Assignment 12 (Group)

Pratik Bhandari and Simon Swenson
pratikbhandari@email.arizona.edu
simonswenson@email.arizona.edu
Graduate Students

May 1, 2019

Group Contribution

For the assignment, the two of us worked together for each question. Every question was brainstormed and discussed with both members being involved. Similarly, the coding was done by both members in parts, one adding on to the work of the other and vice versa. The report was also worked on together as a team by using *Overleaf* as a platform for both team members to work concurrently on the report.

Part A

The first part of the assignment dealt with implementing a simple stereo processing. Just by the optical illusion effect, we figured out that the first image pair displayed an o in the center and the second image pair displayed a pi in the center.

To reason more concretely about what we were seeing, we used the simple stereo model. We considered the left image to be the "source" image, and, for each pixel in that image (excluding some pixels on the edges for padding), we attempted to find a matching pattern along the same row in the right image. Finding such a match would give us a disparity value, the offset between the pattern in the two images. We implemented this by traveling row-wise from left to right in the range -15:15, centered at the source pixel. For each pixel in this range, we considered a locality ranging from -10:10, centered on that pixel (in the second image). Basically, by taking the dot product of two vectors centered on each that are long enough to capture the similarity, we tried to answer how well the two points matched. We remembered the offset of the best such match. This way the value of the shift for the best match for each point was recorded and the corresponding disparity match was displayed.

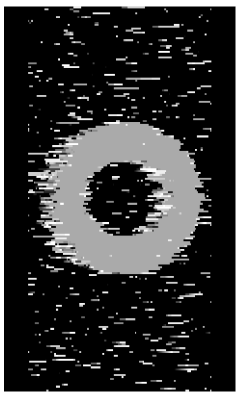

Figure 1: The disparity map between the left and right images in the first image pair. Here, we represent low disparity as dark and high disparity as bright. While there is some interesting noise, the character "o" is clearly visible.

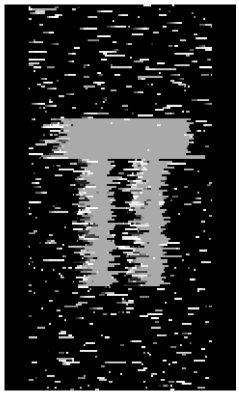

Figure 2: The disparity map between the left and right images in the second image pair. Here, we represent low disparity as dark and high disparity as bright. While there is some interesting noise, the character "pi" is clearly visible.

The images have some interesting noise. This could be caused by two things: (1) patterns with only black pixels, in which case the dot product with any other pattern is always 0 and (2) A pixel's local pattern is repeated more than once, and our algorithm picks the first, not the one with lowest disparity.

In addition to this, the average of the largest 10% absolute value disparities were calculated for both the images. This gave a single interesting distance relative to the viewer. This value of disparity for the first image image pair is 11.5 and that for the second image is 11.86.

Now, supposing that pixels are 0.025cm, and that the focal length is 2cm, and the distance between eyes is 10cm, using the formula developed in class the distance associated with the surface is calculated by:

$$z = f\frac{d-D}{D}$$

Here, $f = 2cm$, $d = 10cm$. We estimated $d - D$ to be d, since d is much larger than D. The value of D will be 0.025 times the disparity average for each image. So, the value of z for the two image pairs are as follows:

- $z_1 = 69.565cm$
- $z_2 = 67.449cm$

Part B

This problem is an application of the equation we covered in the slides:

$$z = f\frac{d-D}{D}$$

We're given that:

- $f = 2m \approx 0.0012427mi = 1.2427 \times 10^{-3}mi$
- $d = 8 \times 10^7 mi$
- $D = 6.0\mu m = 6 \times 10^{-6}m \approx 0.0037282 \times 10^{-6}mi = 3.7282 \times 10^{-9}mi$

Since we've converted everything into miles, we can just plug them into the equation. Also, we ignore the D term on the top, since d dominates it:

$$\begin{aligned} z &= 1.2427 \times 10^{-3}mi\frac{8 \times 10^7 mi}{3.7282 \times 10^{-9}mi} \\ &\approx 1.2427 \times 10^{-3}mi \times 2.1458 \times 10^{16} \\ &\approx 2.6666 \times 10^{13}mi \end{aligned}$$

To convert to light years, we use the conversion factor:

$$1ly \approx 5.88 \times 10^{12} mi$$

So:

$$2.6666 \times 10^{13} mi \frac{1ly}{5.88 \times 10^{12} mi} \approx 0.45350 \times 10ly = 4.5350ly$$

To check our answer, we consulted NASA on the closest stars: Alpha Centauri. NASA confirms that Alpha Centauri A and B are about 4.35 light years away, coinciding closely with our calculation.

