ICS 483, Fall 2025
Programming Assignment 2
Augmented Reality with Planar Homographies

Kanta Saito

Due Date: (Thur) Oct 16, 2025 23:55 (HST)

Q2.1 Correspondences (15 points)

Let $\mathbf{x}_1$ be a set of points in an image and $\mathbf{x}_2$ be the set of corresponding points in an image taken by another camera. Suppose there exists a homography $\mathbf{H}$ such that:

$$\mathbf{x}_1^i \equiv \mathbf{H}\mathbf{x}_2^i \quad (i \in \{1 \ldots N\})$$

where $\mathbf{x}_1^i = [x_1^i \quad y_1^i \quad 1]^T$ are in homogeneous coordinates, $\mathbf{x}_1^i \in \mathbf{x}_1$ and $\mathbf{H}$ is a 3×3 matrix. For each point pair, this relation can be rewritten as

$$\mathbf{A}_i\mathbf{h} = 0$$

where $\mathbf{h}$ is a column vector reshaped from $\mathbf{H}$, and $\mathbf{A}_i$ is a matrix with elements derived from the points $\mathbf{x}_1^i$ and $\mathbf{x}_2^i$. This can help calculate $\mathbf{H}$ from the given point correspondences.

1. How many degrees of freedom does $\mathbf{h}$ have? (3 points)

2. How many point pairs are required to solve $\mathbf{h}$? (2 points)

3. Derive $\mathbf{A}_i$. (5 points)

4. When solving $\mathbf{Ah} = 0$, in essence you're trying to find the $\mathbf{h}$ that exists in the null space of $\mathbf{A}$. What that means is that there would be some non-trivial solution for $\mathbf{h}$ such that that product $\mathbf{Ah}$ turns out to be 0.

   What will be a trivial solution for $\mathbf{h}$? Is the matrix $\mathbf{A}$ full rank? Why/Why not? What impact will it have on the singular values? What impact will it have on the singular vectors? (5 points)

Q3.1 FAST Detector (5 points)

How is the FAST detector different from the Harris corner detector that you've seen in the lectures? (You will probably need to look up the FAST detector online.) Can you comment on its computational performance vis-à-vis the Harris corner detector?

Q3.2 BRIEF Descriptor (5 points)

How is the BRIEF descriptor different from the filterbanks you've seen in the lectures? Could you use any one of those filter banks as a descriptor?

Q3.3 Matching Methods (5 points)

The BRIEF descriptor belongs to a category called binary descriptors. In such descriptors the image region corresponding to the detected feature point is represented as a binary string of 1s and 0s. A commonly used metric used for such descriptors is called the *Hamming distance*. Please search online to learn about Hamming distance and *Nearest Neighbor*, and describe how they can be used to match interest points with BRIEF descriptors. What benefits does the Hamming distance distance have over a more conventional Euclidean distance measure in our setting?

Write-Up

Q2.1 Correspondences

1. Degrees of Freedom of $\mathbf{h}$:

The homography matrix $\mathbf{H}$ is a 3×3 matrix with 9 elements. However, since the homography is defined up to scale (multiplication by a scalar does not change the transformation), we have one constraint. Therefore, the degrees of freedom of $\mathbf{h}$ is $9 - 1 = 8$.

2. Number of Point Pairs Required:

Since each point correspondence provides 2 independent equations and 8 degrees of freedom, we require $\frac{8}{2} = 4$ point pairs.

3. Derivation of $\mathbf{A}_i$:

The homography relationship is given by $\mathbf{x}' \equiv \mathbf{H}\mathbf{x}$. Since $\mathbf{x}'$ and $\mathbf{H}\mathbf{x}$ are parallel vectors in homogeneous coordinates, their cross product must be the zero vector:

$$\mathbf{x}' \times \mathbf{H}\mathbf{x} = \mathbf{0}$$

Let $\mathbf{x} = [x, y, 1]^T$, $\mathbf{x}' = [x', y', 1]^T$, and let the rows of $\mathbf{H}$ be the vectors $\mathbf{h}_1^T, \mathbf{h}_2^T, \mathbf{h}_3^T$. Then $\mathbf{H}\mathbf{x}$ can be written as:

$$\mathbf{H}\mathbf{x} = \begin{bmatrix} \mathbf{h}_1^T\mathbf{x} \\ \mathbf{h}_2^T\mathbf{x} \\ \mathbf{h}_3^T\mathbf{x} \end{bmatrix}$$

