Q1.1 (5 points): Prove that there exists a homography H that satisfies equation 1 given two 3 by 4 camera projection matrices P1 and P2 corresponding to the two cameras and a plane π.

Q1.2 (15 points):

1. How many degrees of freedom does h have? (3 points)  
2. How many point pairs are required to solve h? (2 points)  
3. Derive Ai. (5 points)  
4. When solving Ah = 0, in essence you’re trying to find the h that exists in the null space of A. What that means is that there would be some non-trivial solution for h such that that product Ah turns out to be 0. What will be a trivial solution for h? Is the matrix A full rank? Why/Why not? What impact will it have on the singular values (i.e. eigenvalues of ATA)? (5 points)

Q1.4.3: Limitations of the planar homography

Why is the planar homography not completely sufficient to map any arbitrary scene image to another viewpoint? State your answer concisely in one or two sentences.

Planar homography is not completely sufficient to map any arbitrary scene image to another viewpoint because the assumption that all points lie on the same plane is not always true. There may be varying depths and 3D structures.

Q2.1.1 (5 points): FAST Detector

How is the FAST detector different from the Harris corner detector that you’ve seen in the lectures? Can you comment on its computational performance compared to the Harris corner detector?

The main difference between the FAST (features accelerated segment test) detector and the Harris corner detector is the method they use in determining corners. The FAST detector utilizes pixel intensity/intensity threshold as its corner-detecting logic. A candidate pixel will have a surrounding pixel ring, usually a circle of 16 pixels (Bresenham circle) for corner classification. Since the FAST detector uses a binary decision, it is more computationally efficient but less robust.

On the other hand, the Harris corner detector relies on matrix construction and computing eigenvalues for corner detection. For example, when both eigenvalues are large, the algorithm will classify it as a corner. The Harris corner detector method is more math-intensive and can be more robust but less computationally efficient.

When comparing computational performance between the two algorithms, the FAST detector is significantly faster than the Harris corner detector. The Harris corner detector will be better suited for applications that require high accuracy. The FAST detector is more beneficial for real-time implementations.

Q2.1.2 (5 points): BRIEF Descriptor

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

The main difference between the BRIEF (binary robust independent elementary features) and the filter banks is the underlying technique used in feature detection. The BRIEF descriptor uses pixel intensity differences to represent an image patch as a binary string. It yields higher recognition rates and is more computationally efficient. However, the BRIEF descriptor is less accurate when dealing with rotation and scaling.

On the other hand, filter banks apply linear filters (Gaussian, Laplacian of Gaussian, etc.) to extract unique features such as edges, gradients, orientations, etc. Filter banks are more robust and computationally heavy, better suited for images with more complex features or transformations. Any one of these filter banks can be used as a descriptor, but one must consider the trade-off between robustness and implementation time.

Q2.1.3 (5 points): Matching Methods

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 have over a more conventional Euclidean distance measure in our setting?

When matching interest points with BRIEF descriptors, the Hamming distance is more computationally and memory efficient compared to the conventional Euclidean distance. The faster implementation is due to the XOR operation in calculating Hamming distance compliments the binary string data used in BRIEF descriptors.

Q2.1.5 (10 points): Feature Matching and Parameter Tuning

Conduct a small ablation study by running displayMatch.py with various sigma and ratio values. Include the figures displaying the matched features with various parameters in your writeup, and explain the effect of these two parameters respectively.

Q2.1.6: BRIEF and Rotations

Write a script briefRotTest.py that: Visualize the histogram and the feature matching result

at three sufficiently different orientations and include them in your write-up. Explain why

you think the BRIEF descriptor behaves this way. Please include the code snippet in your

writeup.

Q2.1.7 (Extra Credit): Improving Performance

1. As we have seen, BRIEF is not rotation invariant. Design a simple fix to solve this

problem using the tools you have developed so far (think back to edge detection and/or Harris corner’s covariance matrix). You are not allowed to use any additional OpenCV or Scikit-Image functions. Include the code in your PDF, and explain your design decisions and how you selected any parameters that you use. Demonstrate the effectiveness of your algorithm on image pairs related by large rotation.

2. This implementation of BRIEF has some scale invariance, but there are limits. What

happens when you match a picture to the same picture at half the size? Look to section 3

of [Lowe2004], for a technique that will make your detector more robust to changes in scale.

Implement it and demonstrate it in action with several test images. Include your code and the test images in your PDF. You are not allowed to call any additional OpenCV or Scikit-Image functions. You may simply rescale some of the test images we have given you.

From the above study, we can observe that higher number of iterations and lower tolerances generates better results compared to lower number of iterations and higher tolerances. Number of iterations allows the algorithm to sample more data, which increases the robustness but requires more time for computation. On the other hand, the inlier tolerance value dictates the inlier selection strictness. High tolerance leads to more matching features but lower accuracy (outliers).

References:

<https://en.wikipedia.org/wiki/Harris_corner_detector>

<https://en.wikipedia.org/wiki/Features_from_accelerated_segment_test>

<https://medium.com/@deepanshut041/introduction-to-brief-binary-robust-independent-elementary-features-436f4a31a0e6>

<https://www.tutorialspoint.com/what-is-hamming-distance>