2025 Spring CV HW1

Yelin Zhang

April 17, 2025

Contents

List of Figures

1 Linear algebra

1.1 1-1

1.1.1 a

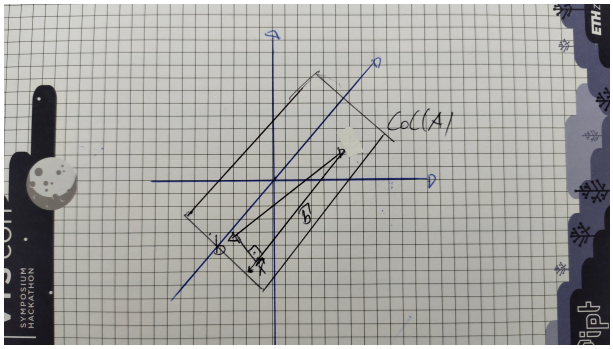


Figure 1: Drawing of the plane

1.1.2 b

Given residual vector:

$$r = b - A\hat{x}$$

Due to the residual vector being orthogonal to the column space of A, the dot product can be used to get the following equation:

$$A^\top r = 0$$

By substituting the residual vector into the equation we get:

$$A^\top (b - A\hat{x}) = 0$$

Which can be simplified to the given least squares solution with the following steps: Multiplying out the equation:

$$A^\top b - A^\top A\hat{x} = 0$$

Start to isolate $\hat{x}$:

$$A^\top b = A^\top A\hat{x}$$

Finally, by isolating $\hat{x}$ with the inverse of $A^\top A$ we get the given least squares solution.
Given least squares solution:

$$\hat{x} = (A^\top A)^{-1} A^\top b$$

1.1.3 c

$$AA^\dagger b \in \mathbb{R}^m$$

When $AA^\dagger$ is applied to a vector b, it projects b onto the column space of A. This is because $AA^\dagger$ acts as the projection matrix and multiplying it by b results in the vector in the column space of A that is closest to b. Therefore formula is the orthogonal projection of the vector b onto the column space of A.

1.2 1-2

1.2.1 a

The last Eigenvectors corresponds with the smallest eigenvalue of the covariance matrix. Therefore, it is the direction of the least variance in the data. Which means it minimizes $||Ax||$, due to the fact that the data is projected onto the direction of the least variance.

1.2.2 b

It is still useful to minimize $||Ax||$, because this identifies the direction in which A the solution is closest to 0 and the corresponds to the right singular vector associated with the smallest singular value of A

1.2.3 c

$$A = \begin{bmatrix} -0.99994899 & -0.01010048 \\ -0.01010048 & 0.99994899 \end{bmatrix} \begin{bmatrix} 1.0000505 & 0 \\ 0 & 0.09999495 \end{bmatrix} \begin{bmatrix} -0.99999949 & -0.00101 \\ -0.00101 & 0.99999949 \end{bmatrix}$$

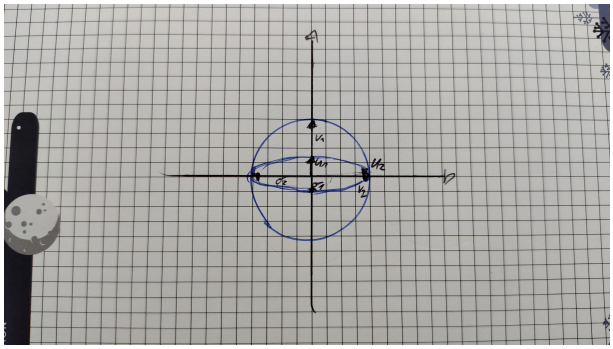


Figure 2: Drawing of the circle

u_1 is the shortest output vector which is mapped by the input vector v_1.

1.2.4 d

The direction $x = v2$ is the direction of the least variance in the data. Therefore, it is the direction in which the data is projected onto the least variance.