The cross product becomes:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{h}_1^T\mathbf{x} \\ \mathbf{h}_2^T\mathbf{x} \\ \mathbf{h}_3^T\mathbf{x} \end{bmatrix} = \begin{bmatrix} y'(\mathbf{h}_3^T\mathbf{x}) - 1(\mathbf{h}_2^T\mathbf{x}) \\ 1(\mathbf{h}_1^T\mathbf{x}) - x'(\mathbf{h}_3^T\mathbf{x}) \\ x'(\mathbf{h}_2^T\mathbf{x}) - y'(\mathbf{h}_1^T\mathbf{x}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

This gives us three equations, although only two are linearly independent. Let's write out the first two equations.

For the first row:

$$y'(\mathbf{h}_3^T\mathbf{x}) - \mathbf{h}_2^T\mathbf{x} = 0$$

$$y'(h_{31}x + h_{32}y + h_{33}) - (h_{21}x + h_{22}y + h_{23}) = 0$$

$$(0)h_{11} + (0)h_{12} + (0)h_{13} - (x)h_{21} - (y)h_{22} - (1)h_{23} + (y'x)h_{31} + (y'y)h_{32} + (y')h_{33} = 0$$

For the second row:

$$\mathbf{h}_1^T\mathbf{x} - x'(\mathbf{h}_3^T\mathbf{x}) = 0$$

$$(h_{11}x + h_{12}y + h_{13}) - x'(h_{31}x + h_{32}y + h_{33}) = 0$$

$$(x)h_{11} + (y)h_{12} + (1)h_{13} + (0)h_{21} + (0)h_{22} + (0)h_{23} - (x'x)h_{31} - (x'y)h_{32} - (x')h_{33} = 0$$

These two equations for a single point correspondence i can be written in the matrix form $\mathbf{A}_i\mathbf{h} = \mathbf{0}$, where $\mathbf{h}$ is the 9x1 vector of the elements of $\mathbf{H}$. The 2×9 matrix $\mathbf{A}_i$ is:

$$\mathbf{A}_i = \begin{bmatrix} 0 & 0 & 0 & -x_i & -y_i & -1 & y_i'x_i & y_i'y_i & y_i' \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i'x_i & -x_i'y_i & -x_i' \end{bmatrix}$$

where $\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^T$.

4. Null Space Analysis:

The trivial solution for $\mathbf{h}$ is $\mathbf{h} = \mathbf{0}$ (the zero vector). The matrix $\mathbf{A}$ is not full rank because we have 8 degrees of freedom but 9 unknowns. With N point pairs, $\mathbf{A}$ is $2N \times 9$. For $N \geq 4$, $\mathbf{A}$ has rank at most 8, meaning one singular value will be zero, corresponding to the null space. The singular vector corresponding to the smallest singular value gives the solution for $\mathbf{h}$.

Q3.1 FAST Detector

The FAST detector is different from the Harris corner detector in several fundamental ways. FAST uses a circle of 16 pixels (Bresenham circle of radius 3) around a candidate point p and classifies it as a corner if a set of N contiguous pixels are all brighter than $I_p + t$ or darker than $I_p - t$, where I_p is the intensity at p and t is a threshold. Harris, computes the structure tensor from image gradients using I_x^2, I_y^2, and $I_x I_y$, then evaluates a corner response function.

Computationally, FAST is more efficient than Harris. FAST requires only simple intensity comparisons without computing gradients, convolutions, or matrix operations. Harris requires computing derivatives, Gaussian smoothing, and eigenvalue computations. This makes FAST ideal for real-time video, while Harris provides more stable and quality corner detection at the cost of computational complexity.

Q3.2 BRIEF Descriptor

The BRIEF descriptor is different from traditional filterbanks. Filterbanks (Gaussian, Laplacian Pyramid, and Gabor Filter) produce a response by convolving the image with a set of filters at different scales and orientations, resulting in quality point descriptors.

BRIEF, creates a binary string descriptor by performing simple intensity difference tests on pairs of pixels within a patch around the keypoint. For each pair of pixels (p, q), BRIEF computes: $\tau(p, q) = \begin{cases} 1 & \text{if } I(p) < I(q) \\ 0 & \text{otherwise} \end{cases}$. This produces a binary vector that is small and fast to compute and compare.