Aside: While rounding was carried out between each calculation above, we also performed the calculation in Python, with only double-precision rounding, and got 4.5351 as our answer.

Part C

Simon's comments and ratings:

1. Mostly review. PCA was cool. I covered it way back in undergrad Linear Algebra, but I had mostly forgotten it.
2. Light sensors are useful background knowledge, but it didn't srike me as something super useful. Tweaking lambda was a bit tedious. Kobus hinted that this could be done more automatically with cross validation, but maybe that would be too advanced for this assignment.
3. The first part of the assignment wasn't that interesting to me, maybe because the points were arbitrary, not real data, but line fitting was an important skill for this course. The math proofs were fun, and reasoning about perspective was fun.
4. Picking points is tedious. Error metrics were mostly review for me. The math proofs were neat. This assignment was mainly to set up the next assignment, I think.
5. Now we're seeing the payoff for the tedious point picking last assignment! Seeing the camera results was cool, rendering the sphere was cool, finding the intrinsic and extrinsic camera parameters was a cool endeavor, even though I didn't succeed. Maybe you could provide more guidance on that part for students in future years.
6. Another really neat assignment.
7. no comment
8. I had used GIMP filters before. Now I know how they work, and why it's called a "Gaussian blur."
9. I played around with this one, but didn't complete it. The clustering produced some cool banding effects.
10. I liked these two assignments (10-11) the most. Exploring the effect of the different error metrics, thresholds, and ratios was fun. The tweaking, itself, got a little tedious near the end. Also, we're finally using image feature points, which I was working with in my research last semester.
11. Glad we finally dealt with outliers from the last assignment. The line fitting part of the assignment was an interesting take on k-means, clustering based on distance to a line rather than distance to a centroid.
12. Both parts were straightforward. I felt like the second part was a good basis for actually finding the 3-d coordinates of the points. If we add a new camera, we get a new epipolar line for each point. Then, if we represent those lines in 3-d space, we can just find the intersection.

Assignment	Interest	Usefulness	Length
1	1	2	2
2	4	2	4
3	4	5	4
4	3	4	4
5	5	5	4
6	5	5	4
7	4	4	3
8	5	5	4
9	NA	NA	NA
10	5	5	5
11	5	5	5
12	4	5	3

Pratik's comments and ratings:

1. Mostly a review of MATLAB and its usage. Having used MATLAB in my undergraduate, this was easy for me. I had never done PCA before so that required some extra learning but found it to be interesting.

2. Starting off with laying the groundwork for understanding the basics of computer vision, I liked this assignment. Similar to Simon, I found tweaking lambda a bit tedious.

3. Learnt line fitting which will be useful moving forward in various other fields as well. Reasoning about perspective was interesting too.

4. A little more time consuming assignment because we had to manually pick points in the image. Learnt about the error metrics and its usage though which was a plus.

5. Continuing from the previous assignment's work, sphere rendering and the results of the camera were interesting. The part to find the intrinsic and extrinsic parameters was very challenging.

6. A shorter assignment compared to the previous ones but interesting nonetheless.

7. Learnt about color constancy through examples which was pretty cool to see. The grad questions were interesting in this assignment.

8. By far the most interesting and fun assignment for me since it was very relatable in the sense of how I'd been using Gaussian Filters to edit images for so long and now understood the actual concept of it and made one myself.

9. This was another interesting assignment. It was tougher than the previous one but the results were cool. Being more interested in this aspect of computer vision, I liked the assignment a lot.

10. Assignments 10 and 11 were interesting and new for me. A bit tedious at times but using image feature points was interesting.

11. Another interesting assignment. RANSAC was really fun and the results were pretty neat too. The grad part of line fitting was also interesting.

