Q1.2 Difference of Gaussian Pyramid for “model_chickenbroth.jpg”

Q1.5 Calculated Keypoints for "model_chickenbroth.jpg"

This one is chickenbroth_05.jpg

With a few exceptions, the keypoints are all located on edges, proving a successful implementation of the Difference of Gaussian pyramid.

Q2.4 BRIEF Examples

Figure 1 model_chickenbroth.jpg and chickenbroth_01.jpg

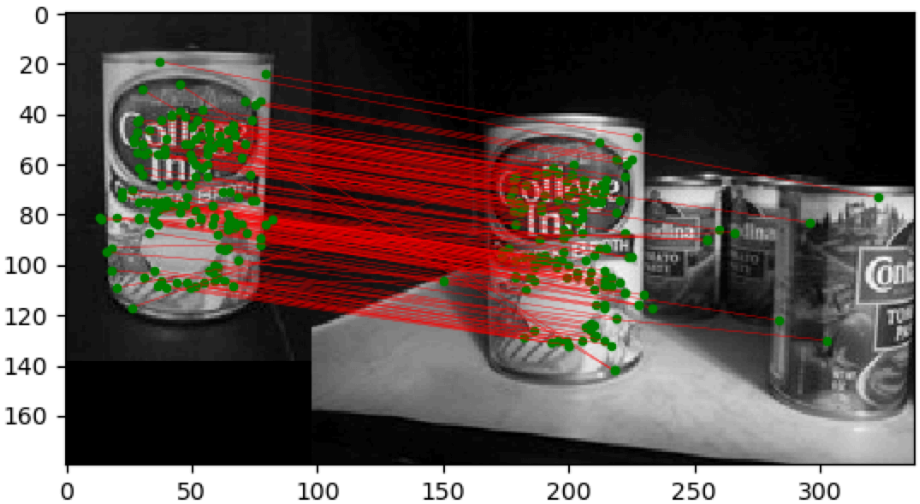