1.3 1-3

1.3.1 a

$$f(k) = k * pixel\ height + k * pixel\ width + k$$

1.3.2 b

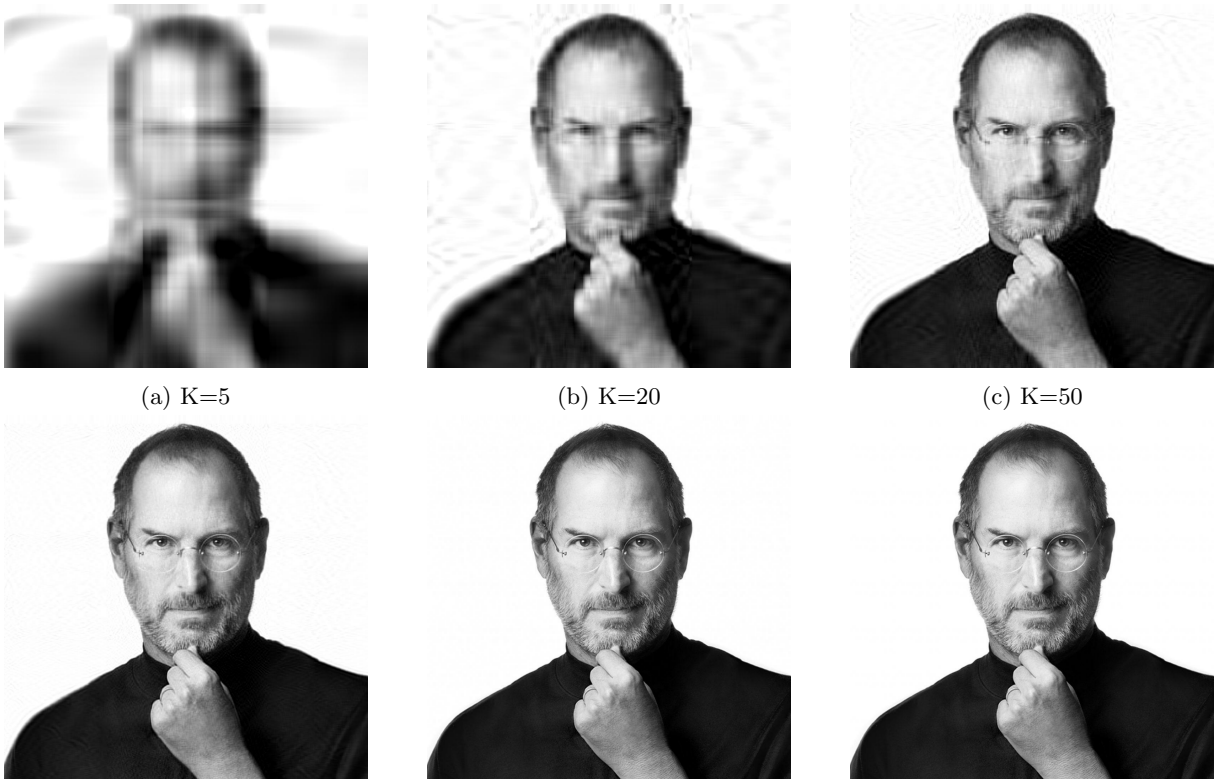
By iterating over values of k and checking if the compression ratio larger than 1, we can find $k^* = 511$ which is a compressed representation.

1.3.3 c

The original image has a file size of 67.9 KB.

The lower the k the higher the compression ratio. Therefore also lower visual quality.

- At $k = 5$ the image is barely recognizable only vague shapes remain. The size has been roughly halved to 36.6KB 3a.
- At $k = 20$ the image is recognizable but very strong compression artifacts are visible. Size at 44.7 KB 3b.
- At $k = 50$ the solid black color has become mosaic like, but otherwise it is very recognizable. With a size of 55 KB 3c.
- At $k = 100$ no large compression artifacts are visible. But it is barely smaller than the original at 66.4 KB 3d.
- At $k = k^*$ the image looks identical to the original image. But happens to be larger than the original 68.6 KB due to floating point inaccuracies while converting 3e.



(a) K=5 (b) K=20 (c) K=50

(d) K=100 (e) K=511 (f) Original

Figure 3: Compression with different k values

2 HOG

2.1 a

(a) HOG Visualization (b) Facial detections

Figure 4: Output for existing target and template

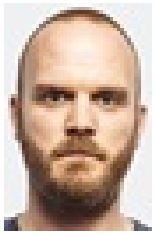
The performance is not very good for the provided target and template. It only manages to detect the central two faces 4b. But the bounding boxes of the detected faces are decent, not quite centered on the face, but it encapsulates the whole face. There are no false positives which is good.

There is room for improvement in the code due to the fact that not all faces are detected. This could also be achieved by some hyperparameter tuning.