Even though filter banks responses could be used as descriptors, they are less efficient than BRIEF. Filterbanks require convolution operations and produce floating point values requiring more storage and slower comparison. BRIEF's binary nature enables very fast Hamming distance computation using XOR operations. However, filterbanks provide quality information and may be more better to certain transformations.

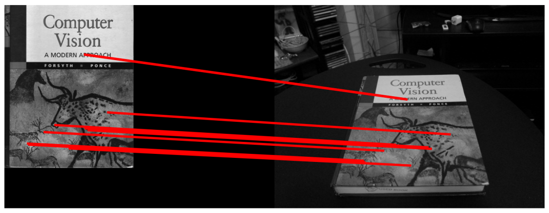
Q3.3 Matching Methods

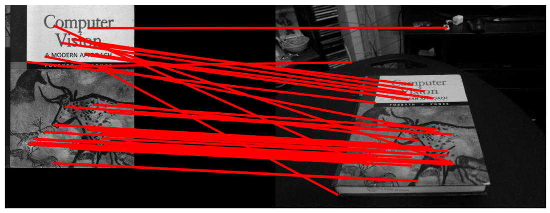
The Hamming distance between two binary strings is the number of positions at which the corresponding bits differ. For two binary descriptors $\mathbf{b}_1$ and $\mathbf{b}_2$, the Hamming distance is: $d_H(\mathbf{b}_1, \mathbf{b}_2) = \sum_{i=1}^{n} (b_{1,i} \oplus b_{2,i})$, where $\oplus$ is the XOR operation.

In Nearest Neighbor matching with BRIEF descriptors, for a query descriptor, we compute the Hamming distance to all descriptors in a set and select the descriptor with the minimum Hamming distance as the match. A enhancement is the nearest neighbor distance ratio test, where we compute the ratio of the distance to the closest neighbor versus the second-closest neighbor and we will take the ones that are below a certain threshold.

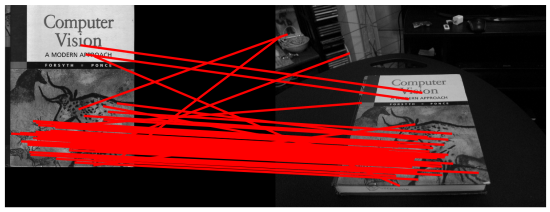
The Hamming distance has significant advantages over Euclidean distance for binary descriptors. First, it can be computed extremely efficiently using bitwise XOR operations followed by a count, which modern CPUs implement in single instructions. Second, binary descriptors require much less memory compared to floating-point descriptors. Finally, Hamming distance is the natural metric for binary data, whereas Euclidean distance is designed for continuous vector spaces.

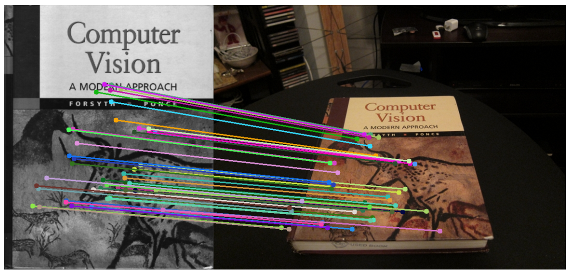
Q3.4 Feature Matching

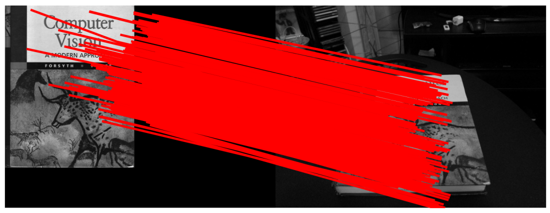

(a) BRIEF Descriptor (Sigma=0.15, ratio=0.65)


(b) BRIEF Descriptor (Sigma=0.15, ratio=0.75)

(a) BRIEF Descriptor (Sigma=0.00, ratio=0.00)


(b) BRIEF Descriptor (Sigma=0.60, ratio=0.70)


(a) Sift Matches Visualization (Cleaner)


(b) SIFT Descriptor (Sigma=0.15, ratio=0.65)

For the first images on the top, I used the BRIEF Descriptor that was provided but it did not work as I expected it to. It seems like it's matching well but that is not the case when I use it in the the later functions.