Figure 2 incline_L.png and incline_R.png

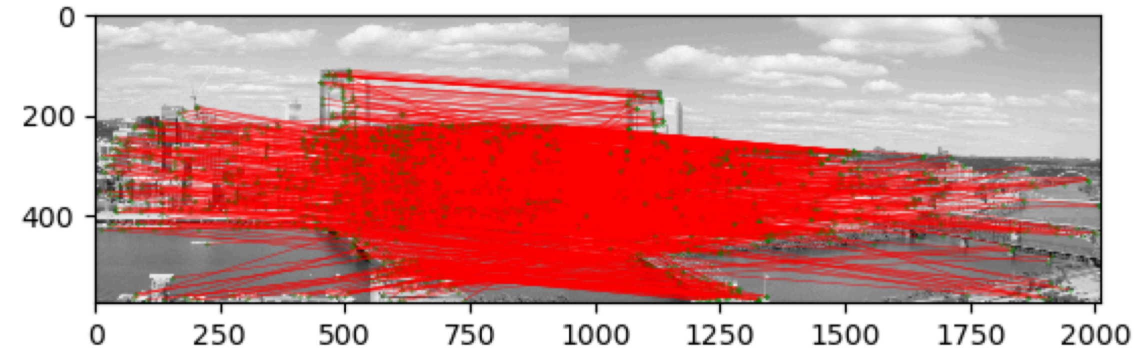


Figure 3 pf_scan_scaled.jpg and pf_desk.jpg

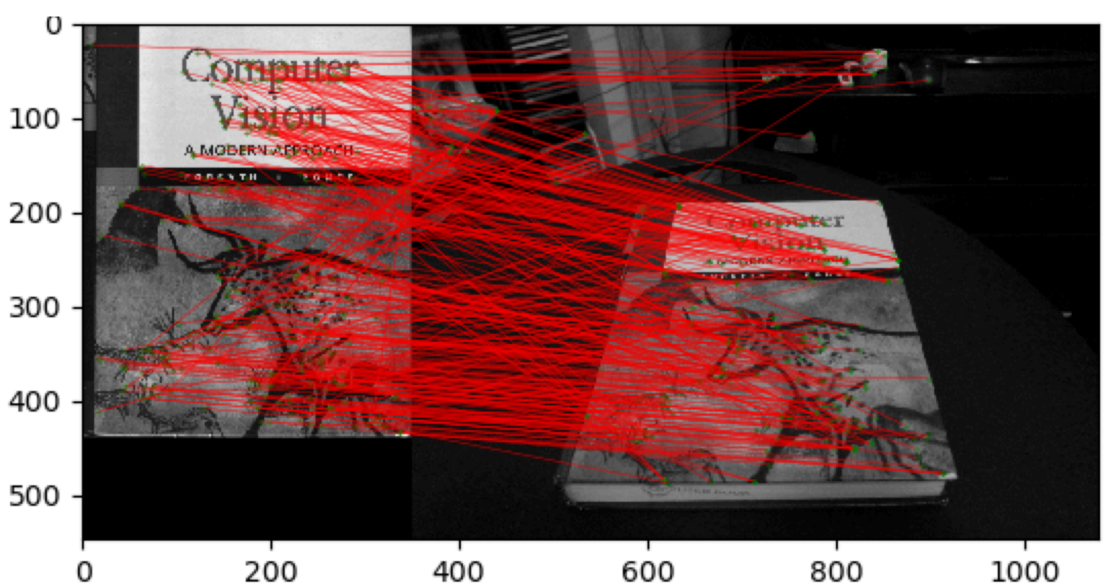


Figure 4 pf_scan_scaled.jpg and pf_floor_rot.jpg

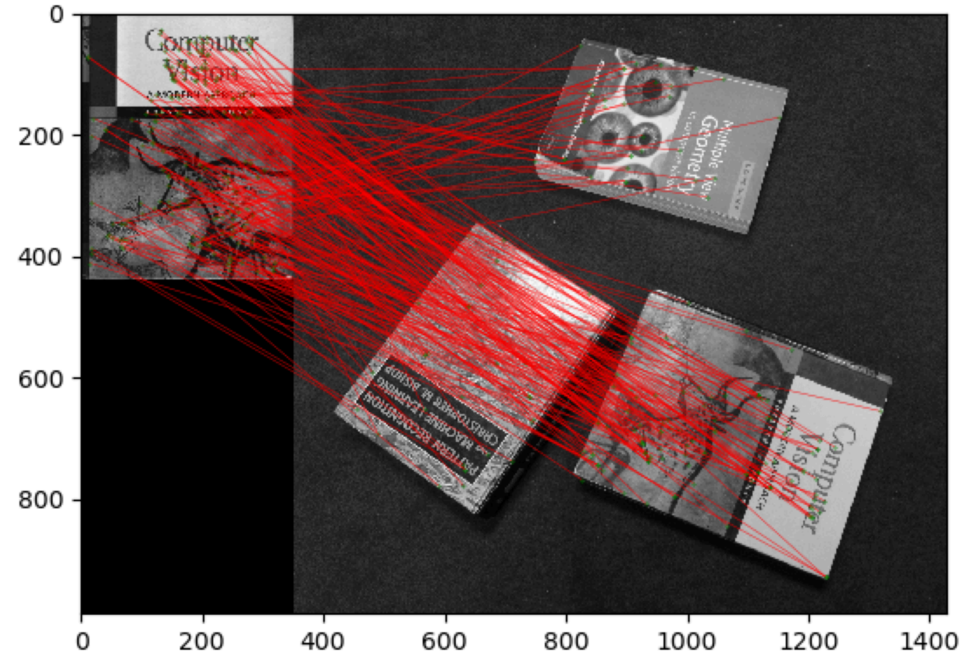


Figure 5 pf_scan_scaled.jpg and pf_floor.jpg

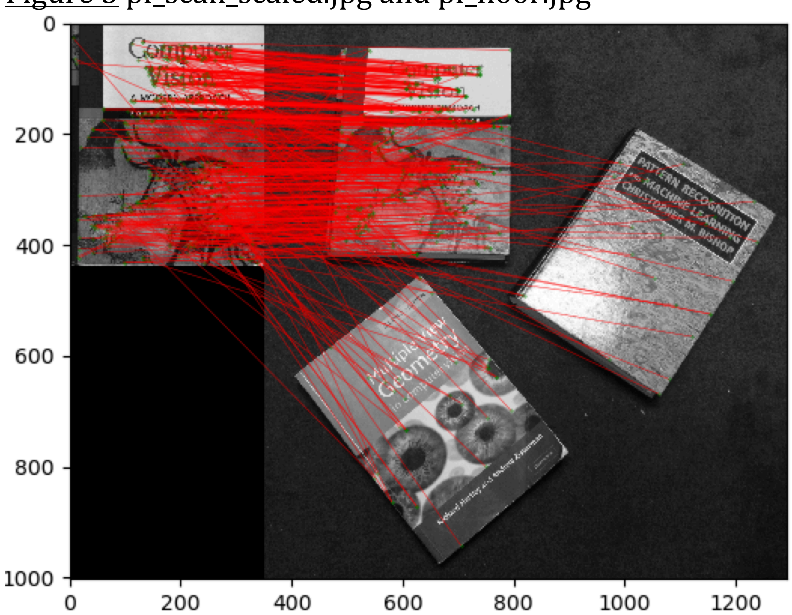


Figure 6 pf_scan_scaled.jpg and pf_pile.jpg