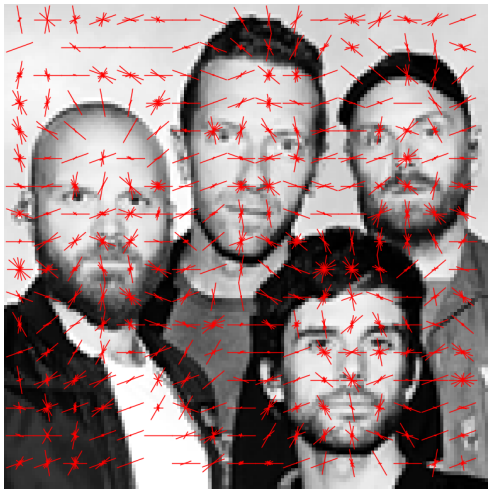
2.2 b

Selecting the target and template image was difficult, due to having to respect a certain size ratio between both. If the template is too small in comparison to the target, no detection will be found. After rescaling both to a similar size the detections worked well.

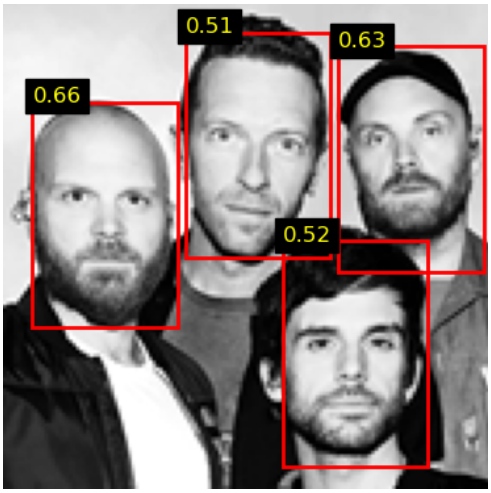
(a) Target



(b) Template



(c) HOG Visualization



(d) Facial detections

Figure 5: Own target and template for HOG and their output

In comparison to the provided template and target the detections are much better with my own selected one. Due to me having problems with the scale between target and template I assume the provided ones are not quite in the correct scale. Mainly because the template is so much smaller than the target.

2.3 c

Sobel Kernels for X & Y directions:

$$X = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad Y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Sobel emphasizes edges by combining smoothing and differentiation, making it less sensitive to noise.

Prewitt Kernels for X & Y directions:

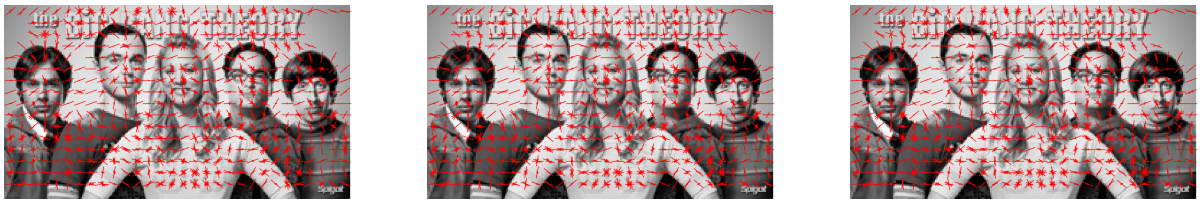
$$X = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \quad Y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Prewitt kernels are simpler, using uniform weights. It detects edges by measuring intensity changes but is slightly more sensitive to noise.

Scharr Kernels for X & Y directions:

$$X = \begin{pmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{pmatrix} \quad Y = \begin{pmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ 3 & 10 & 3 \end{pmatrix}$$

Scharr provides better rotational symmetry and more accurate gradient estimation, especially for diagonal edges.



(a) Prewitt (b) Scharr (c) Sobel

Figure 6: HOG visualizations of different filters

(a) Prewitt (b) Scharr (c) Sobel

Figure 7: Facial detections with different filters

There are some differences which can be seen in the visualizations 6. But the result face detections remain the same 7. With only slight changes to the score. Only very slight changes to the bounding boxes can be seen, where Prewitt is the most centered one. Followed by Scharr and Sobel. Therefore Prewitt performs the best on this set of target and template but only minimally.

3 Harris

3.1 3-1

3.2 a

The horizontal gradient $|I_x|$ mostly highlights the vertical edges in the image. Meanwhile the vertical gradient $|I_y|$ mostly highlights the horizontal edges in the image. This is not clear cut due to the nature of the image. The right side of the cube with both vertical and horizontal edges is completely highlighted with the horizontal gradient. Same applies to the vertical gradient, the bottom side of the cube is highlighted with both vertical and horizontal edges. The gradient magnitude is the sum of the horizontal and vertical gradients. Therefore it highlights all edges in the image. But does not add any additional information.

