

1 Introduction

In this homework we carry out image mosaicing which consists of stitching together a sequence of overlapping photos of a scene to create a single panoramic photo by:

1. First using the SIFT operator to extract interest points between the image pairs and establishing correspondence between the interest points
2. Next, the RANSAC algorithm is used to eliminate the false correspondences (outliers) between the interest points and the homography between the image pairs is estimated using the noisy true correspondences (inliers) with the Linear-Least Squares Method
3. In the last step, a more accurate Nonlinear Least-Squares Method like the Levenberg-Marquardt Method is used to determine a more precise homography using the homography estimated by the Linear-Least Squares Method as an initial guess of the solution

2 Description of Methods

1. *SIFT Operator*

The SIFT operator has already been discussed and implemented in HW 4 so we will not describe the method here. In this study we use the SIFT operator in OpenCV to identify and match the corresponding interest points. The method **detectAndCompute** takes the gray scale image as an input and identifies the interest points and the 128 element descriptor for each interest point. To match the interest points we use the method **BFMatcher** which calculates the Euclidean distance between the descriptor vectors. For each interest point, we retain the best two matches by using the method **BFMatcher.knnMatch**. Finally for these two best matches, we use the distance ratio test described by Lowe in the original SIFT paper ("Distinctive Image Features from Scale-Invariant Keypoints", D.G. Lowe) which mentions to reject the matches if the ratio of the Euclidean distance is greater than 0.8.

2. *Random Sample Consensus (RANSAC) Algorithm*

The interest correspondences identified by the SIFT operator can have some false correspondences or outliers along with noisy correspondences which are inliers. If we automate the homography detection method based on the correspondences determined by the SIFT operator, there is possibility that the false correspondences might be used for determining the homography. To avoid this we use an outlier rejection algorithm called RANSAC.

In the RANSAC algorithm, the least number of pairs of correspondences required to construct a homography using the Linear-Least Squares Method is randomly selected from the entire dataset of pairs of interest points n_{total} identified by the SIFT operator and then the inlier support for the constructed homography is determined. This is repeated for N number of trials and the homography estimate which has the maximum inlier support is the final estimate of the homography based on the Linear-Least Squares Method. The steps involved in the RANSAC are described below.

(a) First we need to set up the following parameters in RANSAC:

- δ = Decision threshold to determine if a data point lies in the inlier set
- p = Probability that at least one of the N trials. Here we use $p = 0.999$.
- n = number of correspondences used to calculate homography. Here we use $n = 6$.

- ϵ = Probability that a randomly selected correspondence is an outlier. Here we use $\epsilon = 0.4$.
 - N = Total number trials for estimating the homography. This is given by $N = \frac{\ln(1-p)}{\ln[1-(1-\epsilon)^n]}$.
 - M = Minimum value of the size of the inlier set which can be considered acceptable. Here we use $M = n_{total}(1 - \epsilon)$.
- (b) In the second step, for each of the N trials we randomly selected n pairs of correspondences of interest points and use the Linear-Least Squares Method to obtain an estimate of the homography H . If (\vec{X}_i, \vec{X}'_i) is the i^{th} pair of interest points where $\vec{X}_i = (x_i, y_i, w_i)$ and $\vec{X}'_i = (x'_i, y'_i, w'_i)$ then we setup,

$$\begin{bmatrix} 0 & 0 & 0 & -w'_1x_1 & -w'_1y_1 & -w'_1w_1 & y'_1x_1 & y'_1y_1 & y'_1w_1 \\ w'_1x_1 & w'_1y_1 & w'_1w_1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1w_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & -w'_nx_n & -w'_ny_n & -w'_nw_n & y'_nx_n & y'_ny_n & y'_nw_n \\ w'_nx_n & w'_ny_n & w'_nw_n & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n & -x'_nw_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = 0, \quad (1)$$

where the homography H is given by,

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (2)$$

The solution to Eq. 1 is a linear least squares minimization problem $\min \|A\vec{h}\|$ subject to $\|h\| = 1$. This is solved using SVD. The solution is the eigen vector corresponding to the smallest eigen value of AA^T which is the last column in the matrix V where SVD gives $A = UDV^T$.

- (c) Next we compute the distance between the true observation \vec{X}'_i , and the estimated value of \vec{X}'_i using Linear-Least Squares Method given by $\widetilde{\vec{X}}'_i = H\vec{X}_i$. If this distance is less than the decision threshold δ then this correspondence lies in the inlier set.
- (d) We repeat Steps (b) and (c) for N trials and select the homography H which has the maximum inlier support i.e. maximum number of inliers within δ .

This final homography obtained using the RANSAC algorithm and Linear-Least Squares method is still not very accurate since it is estimated based on noisy correspondences which have localization error arising from locating pixel position in the original image from an octave level high-up in the DoG pyramid in SIFT. To get a more accurate estimate of the homography from the homography obtained using this method, we use a Nonlinear Least-Squares method. In this study we use the Levenburg-Marquardt Nonlinear Least-Squares Method which is described in the next section.