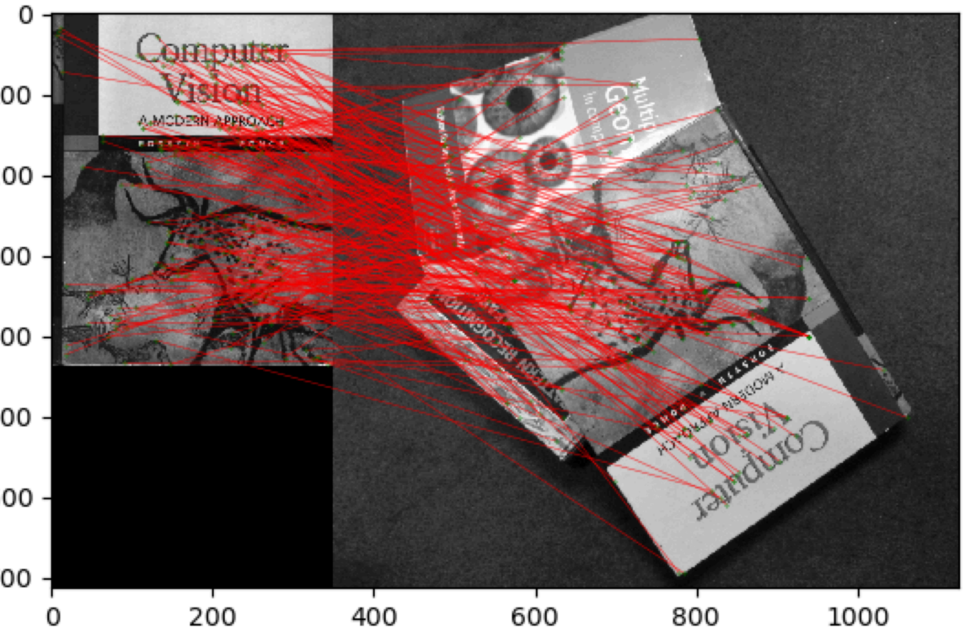
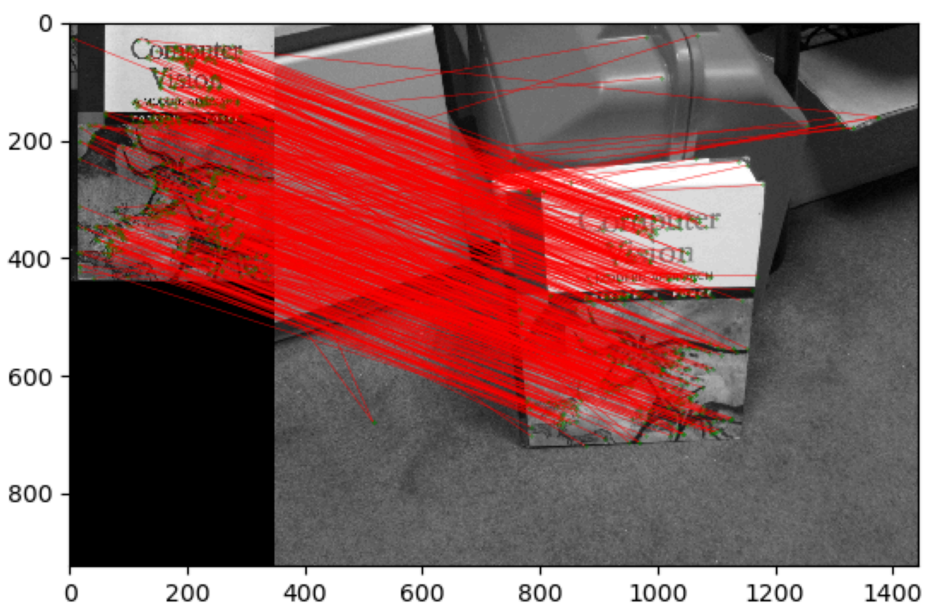
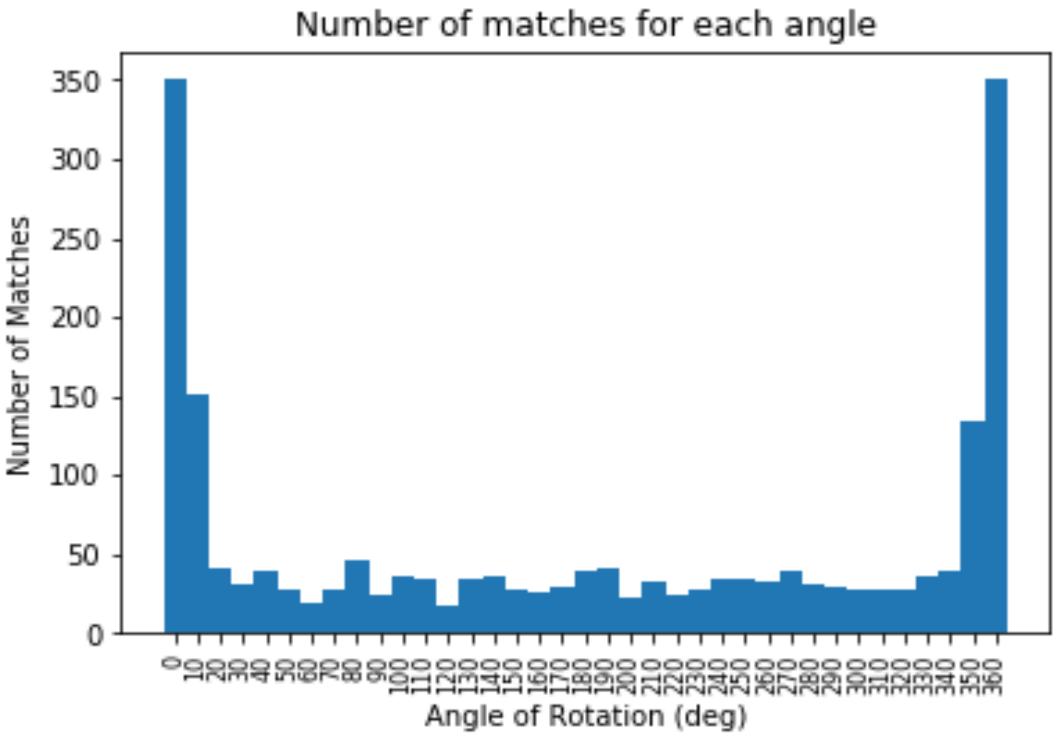


Figure 7 pf_scan_scaled.jpg and pf_stand.jpg



Judging by these results, it is very obvious that BRIEF performs very poorly when the target points are rotated. It will find some correct matches if the second image is rotated, but judging by the fact that nearly all of the points are matched in Figure 7, it is safe to say that BRIEF performance is maximized when the two images share the same orientation.

Q2.5 BRIEF Rotation Histogram



The descriptor behaves poorly under rotations because the test patch indices that BRIEF uses do not change under rotation, which will produce difference descriptor results for an original vs a rotated keypoint.

