Homework Assignment 11

Marium Yousuf
myousuf@email.arizona.edu
Kayla Bennett
mbennett3@email.arizona.edu

May 4, 2022

1 Introduction

In this assigment we implemented the RANSAC algorithm to fit a line, solidified understanding of planar homography, and explored multi-view geometric constraints for keypoint matching. We also looked at the configuration of a **fundamental matrix** and image stitching using SIFT, homography, and RANSAC.

1.1 Contributions

Kayla worked on SIFT visualizations and fundamental matrix (*), while Marium implemented RANSAC algorithm, the Direct Linear Transform method to compute homography between two planar images, and used the homography with RANSAC to improve keypoint matching. Both implemented image stitching using SIFT, homography, and RANSAC together (*). Please note that this assignment is being turned in incomplete. We could not get done with D1 and D2 on time but have started it to submit it with the HW12 (we believe this is what was suggested to do in case if students were not able to get done with HW11 in time).

2 A: RANSAC

For this part, we used a dataset of (x, y) coordinates to fit a line model using a RANSAC algorithm. As provided, at least $1/4$th of the points were assumed to be close to the good fit. Fig. 1 shows the scatter plot of the data points together with the line fit found with an error of 0.0928. The RANSAC algorithm was implemented as follows:

- we computed k, the number of iterations, using the formula $\left\lceil \frac{\log(1-p)}{\log(1-w^n)} \right\rceil$, where p quantifies the probability of success we require from the algorithm, w is the probability of getting an inlier, and n is the number of points needed to fit the model. We set $p = 0.99$, that is we required the algorithm to compute with 99% success rate, $w = 0.6$, that is we wanted the inlier points ratio to be $3/5$ of the data points around the line that fit the model, and $n = 2$ since we only need 2 points to fit a line. This computed the value for our $k = 11$ as the number of iterations.

- using the k found, we excluded picked two random points from the dataset and used the homogenous least squares model to find a line instance. For each of these instances, we computed a perpendicular distance for each of the points using a threshold of 0.2. That is, if for any point the distance was within 0.2 of the line distance, we included that point as an inlier. The value for the threshold was chosen by consulting the scatter plot of the data points.

- if the amount of inliers found exceeded 75 ($1/4$th of our 300 data points), then we assume to have found a good model and fit a line using all the inliers and the two randomly sampled points. The red line shown in Fig. 1 is this line model.

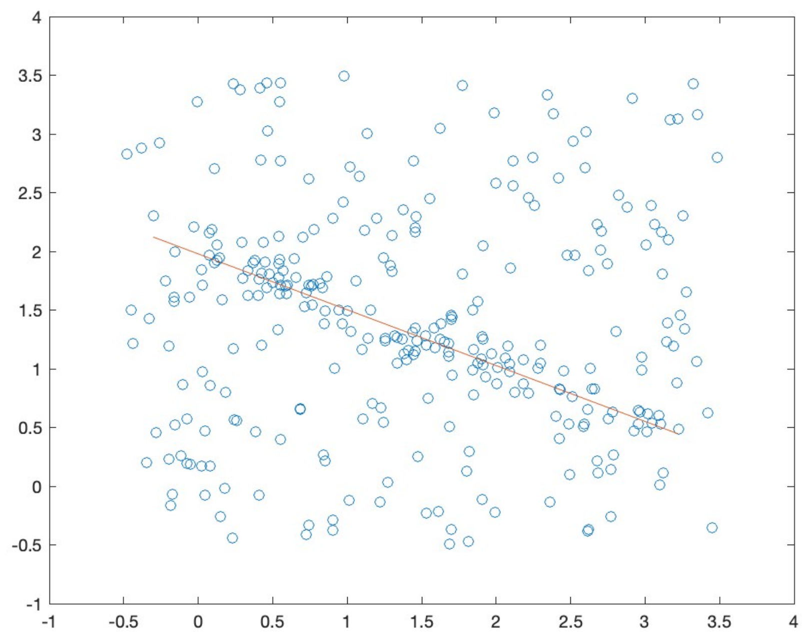


Figure 1: Figure showing the data points of (x, y) coordinates as a scatter plot and a line model fitted using the RANSAC algorithm. This line was found with an error of 0.0928 based on inliers found from the algorithm.