3. *Levenburg-Marquardt (LM) Nonlinear Least-Squares Method*

The LM method combines the best of the Gradient-Descent (GD) Method with the best of the Gradient-Newton (GN) Method. The principal equation for the LM Method can be

represented by,

$$\left(J_{\vec{f}}^T J_{\vec{f}} + \mu I \right) \vec{\delta}_p = J_{\vec{f}} \vec{\epsilon}(\vec{p}_k) \quad (3)$$

where $\vec{f}(\vec{p})$ is given by

$$\vec{f}(\vec{p}) = \begin{bmatrix} f_1(\vec{p}) \\ f_2(\vec{p}) \\ \dots \\ f_{2N-1}(\vec{p}) \\ f_{2N}(\vec{p}) \end{bmatrix} = \begin{bmatrix} \frac{h_{11}x_1+h_{12}y_1+h_{13}}{h_{31}x_1+h_{32}y_1+h_{33}} \\ \frac{h_{21}x_1+h_{22}y_1+h_{23}}{h_{31}x_1+h_{32}y_1+h_{33}} \\ \dots \\ \frac{h_{11}x_N+h_{12}y_N+h_{13}}{h_{31}x_N+h_{32}y_N+h_{33}} \\ \frac{h_{21}x_N+h_{22}y_N+h_{23}}{h_{31}x_N+h_{32}y_N+h_{33}} \end{bmatrix} \quad (4)$$

$$\vec{p} = [h_{11} \quad h_{12} \quad h_{13} \quad h_{21} \quad h_{22} \quad h_{23} \quad h_{31} \quad h_{32} \quad h_{33}] \quad (5)$$

and is a vector used to represent the mapping of a physical point X in the domain plane to a physical point X' in the range plane using the homography H . It should be noted that X' is an estimate of the point \tilde{X}' in the range plane. In Eq. (3) $J_{\vec{f}}$ is the Jacobian with respect to components of \vec{p} , $\vec{\epsilon}(\vec{p}_k) = \tilde{X}' - \vec{f}(\vec{p}_k)$ represents the error in the estimation of \tilde{X}' at the k^{th} iteration, μ is the damping coefficient, I is the identity matrix and $\vec{\delta}_p$ is change in \vec{p}_k after one iteration so the value of \vec{p} at the $k+1^{th}$ iteration is $\vec{p}_{k+1} = \vec{p}_k + \vec{\delta}_p$. The following steps are involved in the LM Method:

- (a) First we estimate an initial value of the damping coefficient μ_0 . This is done using $\mu_0 = \tau \cdot \max\{\text{diag}(J_{\vec{f}}^T J_{\vec{f}})\}$ where τ is constant between 0 and 1. Here we assume $\tau = 0.5$.
- (b) Next we calculate $\vec{\delta}_p$ by taking the inverse of Eq. (3) i.e $\vec{\delta}_p = \left(J_{\vec{f}}^T J_{\vec{f}} + \mu I \right)^{-1} J_{\vec{f}} \vec{\epsilon}(\vec{p}_k)$
- (c) In the next step to check the quality of the computed value of $\vec{\delta}_p$ we compute the ratio $\rho^{LM} = \frac{C(\vec{p}_k) - C(\vec{p}_{k+1})}{\vec{\delta}_p^T J_{\vec{f}}^T \vec{\epsilon}(\vec{p}_k) + \vec{\delta}_p^T \mu_k I \vec{\delta}_p}$. Here $C(\vec{p}_k)$ is the cost function at the k^{th} iteration and is given by $C(\vec{p}_k) = \|\vec{X} - \vec{f}(\vec{p}_k)\|^2$ where $\|\cdot\|$ represents the vector norm and $\|\vec{A}\|^2 = \vec{A}^T \vec{A}$. The value of ρ^{LM} obtained in this step is used to calculate the value of the damping coefficient for the next iteration.
- (d) The value of the damping coefficient for the next iteration is given by $\mu_{k+1} = \mu_k \cdot \max\{\frac{1}{3}, 1 - (2\rho^{LM} - 1)^3\}$. The value of the damping coefficient calculated using this formulation steers the LM method correctly either in the direction of the GD method or the GN method.
- (e) The above 4 steps are repeated till $C(\vec{p})_k$ stops decreasing indicating convergence. In the current implementation 100 iterations were found to be more than enough for achieving convergence. The homography from the LM method can now be constructed using \vec{p}_k

The LM method has been implemented from scratch in this study. In the next section we discuss how the Image Mosaicing is done to get a panoramic image.

3 Image Mosaicing

For the five images we calculate the homography H_{ij} between the i^{th} image and the j^{th} image. Let these homographies be $H_{12}, H_{23}, H_{34}, H_{45}$. To generate a panoramic image, we need to map all

images to the plane of the center Image 3. For Images 2 and 4 this is straightforward by using H_{23} and H_{34} . For mapping Images 1 and 5 we need to know H_{13} and H_{53} . This is calculated using

$$H_{13} = H_{12}H_{23} \quad (6)$$

$$H_{53} = H_{54}H_{43} \quad (7)$$

With $H_{13}, H_{23}, H_{34}, H_{53}$ we can now generate the panoramic image. In the following section we discuss the implementation in the code.