3.3 b

Flat region

- Pixel coordinates: X: 144 Y: 268
- Eigenvalues: 0, 0
- Interpretation: Flat region

Edge like region

- Pixel coordinates: X: 362 Y: 308
- Eigenvalues: 1113.2, 0
- Interpretation: Edge-like region

Corner

- Pixel coordinates: X: 166 Y: 195
- Eigenvalues: 124.6, 31.8
- Interpretation: Flat region

3.4 c

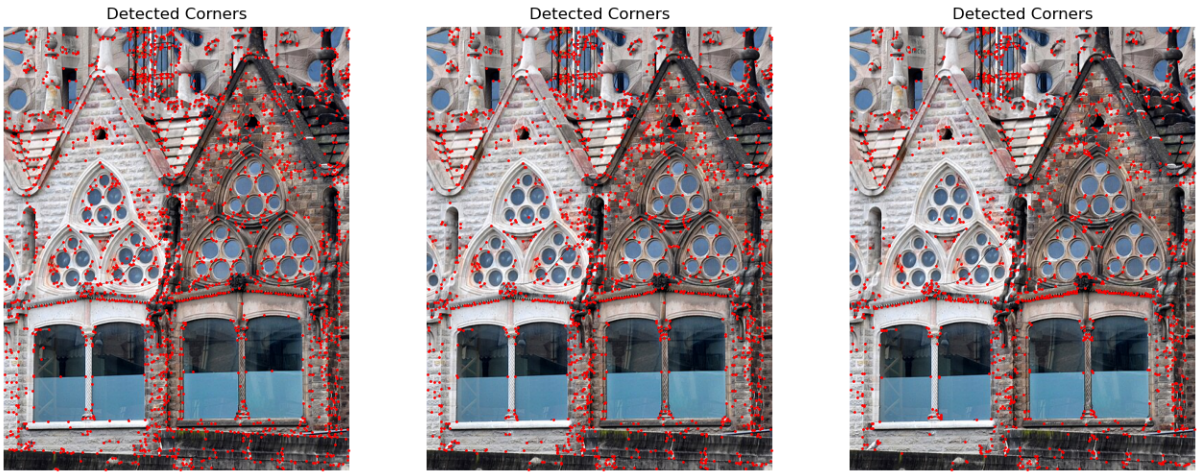
- Small round ellipse should be a flat region due to both eigenvalues being small.
- Elongated ellipse are edges due to one eigenvalue being larger than the other.
- Circular large ellipse should be a corner due to large eigenvalues in both directions.

3.5 3-2

3.6 a

2030 corners are detected with $sigma = 1, k = 0.05, threshold\_ratio = 0.01$

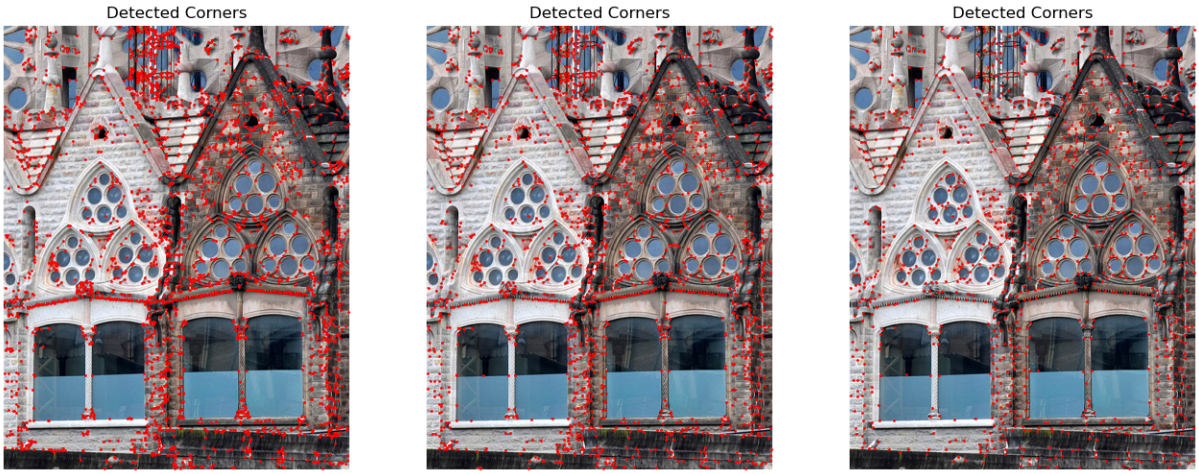
3.7 b



(a) k 0.01 (b) k 0.1 (c) k 0.2

Figure 8: Harris corners with different k values

- k=0.01: 2254 corners, high density and many corner detections do not correspond to an actual corner 8a.
- k=0.1: 1861 corners, lower density compared to lower k values but still many corners are accurately detected 8b.
- k=0.2: 1463 corners, lowest density of corners, many corners are not detected in the areas where color is similar 8c.



