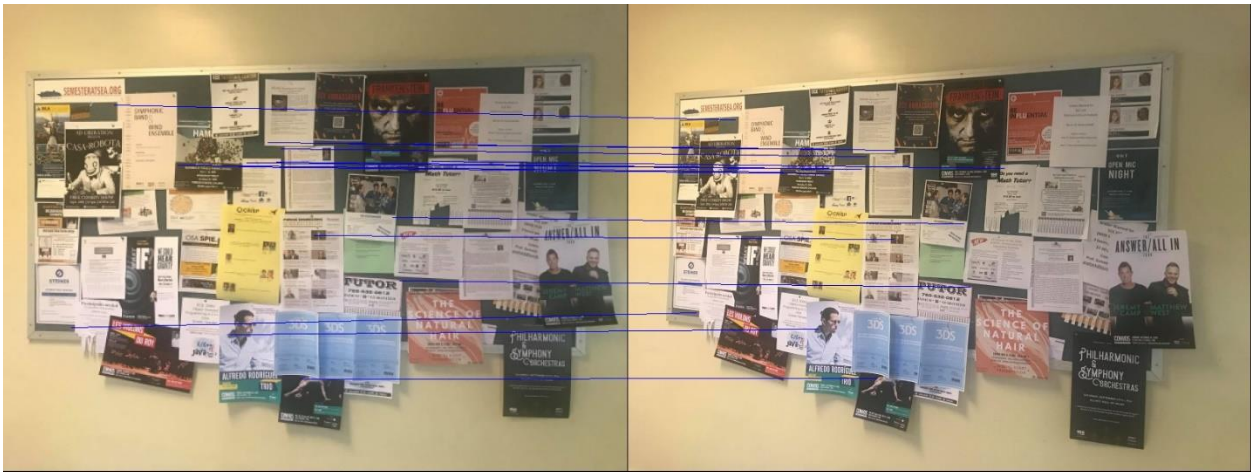

Figure 39: Point correspondence found using SSD in pair 4 at sigma = 2.5

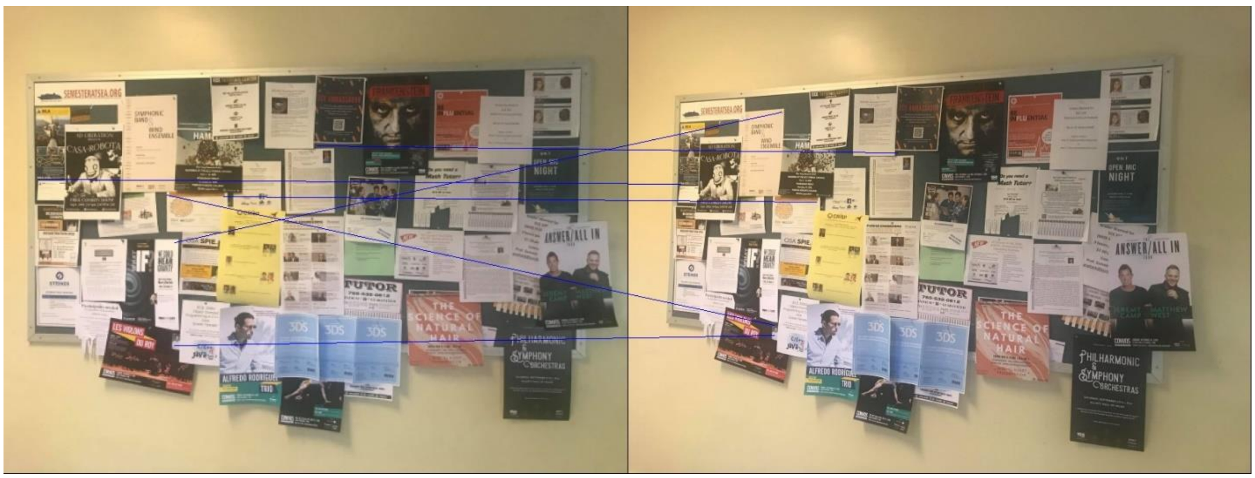

Figure 40: Point correspondence found using NCC in pair 4 at sigma = 2.5

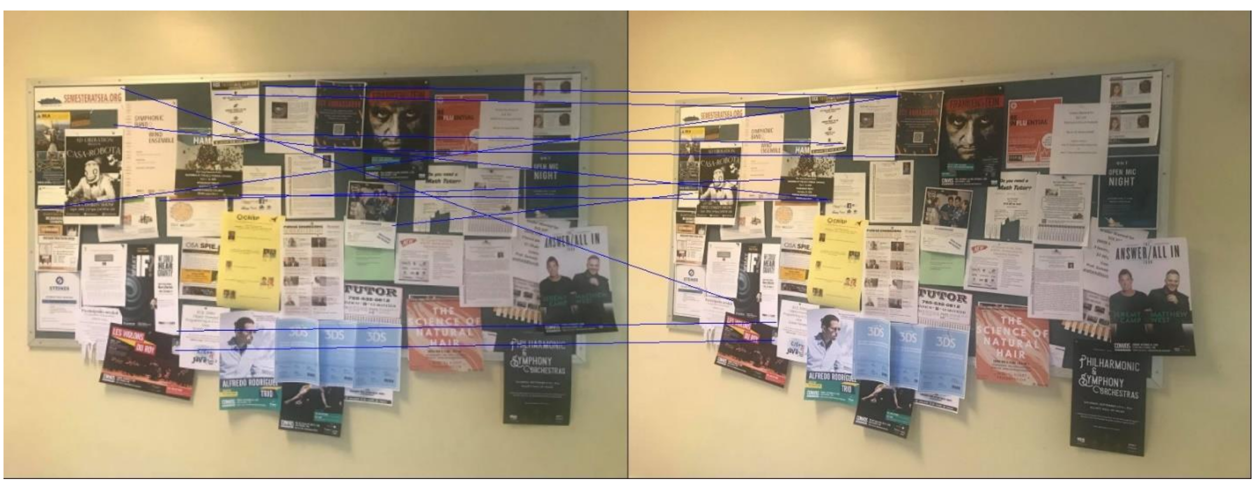