4 Implementation in Code

The code has been implemented in the file **hw5_TejasPant.py**

1. In the first step, all the 5 images are loaded using **cv2.imread** and converted into its grayscale form using **cv2.cvtColor**
2. Next we identify the interest points and obtain the 128 element descriptor for each interest point. This is done in the method **SIFTDetectInterestPoints**. This method uses the built-in functions **xfeatures2d.SIFT_create** and **sift.detectAndCompute** in OpenCV to extract the interest points
3. Once the interest points are determined in the two images the correspondence between the interest points is established in the method **BruteForceMatcher**. This method uses the built-in methods **cv2.BFMatcher** and **bf.knnMatch** to determine the correspondence.
4. To display a combined image pair of images with correspondences we need to get the XY coordinates of the keypoints. This is done in the method **getXYCoordinateInteresPoints**. The combined image with all the pair of correspondences is plotted in the method **displayImagewithInterestPoints**.
5. Next, we eliminate the false correspondences using the RANSAC outlier rejection algorithm. This algorithm has been implemented in the method **RANSAC**. The **RANSAC** method calls the **LinearLeastSquaresHomography** method to calculate the Linear-Least Squares homography using SVD. The number of inliers are counted in the method **getInliersCount**.
6. Once the Linear-Least Squares homography with the maximum inlier support is calculated using the method **RANSAC**, we move ahead to refine it using the LM Non-linear Least Squares Method. This has been implemented in the method **Levenberg-Marquardt_Method**. In this method, the jacobian is calculated using the method **getJacobian** which takes the homography H_k at the k^{th} iteration and the true value \tilde{X}' . To calculate $\vec{f}(\vec{p}_k)$ the method **applyHomography** has been defined.
7. After the homography has been refined using the LM method, we determine the size of the final panoramic image. To determine the final size, we determine the maximum X and Y coordinate of each image mapped into the plane of Image 3 using the method **ImageExtent**.
8. In the final step we copy the pixel values of each image into the single panoramic image in the method **getPanaromicImage**. Here we use bilinear interpolation implemented in the method **BilinearInterpforPixelValue** to determine the pixel value of the mapped pixel in the panoramic image.

5 Parameters Selected

The final parameters selected for this study are:

SIFT Operator

1. nfeatures = 5000
2. nOctaveLayers = 4
3. contrastThreshold = 0.03
4. edgeThreshold = 10
5. sigma = 4

RANSAC Algorithm

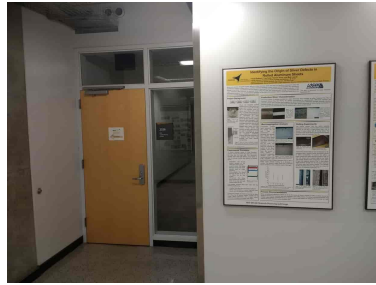
1. $\sigma = 1$
2. $\delta = 3\sigma$
3. $n = 6$
4. $\epsilon = 0.4$
5. $p = 0.999$

LM Method

1. $\tau = 0.5$
2. Maximum number of iterations = 50



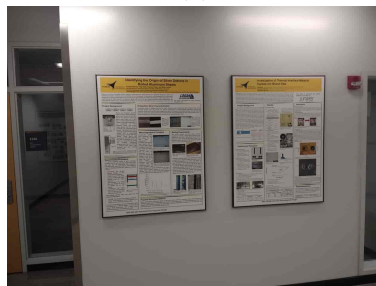
(a)



(b)



(c)



(d)



(e)

Figure 1: Set of 5 images selected for this study.