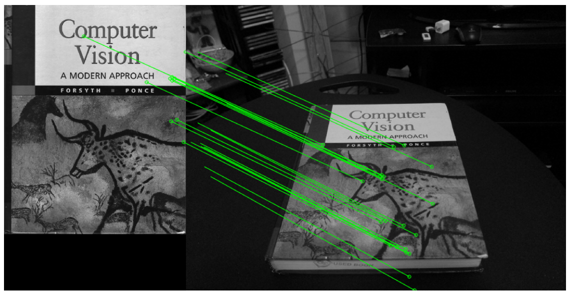
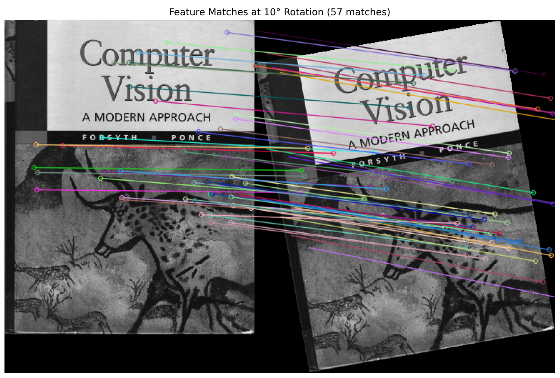


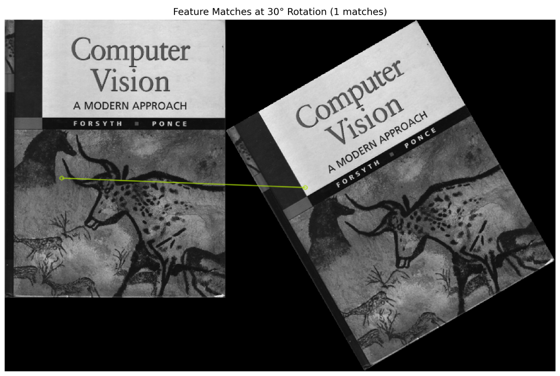
Figure 4: BRIEF Descriptor (patch=15, nbits=512

I resorted in using the SIFT descriptor instead due to this error. I tried changing the PATCHWIDTH and the nbits of the BRIEF helper function but it still gave me mismatches in my descriptor.