(a) sigma 0.5 (b) sigma 1 (c) sigma 2

Figure 9: Harris corners with different sigma values

3.8 c

- sigma=0.5: 2899 corners, the density of the detections is very high, many of them are overlapping 9a.
- sigma=1: 2030 corners, this is the default setting of the given functions. Corners are well detected in darker areas, but corners in the white areas are not well detected 9b.
- sigma=2: 1311 corners, very low density of detections, but therefore they are seemingly more accurate 9c.

3.9 d

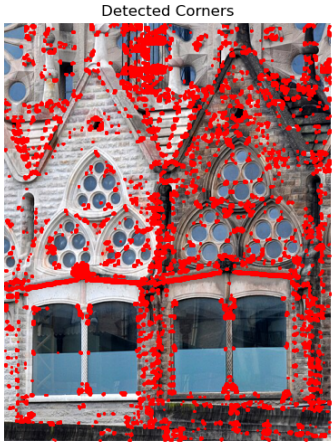


Figure 10: Harris corners without non-maximum suppression

The 25056 corners are detected with $sigma = 1, k = 0.05, threshold\_ratio = 0.01$ and no non-maximum suppression. This is 10 times more than with non-maximum suppression. The corners are not very well defined, due to many of them overlapping as well as whole areas and lines being detected as corner 10. NMS is needed to remove the overlapping corners and keep only the most prominent ones.

4 SIFT

4.1 4-1

4.2 a

According to the SIFT paper +3 is needed so that the extrema detection covers the whole octave.

4.3 b

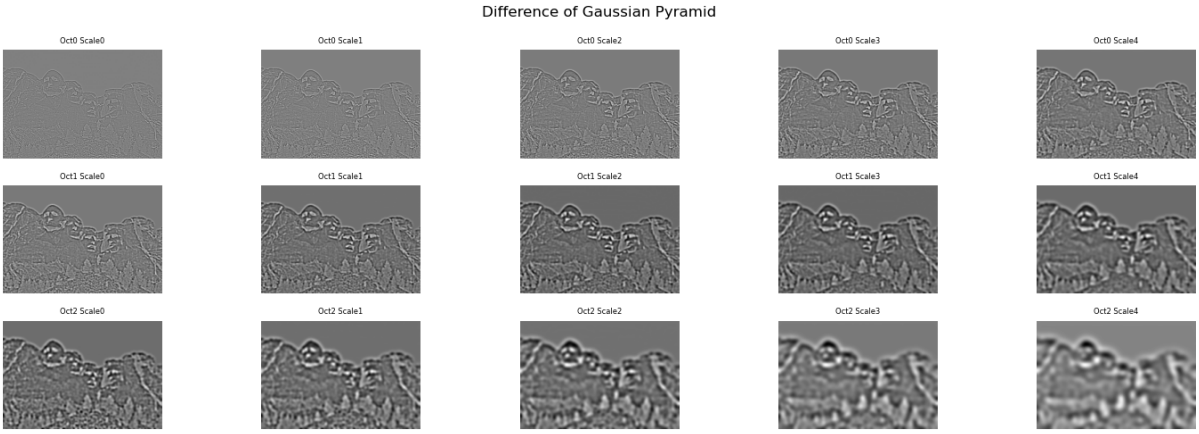
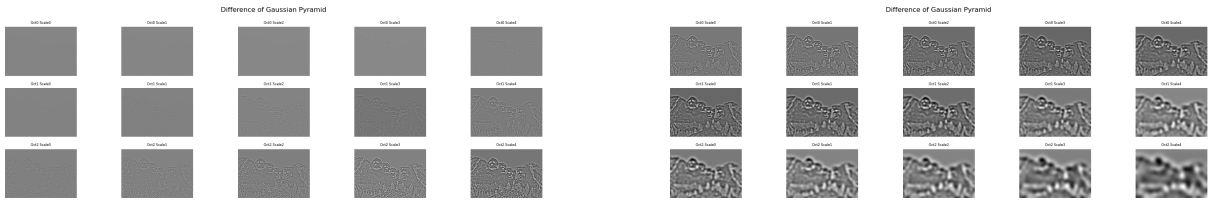