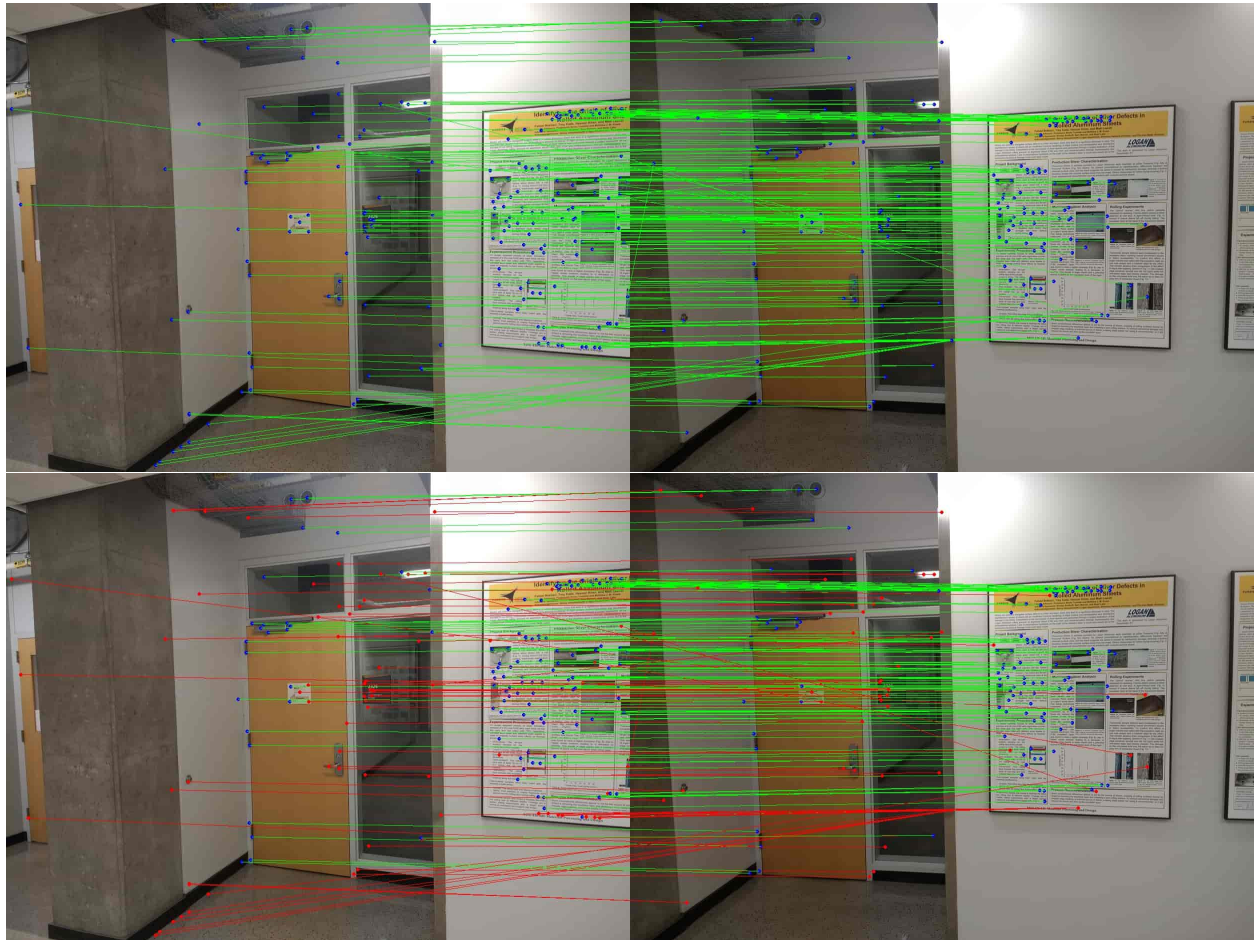


Figure 2: All correspondences detected by SIFT operator (top) and outliers rejected by RANSAC algorithm in red (bottom) for image pair (a)(b).