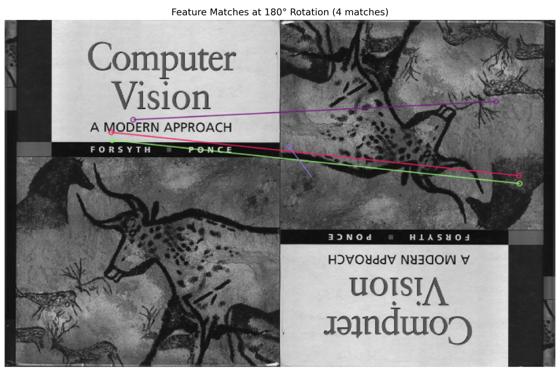
Q3.5 Feature Matching



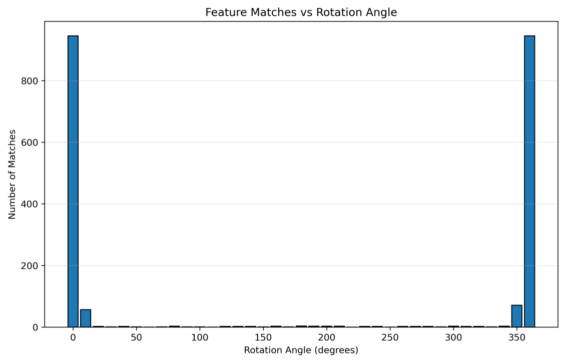
(a) Rotation 10 degrees



(b) Rotation 30 degrees



(a) Rotation 180 degrees

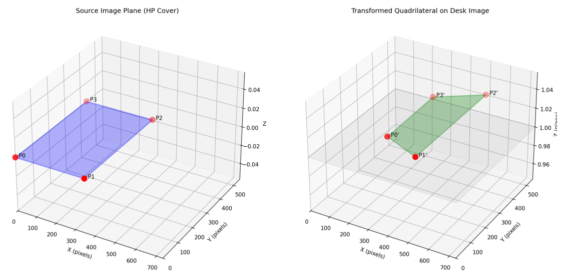


(b) Histogram of Rotation

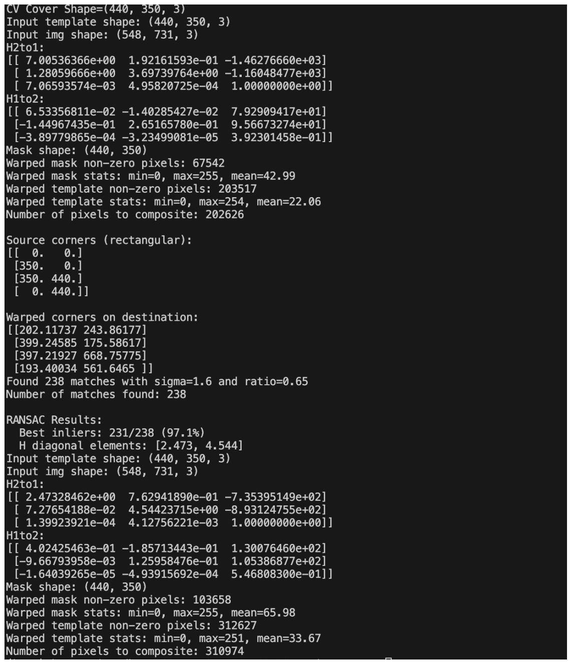
I used my own visualization to better see the matches. Comparing to SIFT, these images don't have enough that matches. The 30 degrees have only 1 match which is not enough for RANSAC.

BRIEF is not rotation-invariant since it uses fixed pixel pair comparisons that don't account for rotation. It's also not scale-invariant so it's not that good for both but is just faster than SIFT and ORB.