Figure 11: DoG pyramid with sigma 1.6

The smaller the octave the more detail is preserved 11. But also all edges and textures are less pronounced than with higher octaves. Only moving up the octaves makes the edges and textures more contrasted. For example octave 0 and scale 0 the edges and textures are very faint, but well preserved. In contrast to octave 2 scale 0 where edges and textures are very pronounced, but only slightly more blurry. Moving up the scale makes the features more blurry, which created more general blobs. When comparing octave 2 scale 0 and octave 2 scale 4 the blurring is very noticeable.

4.4 c



(a) DoG pyramid with sigma 0.5

(b) DoG pyramid with sigma 3.2

Figure 12: DoG pyramid with different sigma values

Sigma controls the size of the detected blobs, by controlling the smoothing. With a large sigma like 3.2 the processed image is a lot more smoothed 12b. Therefore the detection would only pick up on large keypoints. On the other hand with a small sigma like 0.5 the image a lot more features are preserved 12a. Which is very noisy and the detection would pick up on a lot of noise with smaller keypoints as well.

4.5 4-2

4.6 a

Each pixel in the image is compared to its neighbors in the same scale in a 3D patch. The 3D patch is a 3x3x3 cube, where the center pixel is the one being evaluated. The scale above and below the center pixel

are also included in the patch, forming the 3x3x3. This is done to ensure that the detected keypoints are stable across different scales and orientations.

4.7 b

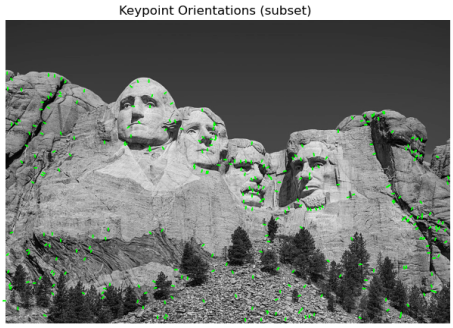
650 keypoints are detected with a contrast threshold of 0.03. By increasing to the threshold less keypoints are detected. By decreasing the threshold more keypoints are detected. For example at 0.01 1479 keypoints are detected and at 0.06 40 keypoints are detected.

4.8 c

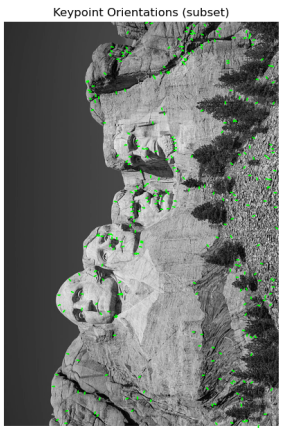
What types of regions (e.g., edges, flat areas, corners, blobs) tend to dominate the keypoint candidates? Use visual evidence from the DoG pyramid to support your claim. Regions with more contrast are more likely to be detected as keypoints 11. Therefore edges and corners are more likely due to the contrast difference they create. By looking at the image of keypoint orientations 13a and comparing that to DoG 11.

4.9 4-3

4.10 a



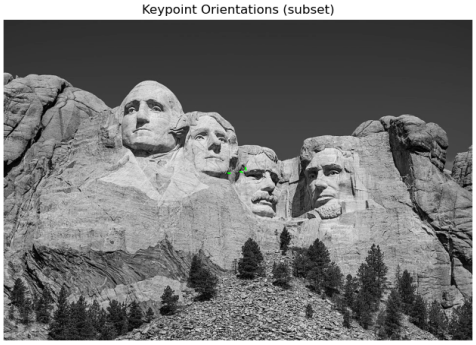
(a) Original keypoints



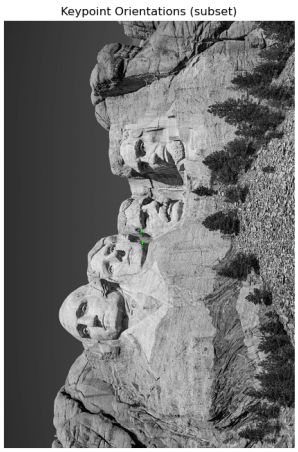
(b) Rotated keypoints

Figure 13: SIFT keypoints in original and rotated image

Selected the same few keypoints around the center of the image 14b.