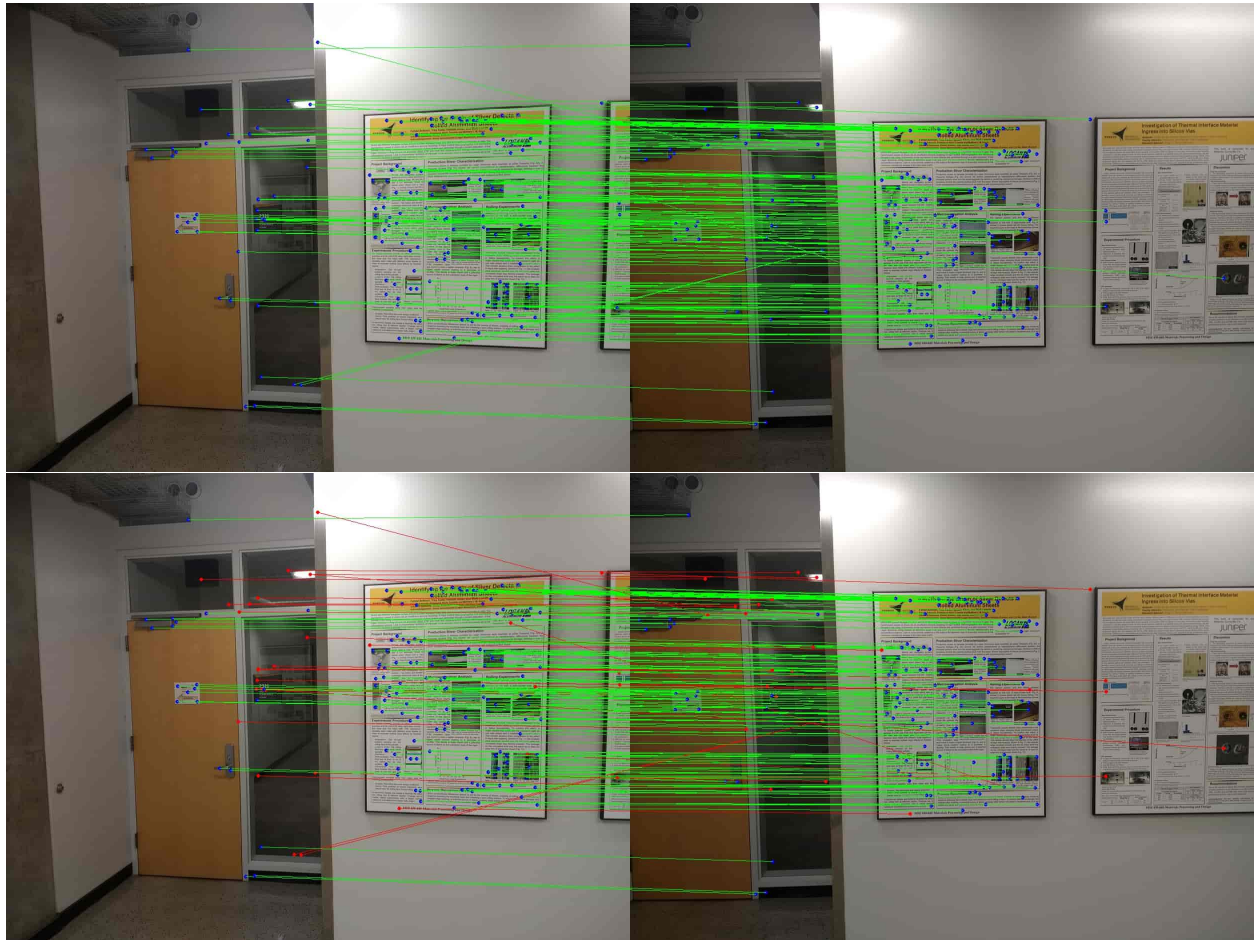


Figure 3: All correspondences detected by SIFT operator (top) and outliers rejected by RANSAC algorithm in red (bottom) for image pair (b)(c).

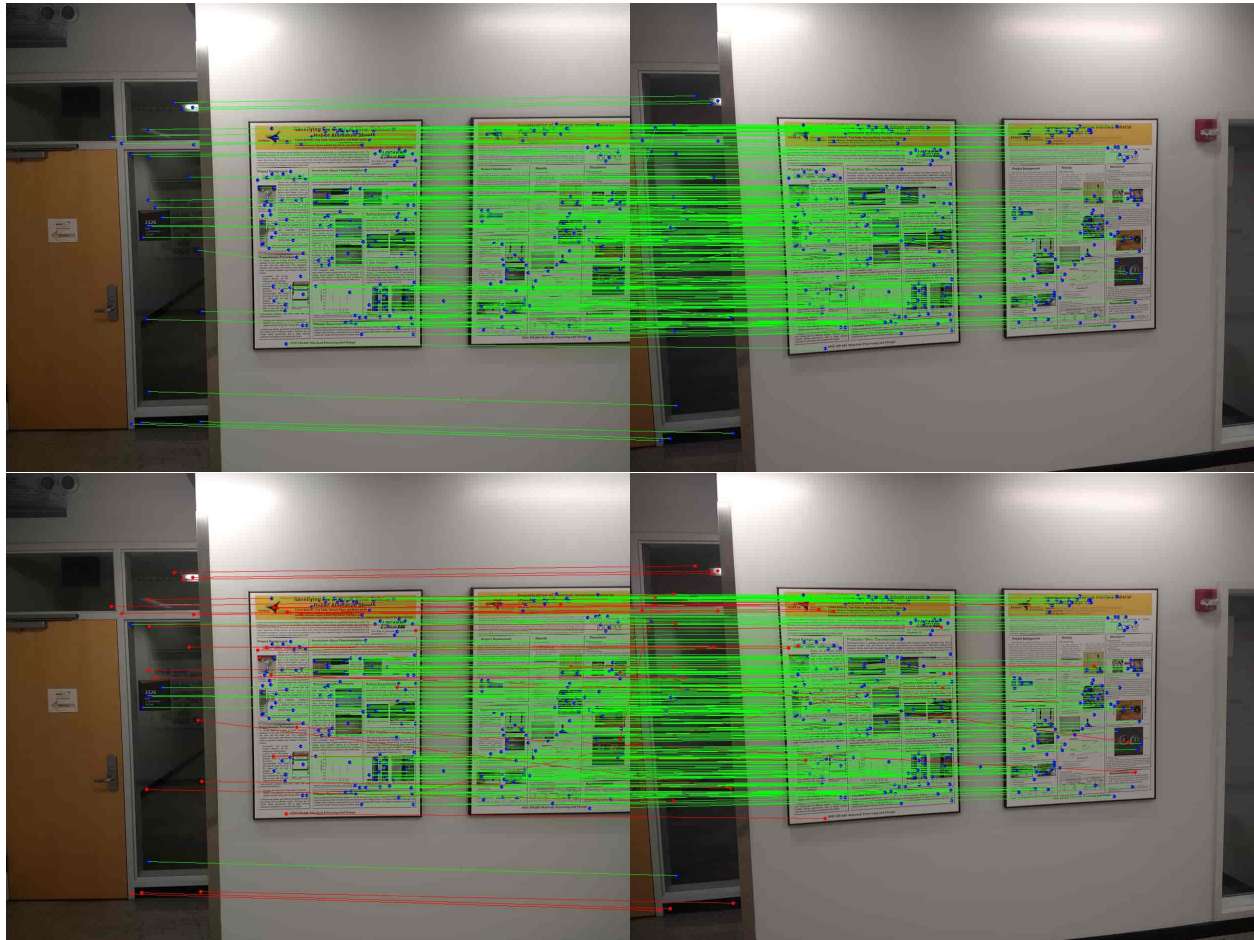


Figure 4: All correspondences detected by SIFT operator (top) and outliers rejected by RANSAC algorithm in red (bottom) for image pair (c)(d).