Q3.9 Putting It Together

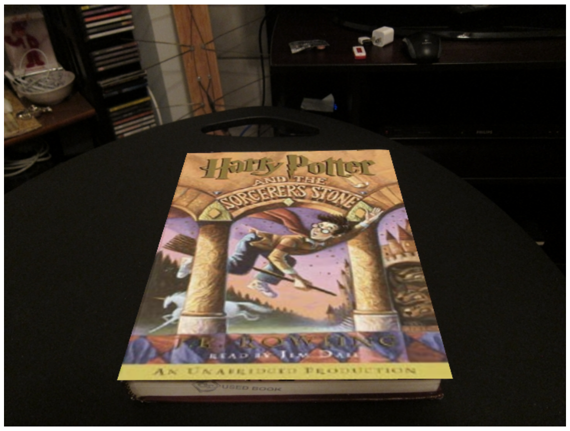


(a) Homography 3D Visualization

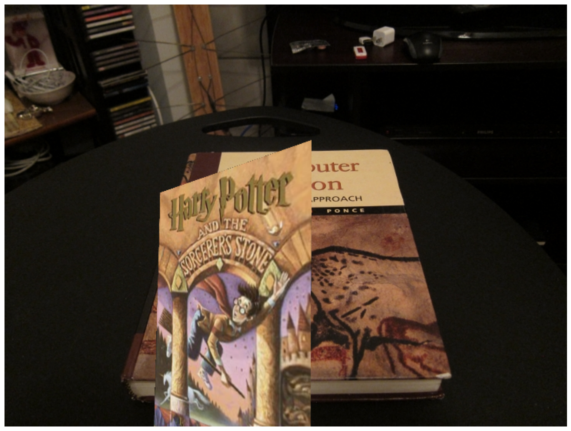


```
CV Cover Shape=(440, 350, 3)
Input template shape: (440, 350, 3)
Input img shape: (548, 731, 3)
H2to1:
[[ 7.00536366e+00  1.92161593e-01 -1.46276660e+03]
 [ 1.28059666e+00  3.69739764e+00 -1.16048477e+03]
 [ 7.06593574e-03  4.95820725e-04  1.00000000e+00]]
H1to2:
[[ 6.53356811e-02 -1.40285427e-02  7.92909417e+01]
 [-1.44967435e-01  2.65165780e-01  9.56673274e+01]
 [-3.89779865e-04 -3.23499081e-05  3.92301458e-01]]
Mask shape: (440, 350)
Warped mask non-zero pixels: 67542
Warped mask stats: min=0, max=255, mean=42.99
Warped template non-zero pixels: 203517
Warped template stats: min=0, max=254, mean=22.06
Number of pixels to composite: 202626

Source corners (rectangular):
[[  0.   0.]
 [350.   0.]
 [350. 440.]
 [  0. 440.]]

Warped corners on destination:
[[202.11737 243.86177]
 [399.24585 175.58617]
 [397.21927 668.75775]
 [193.40034 561.6465 ]]
Found 238 matches with sigma=1.6 and ratio=0.65
Number of matches found: 238

RANSAC Results:
  Best inliers: 231/238 (97.1%)
  H diagonal elements: [2.473, 4.544]
Input template shape: (440, 350, 3)
Input img shape: (548, 731, 3)
H2to1:
[[ 2.47328462e+00  7.62941890e-01 -7.35395149e+02]
 [ 7.27654188e-02  4.54423715e+00 -8.93124755e+02]
 [ 1.39923921e-04  4.12756221e-03  1.00000000e+00]]
H1to2:
[[ 4.02425463e-01 -1.85713443e-01  1.30076460e+02]
 [-9.66793958e-03  1.25958476e-01  1.05386877e+02]
 [-1.64039265e-05 -4.93915692e-04  5.46808300e-01]]
Mask shape: (440, 350)
Warped mask non-zero pixels: 103658
Warped mask stats: min=0, max=255, mean=65.98
Warped template non-zero pixels: 312627
Warped template stats: min=0, max=251, mean=33.67
Number of pixels to composite: 310974
```

(b) Terminal Results



(a) SIFT Result



(b) BRIEF Result

The problem with this is that the cv desk shows the Computer Vision book laying flat on the desk but the hp cover shows the Harry Potter cover, which is tall/portrait oriented. This is a problem because the cover has a different aspect ratio than the cv cover, the hp cover is portrait(tall) and the cv cover book is landscape(wide). A solution that I have is to resize

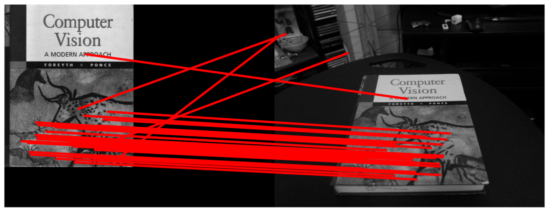


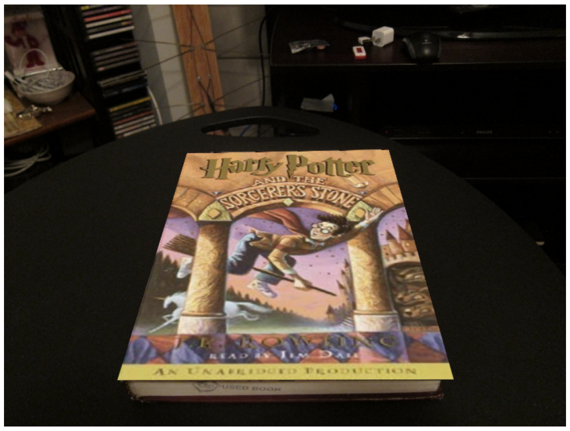
Figure 10: BRIEF Descriptor (Sigma=0.10, ratio=0.65)

the hp cover to exactly match the dimensions of the cv cover before warping. This is done to ensure that the aspect ratio is the same.

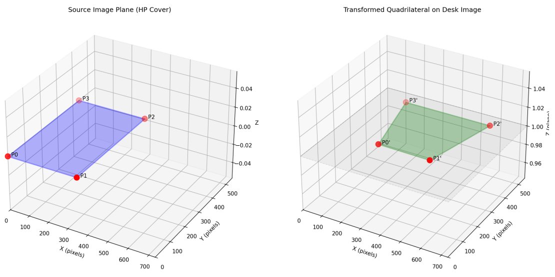
When I tested my composite homography with the BRIEF descriptor, it gave me an image that is rotated and could not fit onto the book on the desk. When I visualized it, I was right, and I spent a few days debugging and having a whole conversation with my LLM friend. I came to a conclusion that it was not my RANSAC nor my compositeH function, but it was the descriptor itself.