Take as an example an edge point on the top of the can. The point has black above it and gray below. When we rotate 180 degrees, the rotated image has gray above it and black below. We then draw our 9x9 patch around each of our keypoints. Let's assume the X patch location is (0,0) and the Y patch location is (81,81). In the original image, the value at X is black (zero) and the value at Y is gray (higher), so that descriptor bit will get a value of 1 because im[X] < im[Y]. In the rotated image, the value at X is gray and the value at Y is black, so that same descriptor bit will get a value of 0.

Using this example we can see that because the indices within the keypoint patches do not change, it follows that BRIEF performs very poorly under rotation, and very well with little/no rotation.

Q3.1.1 Derivation of 2N linear equations used to solve for homography matrix

$$\lambda_\mathrm{n}\tilde{\mathbf{x}}_\mathbf{n} = \mathbf{H}\tilde{\mathbf{u}}_{\boldsymbol{n}} \quad \text{for n = 1:N}$$

$$\lambda_\mathrm{n}\begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = \mathbf{H}\tilde{\mathbf{u}}_{\boldsymbol{n}} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}\begin{bmatrix} u_n \\ v_n \\ 1 \end{bmatrix}$$

$$\lambda_\mathrm{n}x_\mathrm{n} = h_{11}u_n + h_{12}v_n + h_{13}$$

$$\lambda_\mathrm{n}y_\mathrm{n} = h_{21}u_n + h_{22}v_n + h_{23}$$

$$\lambda_\mathrm{n} = h_{31}u_n + h_{32}v_n + h_{33}$$

Dividing by $\boldsymbol{\lambda_\mathrm{n}}$ →

$$x_\mathrm{n} = \frac{h_{11}u_n + h_{12}v_n + h_{13}}{h_{31}u_n + h_{32}v_n + h_{33}}$$

$$y_\mathrm{n} = \frac{h_{21}u_n + h_{22}v_n + h_{23}}{h_{31}u_n + h_{32}v_n + h_{33}}$$

Rearranging →

$$u_nh_{11} + v_nh_{12} + h_{13} - x_\mathrm{n}u_nh_{31} - x_\mathrm{n}v_nh_{32} - x_\mathrm{n}h_{33} = 0$$

$$-u_nh_{21} - v_nh_{22} - h_{23} + y_\mathrm{n}h_{31}u_n + y_\mathrm{n}v_nh_{32} + y_\mathrm{n}h_{33} = 0$$

Matrix form→

$$\begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & y_1u_1 & y_1v_1 & y_1 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -x_1v_1 & -x_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -u_N & -v_N & -1 & y_\mathrm{N}u_N & y_\mathrm{N}v_N & y_\mathrm{N} \\ u_N & v_N & 1 & 0 & 0 & 0 & -x_\mathrm{N}u_N & -x_\mathrm{N}v_N & -x_\mathrm{N} \end{bmatrix}\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix}$$

Q3.1.2 How many elements are there in h?

There are 9 elements in h, which correspond to the 9 elements of the homography matrix H.

Q3.1.3 How many point pairs (correspondences) are required to solve this system?

4 point pairs are required to solve this system. Each point correspondence yields two equations, so we have a system of 8 equations and 9 unknowns. But, because the homography only has 8 degrees of freedom, this is sufficient information to solve the system.

Q3.1.4 Show how to estimate the elements in h to find a solution to minimize this homogeneous linear least squares system.

To minimize the homogeneous linear least squares system, you first compute the singular value decomposition of **A** to get three matrices: **U**, **S**, and $\mathbf{V^T}$. Next, you find the row of **S** that contains the smallest singular value (the 8th row if we only have 8 equations). The index of that row is used to index into $\mathbf{V^T}$. That row of $\mathbf{V^T}$ is then reshaped into a 3x3 matrix to form the homography matrix. Depending on the application of the homography, you can then divide all elements by the 9th element in the homography so that the bottom right element in the matrix is equal to 1.

Q6.1 Panorama with Clipping

This is incline_R.png warped with output dimensions:
output height equal to the input height and with
output width equal to sum of the widths of both incline images

This is the same image and output dimensions as above but with incline_L.png overlayed on the left half of the frame. The image blending was handled by warping both images into the same sized frame and using np.maximum() to take the maximum pixel value each location.

6.2 Panorama using imageStitching_noClip()

Q6.3 Panorama using generatePanorama()

Q7.2 Projection of tennis ball on the CV textbook