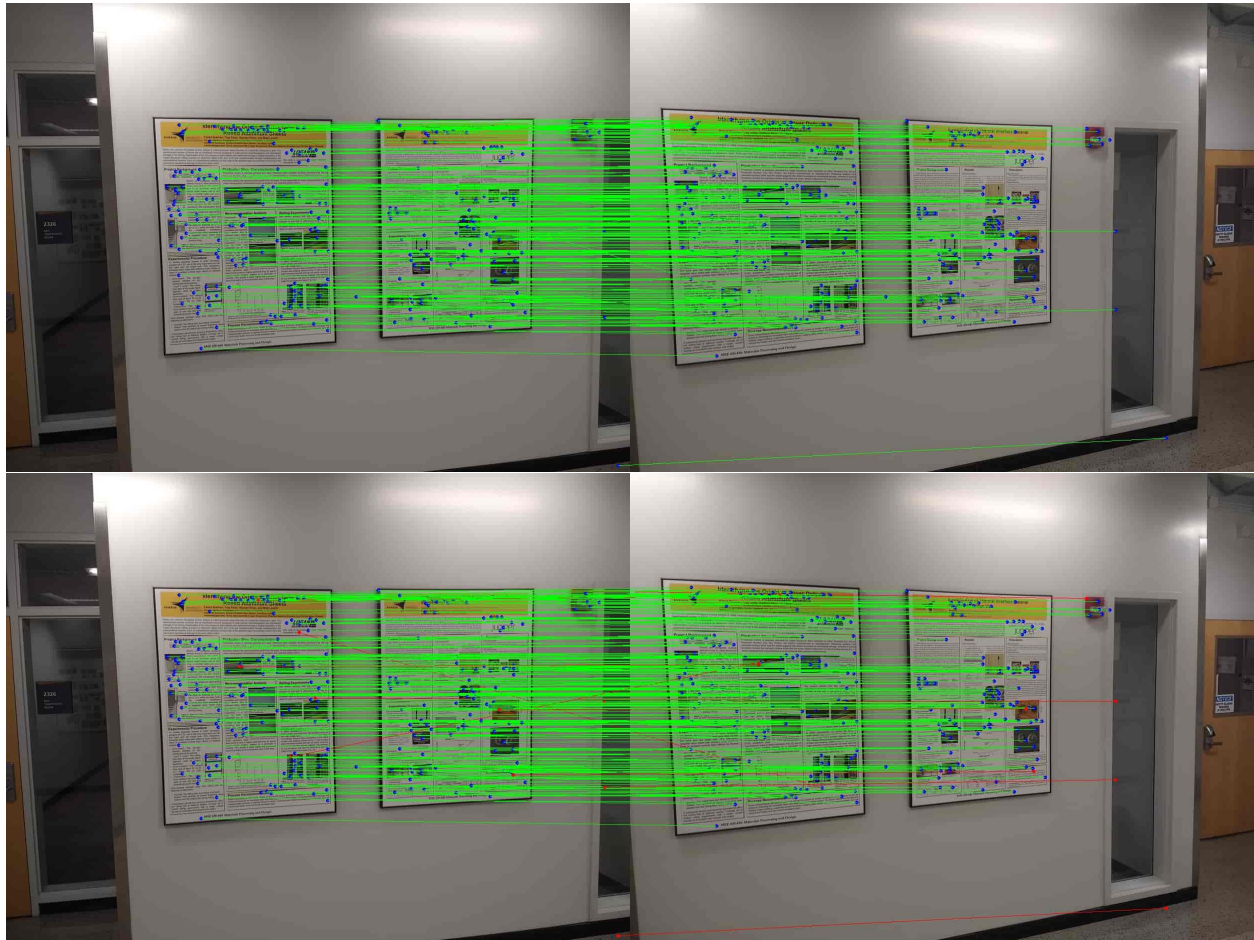


Figure 5: All correspondences detected by SIFT operator (top) and outliers rejected by RANSAC algorithm in red (bottom) for image pair (d)(e).