(a) Original central keypoints

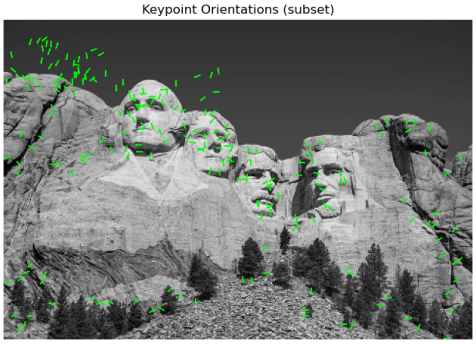


(b) Rotated central keypoints

Figure 14: SIFT central keypoints in original and rotated image

Dominant orientation for the keypoints in the normal image are 100, 200, 40. For the rotated image the orientation is expected to be 90 degrees off. Due to the applied rotation of 90 degrees. The calculated orientations are 10, 110, 310. Which matches the expected result. This means SIFT is invariant to rotation because the keypoints are detected in the same location regardless of orientation.

4.11 b



(a) All keypoints in scaled image

(b) Central keypoints in scaled image

Figure 15: SIFT keypoints in scaled image

The image was scaled down by 0.5. Some similar keypoints are detected but many of them are not 15a. There are keypoints missing and new keypoints in the scaled image. For the similar keypoints their orientation is the same as in the original image. The magnitude/scale of the orientation is also similar, there are slight differences but that might be due to not finding the exact same keypoints. The similar keypoints dominant orientation in original is 2, 1 for the scaled image it is 2, 3. The detected keypoints are not invariant to scale because the keypoints are not detected in the same location. But when the same keypoints are detected the orientation and scale is preserved.

4.12 c

The histogram method is more robust because it takes into account the distribution of gradient directions in the patch. This means that even if there is noise or texture in the image, the histogram will still provide a good estimate of the dominant orientation. In contrast, simply selecting the single strongest gradient direction may be affected by noise or texture, leading to an inaccurate estimate of the orientation.

4.13 4-4

4.14 a

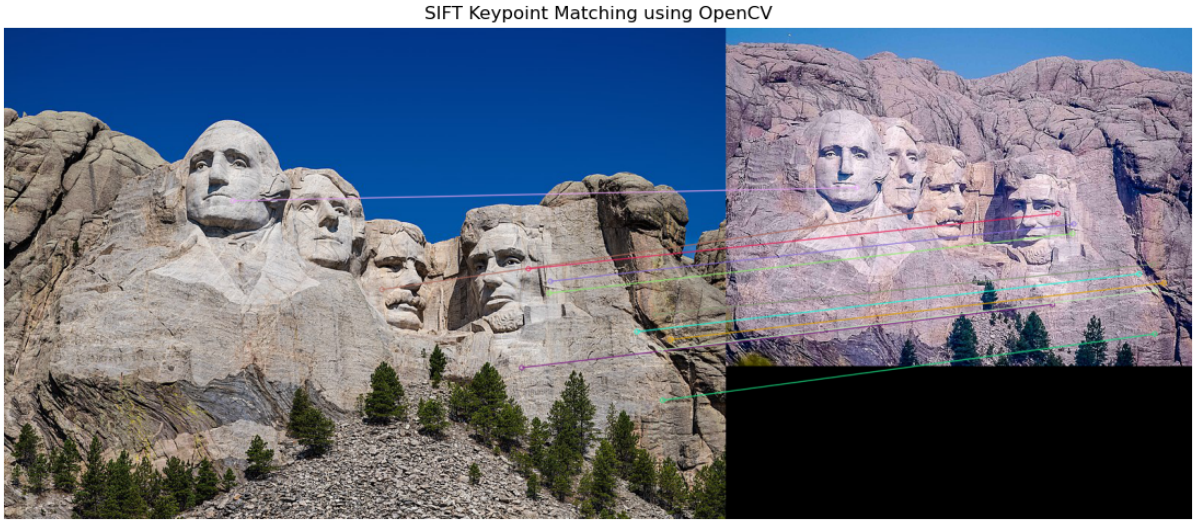


Figure 16: SIFT Keypoints matching between with Lowe's ratio test threshold of 0.4

Using the existing images of Mount Rushmore. The amount of keypoints in the left image are 4295 and in the right image 1640. But after the Lowe's ratio test with a threshold of 0.4 only 11 keypoints are left. Analyzing the keypoints manually shows that there is one false positive and the rest are correctly matched 16.