3 B: Homography

For this part, we implemented the Direct Linear Transform (DLT) method and tested it on the synthetic data and the slide/frame image pair data from assignment 9. For the synthetic data, we randomly generated 4, 5, and 6 pairs of (x,y) coordinates, twice: one as "actual" points and one as the "estimate" or the point match. We did this 10 times and yield the RMS errors as shown in Table 1 below. It's a little difficult to tell if the code is working since the errors stay between (0.3, 0.4) (range tested through trial-and-error) for any number of pairs. We conclude that since we only need 4 good matches, the number of pairs is not making much of a difference in the homography found using the DLT method.

Number of Pairs	RMS
4	0.3108
5	0.3235
6	0.3267

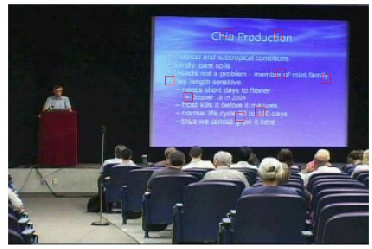
Table 1: RMS error of the transformation for 4, 5, and 6 randomly generated pairs of (x,y)-coordinates lying between a $[0,1]\times[0,1]$ window.

Moving on to the slide/frame image pairs, we collected 8 distinct points using mouse clicks as shown in Fig. **??** for all three slide/frame pairs. Then using 4 of these clicked points, we computed the homography using the DLT method for each slide/frame pair. Using this homography, we estimated the corresponding slide coordinates to the points on the video frame image. The RMS errors yielded for each pair are shown in Table 2, and Fig. **??** shows the visualization on the video frame images, displaying the 8 clicked points in red and the 8 mapped locations in yellow.

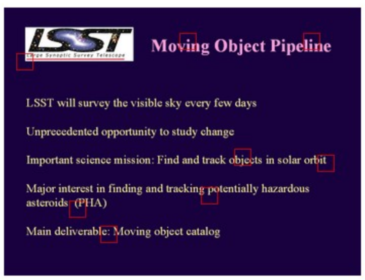
Slide/Frame Pair	RMS
Slide/Frame 1	0.1750
Slide/Frame 2	0.1756
Slide/Frame 3	0.1947

Table 2: RMS error of the difference between the clicked points and the estimated points on the video frame images.