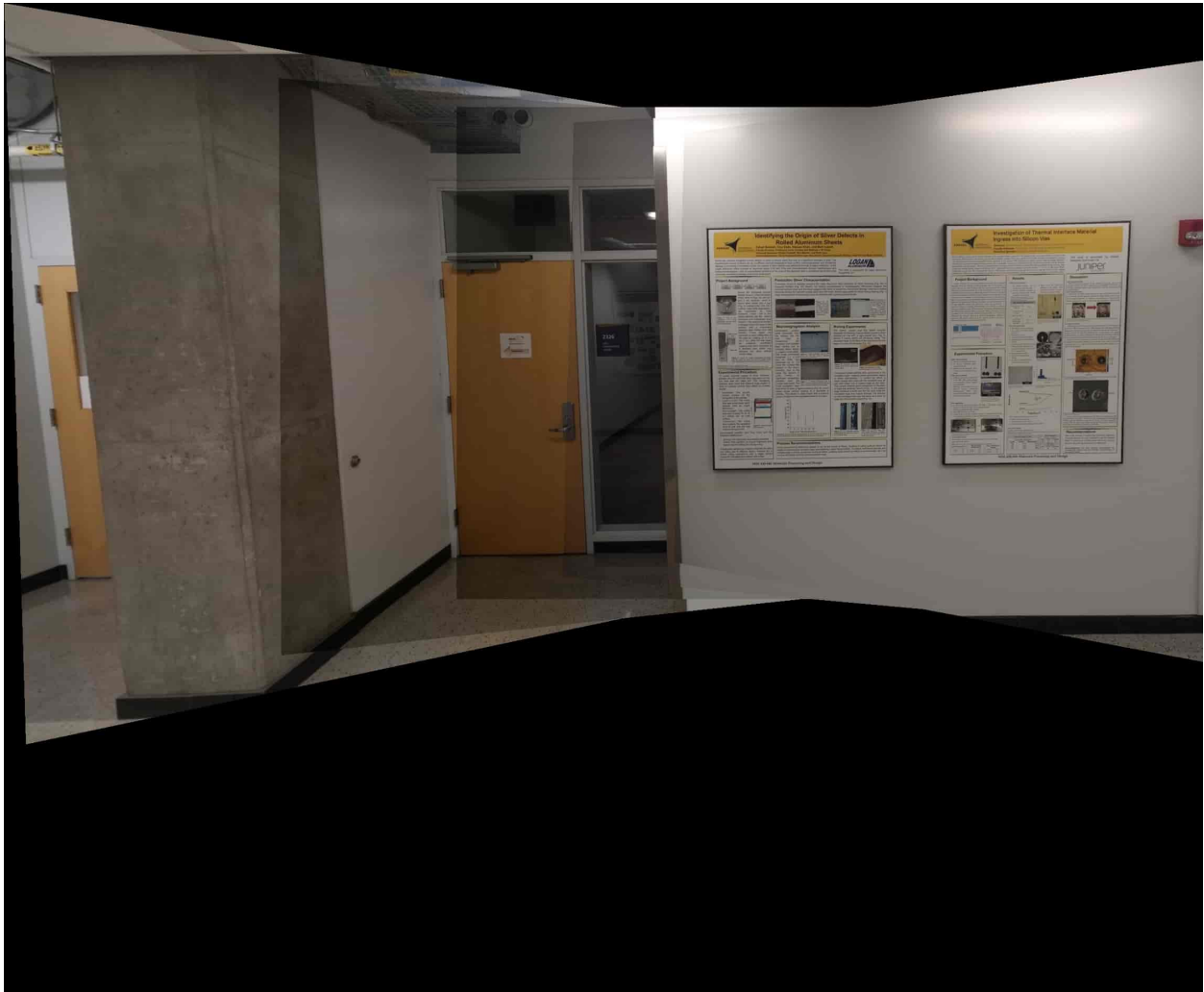


Figure 6: Panoramic image **without using LM method** to refine homographies.