4.15 b

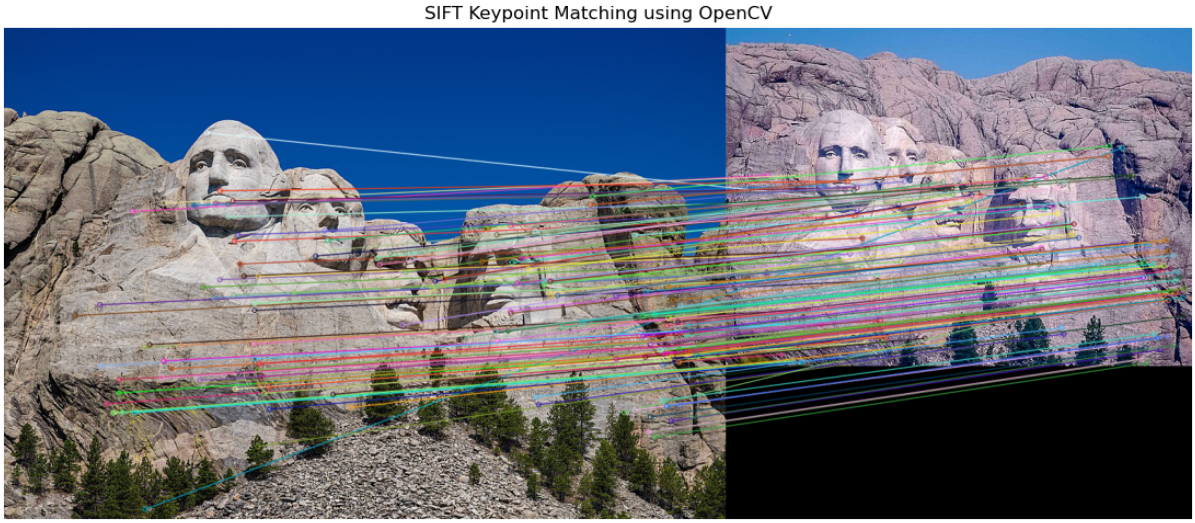


Figure 17: SIFT Keypoints matching between with Lowe's ratio test threshold of 0.75

Increasing the threshold to 0.75 resulted in more matches totaling 155 keypoints. Just by looking at a few of the matches reveals a way higher amount of false positives 17.

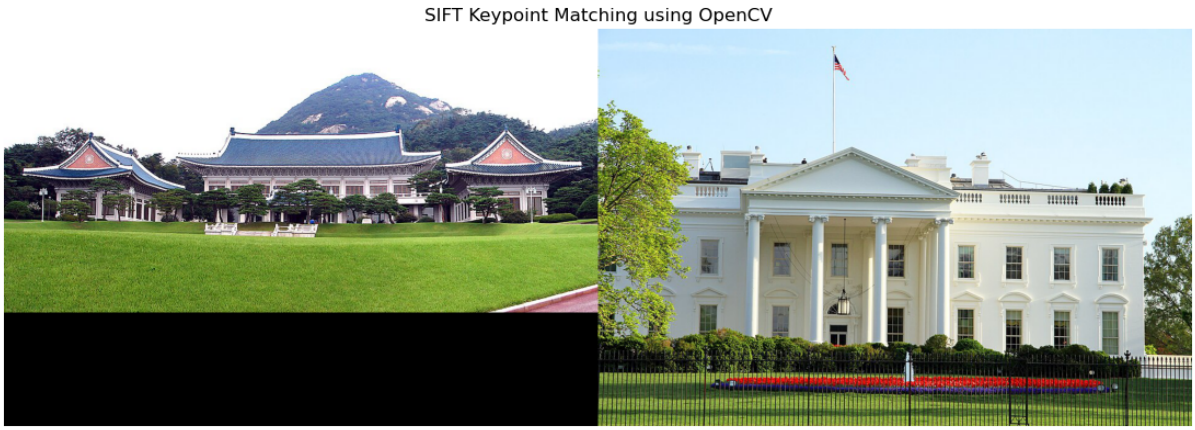


Figure 18: SIFT Keypoints matching between unrelated images

4.16 c

The two unrelated images were of the Blue House and White House 18. Keypoints in each image were 1644 and 3474 respectively. But overlapping matches with a Lowe's ratio test threshold of 0.4 resulted in 0 matches.
Using a basic patch based descriptor would result in more false positives by just matching the same textures and pattern which appear in both images.