Figure 41: Point correspondence found using NCC in pair 4 at sigma = 3

Namrata Vivek Raghavan
nraghav@purdue.edu

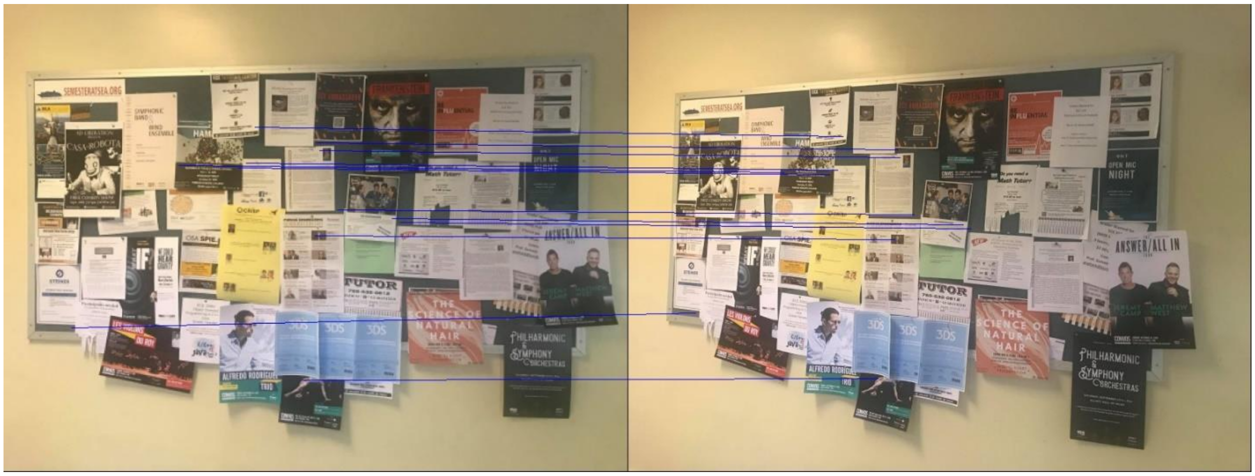

Figure 42: Point correspondence found using SSD in pair 4 at sigma = 3

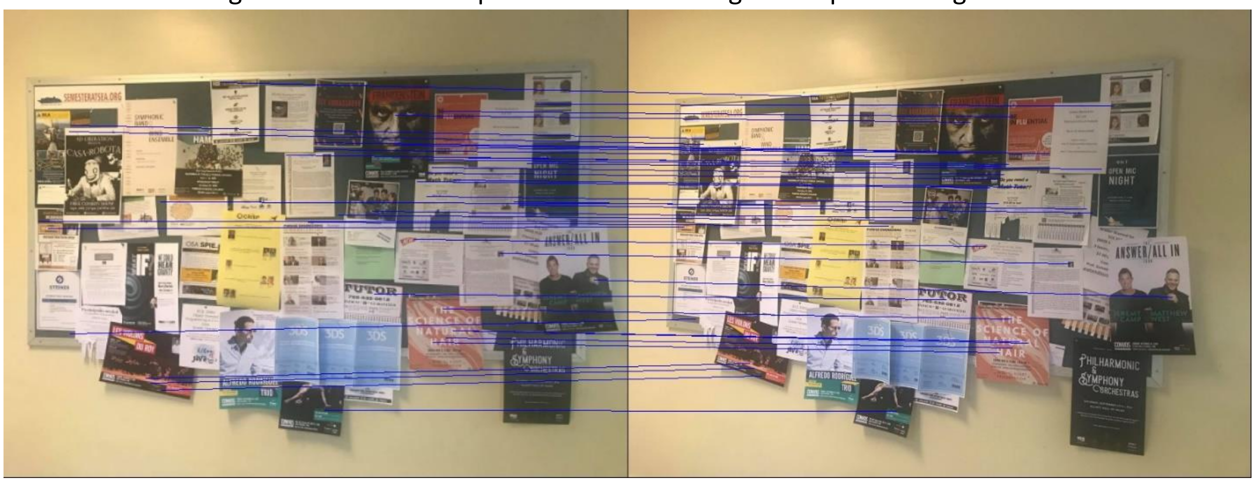

Figure 43: Point correspondence found using SIFT in pair 4