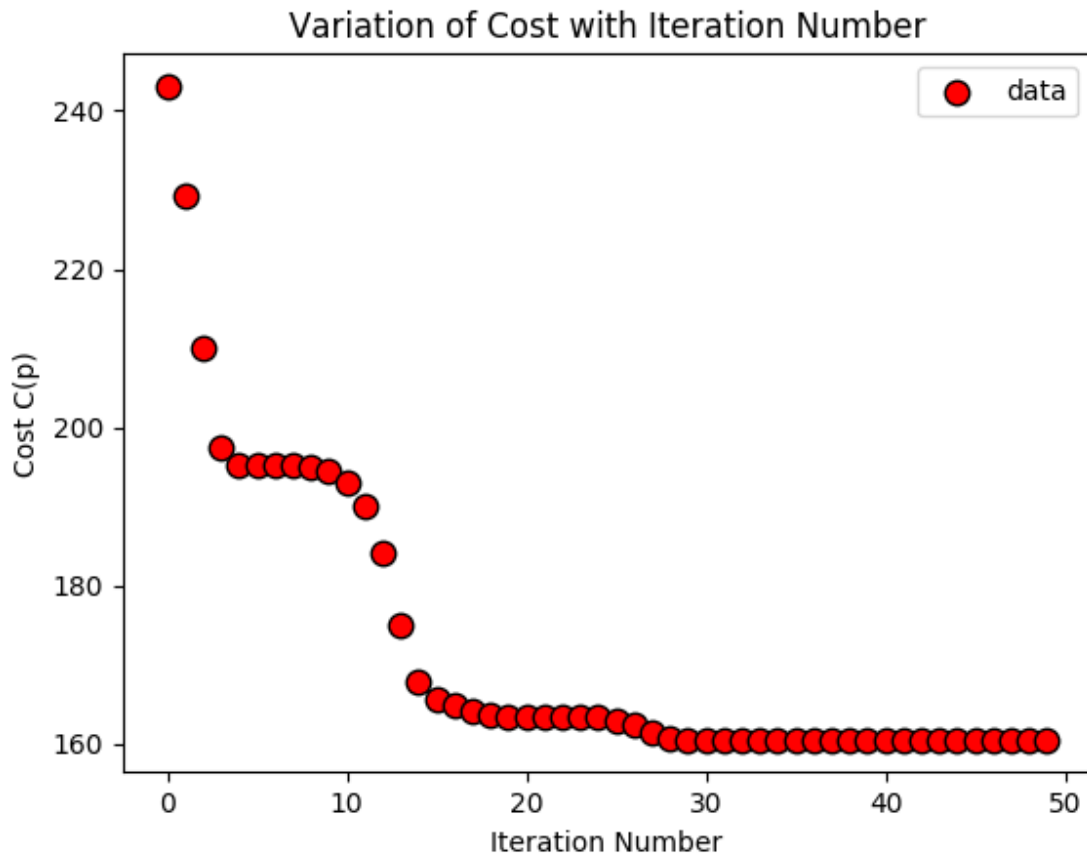


Figure 7: Variation of the cost function $C(\vec{p}_k)$ with the iteration number for the **LM method** for a single homography refinement.

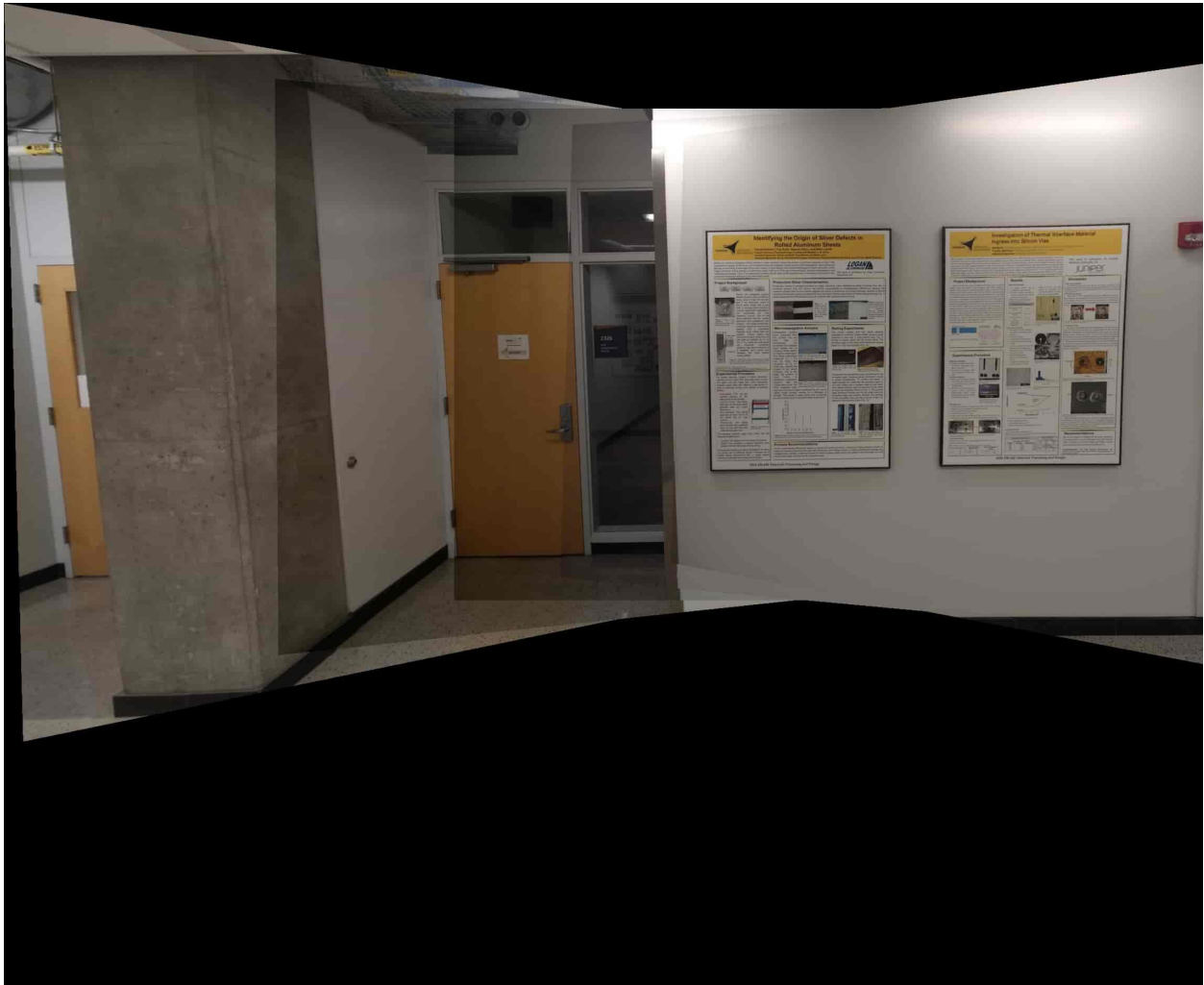


Figure 8: Panoramic image **with LM method** to refine homographies.