(a) Slide 1

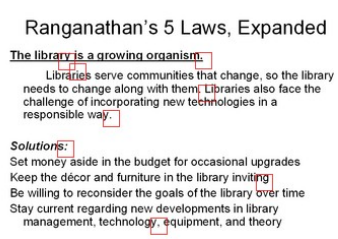


(b) Frame 1

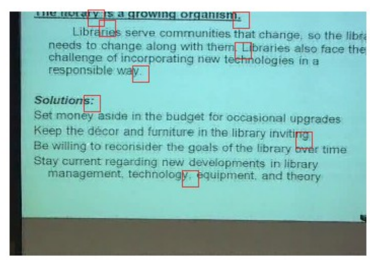


(c) Slide 2

(d) Frame 2

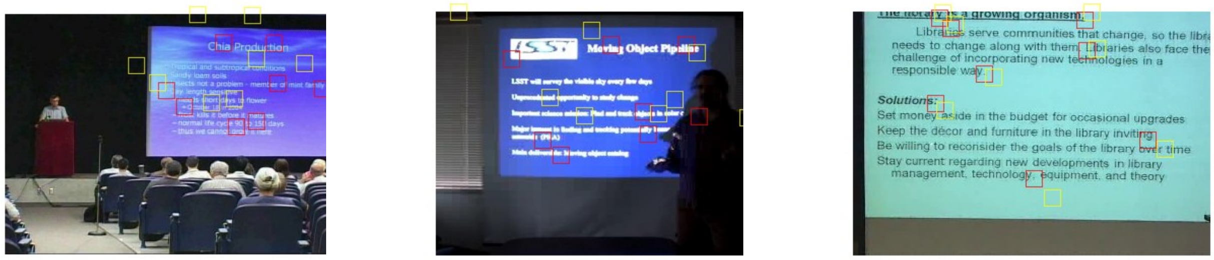


(e) Slide 3



(f) Frame 3

Figure 2: Figure showing collected 8 matching points using manual mouse clicking. The points were chosen by focusing on distinct locations that were least ambiguous.



(a) Estimated points for Frame 1

(b) Estimated points for Frame 2

(c) Estimated points for Frame 3

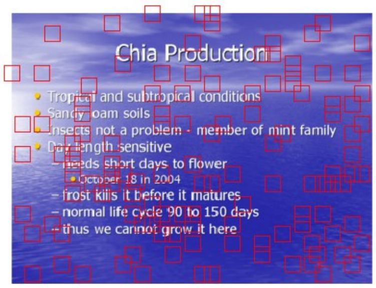
Figure 3: Figure showing the clicked points (in red) against the estimated points computed using homography (in yellow).

4 C: Homography and RANSAC

For this part we use RANSAC to compute homography for several points (instead of the mouse-clicks chosen in Part B). For the RANSAC, the number of iterations were computed to be $k = 178$ and the algorithm was designed as follows:

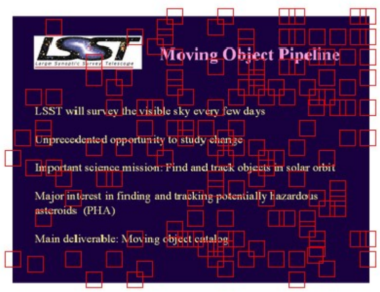
- we used slide image as our data and picked $n = 4$ randomly selected coordinates and used the homography H computed in Part B for the corresponding slide/frame pair as our model
- for other coordinates not previously selected, we used this H to estimate points on the video frame image,
- we randomly selected a point on the video frame image and if the rms error of the difference between randomly selected point and the estimated point was within a chosen threshold of 50, then we selected the random point as our "fit" and the corresponding slide coordinate as the inlier,
- we repeat this for all iterations and until we have total $N = 95$ keypoints.

Fig. **??** shows the mapping of keypoints from slide image to the corresponding video frame.



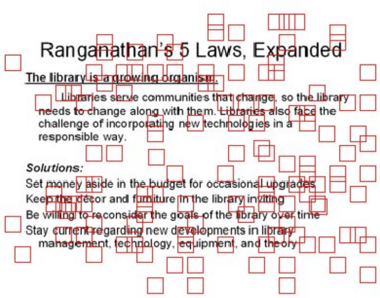
(a) Slide 1

(b) Frame 1

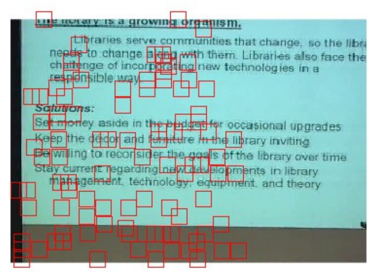


(c) Slide 2

(d) Frame 2



(e) Slide 3



(f) Frame 3

Figure 4: Figure showing the mapping of 95 keypoints for the provided slide-frame pairs. These keypoints were generated using the homography with RANSAC algorithm. (Apologies for the bad visualization, the draw_segment was breaking for me, and I couldn't figure it out to show the mapping using lines).

5 D1: Configuring a fundamental matrix

(Will be turned in with HW12)

6 D2: Image Stitching

(Will be turned in with HW12)