After switching to the SIFT descriptor, I got immediate good results. It seems like since BRIEF is not applicable for scaling and rotation, it was having a hard time matching the two images.

Q3.9(b) Putting It Together BRIEF WORKS



(a) Fixed BRIEF Result



(b) New Homography 3D Look

Taking back what I said about BRIEF descriptor not working on my images. It was actually my compositeH that wasn't working. The reason why it was warping wrong is because I called cv2.warpPerspective on both the mask and the template. I fixed it to only calling it on the template. My mask was then made after the warping and I changed my values to np.float32 for calculations. I then normalized the background and the warped template then combined it to create my composite image. I also thought that using the ratio of 0.65 and sigma of 0.10 gave the best matches out of all my test cases.

Q.4 Extra Credit

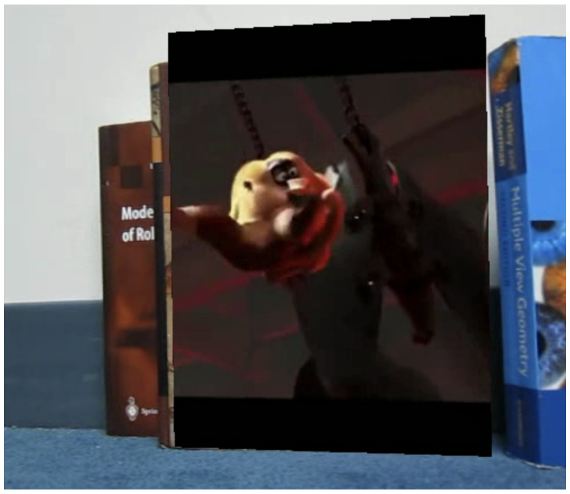
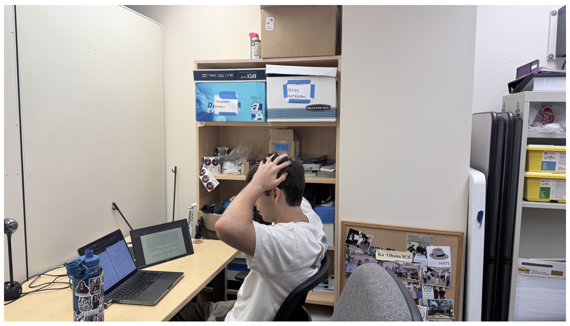
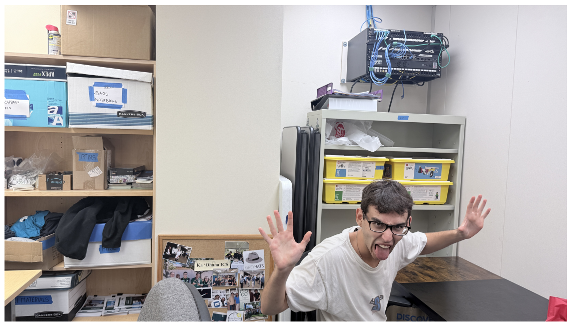


Figure 11: My AR

Q.5 Extra Credit


(a) Lukas Left Image


(b) Lukas Right Image

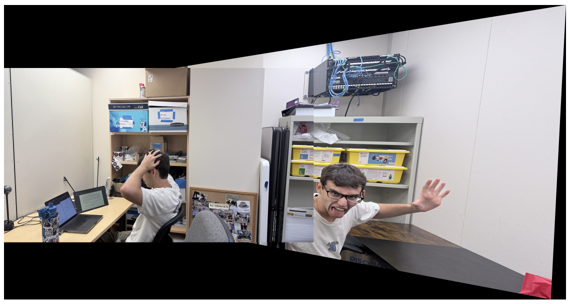


Figure 13: Lukas Panorama

I used Lukas's pictures to make my panorama, the hardest part stitching the images together. I used np.where to blend two images together. I also needed to warp one side with H and translation because I'm trying to switch one image to another.

References

Talked to George, ZWei, Lukas, Wilson, and my LLM friends.
I used GeekforGeeks for the SIFT Functions

LLM

"LLM: How do I visualized the Homography in 3 dimensional"
"LLM: How do I draw matched points with an array of two images coordinates."
"LLM: What is Similarity transform in code?"
"LLM: how do I crop a frame based on aspect ratio python"
"LLM: How do I calculate Homography on a video?"