12. Comparatively a straightforward and easy assignment. Interesting grad part question.s

Assignment	Interest	Usefulness	Length
1	1	2	2
2	3	2	3
3	4	5	4
4	3	4	4
5	5	5	5
6	5	5	4
7	3	4	3
8	5	5	4
9	5	5	4
10	5	5	5
11	5	5	5
12	4	5	4

Part D

Similar to the slides and frames of the previous several assignments, we continued exploring the relationship between matching points between multiple cameras. However, the the slides and frames of the previous assignments has the planar constraint that allowed us to use a homography, but, now, the matching points are no longer coplanar. There is still a valid relationship between the matches, though, using the fundamental matrix:

$$(x')^T F x = 0$$

where x is a point in the first image, x' is the matching point in the second image, and F is the fundamental matrix. This constraint essentially means that, when the line going from the first camera through the point x is projected into the other camera's view, the matching point, x', lies exactly on that line.

First, we picked twenty different matches. You can see these picked points in the nearby figure. We have color-coded the points for convenience.

Figure 3: The two images, now with twenty picked points. For each point in the first image, the matching point in the second image is colored the same.

Now that we have some known good matches, we can work on solving for the fundamental matrix F. The equation above gives us a good strategy to do so. Let's consider the general form of the equation above, and multiply that equation out:

$$\begin{bmatrix} x_1' & x_2' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} x_1' & x_2' & 1 \end{bmatrix} \begin{bmatrix} f_{11}x_1 + f_{12}x_2 + f_{13} \\ f_{21}x_1 + f_{22}x_2 + f_{23} \\ f_{31}x_1 + f_{32}x_2 + f_{33} \end{bmatrix} = 0$$

$$f_{11}x_1x_1' + f_{12}x_2x_1' + f_{13}x_1' + f_{21}x_1x_2' + f_{22}x_2x_2' + f_{23}x_2' + f_{31}x_1 + f_{32}x_2 + f_{33} = 0$$

At this point, we can use a similar strategy to finding the camera matrix and homography matrix: treat the matrix as a large vector, instead, and make use of the dot product:

$$\begin{bmatrix} x_1x_1' & x_2x_1' & x_1' & x_1x_2' & x_2x_2' & x_2' & x_1 & x_2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

Since this was solved in general, this should hold for any such pair of matching points. From here, we have the makings of the homogeneous solution to line fitting. We pick at least 9 points (since the fundamental matrix has 9 unknowns), create a matrix from the left-hand side of the equation above, using a new pair of matches for each row, and solve it as we've done with other similar problems. We used 12 of the matching points to calculate the matrix below:

$$F = \begin{bmatrix} 1.8581e-6 & 8.4743e-6 & -0.0139 \\ -1.1489e-5 & 8.4903e-7 & 0.0023 \\ 0.0117 & -0.0030 & 0.9998 \end{bmatrix}$$

We then used this matrix and the equation above to calculate errors. In this case, we know that $x'Fx$ should be 0, in an ideal world, so we use 0 as the target for the error calculation. We get a RMSE of 0.0123 for the training points and a RMSE of 0.0665 for the test points. A lower error for training data than testing data is a good indicator that we are doing things correctly. In addition, these values are pretty small, which is good, but not zero, which is a reflection of the fact that we were not perfect when we got the pixel coordinates of each match.

Though the numbers look reasonable, it is hard to gauge these results without a visualization. What we really want is to visualize the epipolar lines of the points w.r.t. the first camera in the image space of the second camera. Since epipolar lines can be though of as the line emitting from the camera and passing through a given point, these lines should converge at a point on the image plane that represents the location of the first camera, projected onto the second camera's image plane. To do so, we simply need to remember the original homogeneous line equation:

$$ax + by = d$$
$$ax + by - d = 0$$

Note that, conveniently, this is a dot product:

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ -d \end{bmatrix} = 0$$

In fact, the first vector can be thought of as x' and the second vector can be though of exactly as the result of Fx. Thus, the line equation is simply the result of Fx, which we then draw onto the image plane in the nearby figure.

Figure 4: The epipolar line of each point as seen by the first camera, projected onto the image space of the second image.

Judging by this image, the lines do converge somewhere off the right. Initially, it seemed like the lines were heading toward the right wall, just to the right of the windows, which did not match our intuitive understanding of the first camera's position. However, after realizing that the camera could be positioned anywhere along the *perpendicular* axis of the second camera's image plane at that convergent point, we realized that the position of the first camera based on our intuitive understanding, above and behind the front chair, *did* lie on that convergent point, just pulled away from the image plane, toward the viewer.