

# Homework 3 - Applied Computer Vision

kipngeno koech - bkoech

February 20, 2025

## 1 Answers to theory Questions

### 1.1 Homographies

#### 1.1.1 The Direct Linear Transform (DLT) algorithm

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:

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

where  $\mathbf{x}_1^i = \begin{pmatrix} x_1^i \\ y_1^i \\ 1 \end{pmatrix}^T$  are in homogeneous coordinates,  $\mathbf{x}_1^i \in \mathbf{x}_1$  and  $\mathbf{H}$  is a  $3 \times 3$  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)

$$\mathbf{h} = \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix}$$

Since  $\mathbf{H}$  is a  $3 \times 3$  matrix, it initially has 9 elements. However, homographies are defined up to a scale factor, meaning that multiplying  $\mathbf{H}$  by a non-zero scalar does not change the transformation it represents. Therefore, we can fix one of the elements (usually  $h_{33}$ ) to 1, reducing the degrees of freedom by 1. Thus,  $\mathbf{h}$  has 8 degrees of freedom.

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

$$\mathbf{A} = \begin{pmatrix} -x_2^i & -y_2^i & -1 & 0 & 0 & 0 & x_1^i x_2^i & x_1^i y_2^i & x_1^i \\ 0 & 0 & 0 & -x_2^i & -y_2^i & -1 & y_1^i x_2^i & y_1^i y_2^i & y_1^i \end{pmatrix}$$

Each point pair contributes 2 rows to  $\mathbf{A}$ , so we need at least 4 point pairs to solve  $\mathbf{h}$ .

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

For homogenous coordinates  $\mathbf{x}_1^i = \begin{pmatrix} x_1^i \\ y_1^i \\ z_1^i \end{pmatrix}^T$  and  $\mathbf{x}_2^i = \begin{pmatrix} x_2^i \\ y_2^i \\ 1 \end{pmatrix}^T$ , we can write the relation as:

$$\begin{pmatrix} x_1^i \\ y_1^i \\ z_1^i \end{pmatrix} \equiv \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_2^i \\ y_2^i \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x_1^i \\ y_1^i \\ z_1^i \end{pmatrix} \equiv \begin{pmatrix} h_{11}x_2^i + h_{12}y_2^i + h_{13} \\ h_{21}x_2^i + h_{22}y_2^i + h_{23} \\ h_{31}x_2^i + h_{32}y_2^i + h_{33} \end{pmatrix}$$

To get  $\mathbf{A}_i$ , we can write the above equation as:

$$\begin{pmatrix} x_1^i \\ y_1^i \\ z_1^i \end{pmatrix} \equiv \begin{pmatrix} h_{11}x_2^i + h_{12}y_2^i + h_{13} \\ h_{21}x_2^i + h_{22}y_2^i + h_{23} \\ h_{31}x_2^i + h_{32}y_2^i + h_{33} \end{pmatrix} \equiv \begin{pmatrix} h_{11}x_2^i + h_{12}y_2^i + h_{13} - x_1^i \\ h_{21}x_2^i + h_{22}y_2^i + h_{23} - y_1^i \\ h_{31}x_2^i + h_{32}y_2^i + h_{33} - z_1^i \end{pmatrix} = 0$$

We can then write  $\mathbf{A}_i \mathbf{h}$  as:

$$\mathbf{A}_i = \begin{pmatrix} -x_2^i & -y_2^i & -1 & 0 & 0 & 0 & x_1^i x_2^i & x_1^i y_2^i & x_1^i \\ 0 & 0 & 0 & -x_2^i & -y_2^i & -1 & y_1^i x_2^i & y_1^i y_2^i & y_1^i \end{pmatrix} \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix} = 0$$

Let us extract the first row of  $\mathbf{A}_i$ :

$$\mathbf{A}_i = \begin{pmatrix} -x_2^i & -y_2^i & -1 & 0 & 0 & 0 & x_1^i x_2^i & x_1^i y_2^i & x_1^i \\ 0 & 0 & 0 & -x_2^i & -y_2^i & -1 & y_1^i x_2^i & y_1^i y_2^i & y_1^i \end{pmatrix}$$

4. When solving  $\mathbf{A}\mathbf{h} = 0$ , 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 the product  $\mathbf{A}\mathbf{h}$  turns out to be 0.

(a) What will be a trivial solution for  $\mathbf{h}$ ? Is the matrix  $\mathbf{A}$  full rank? Why/Why not?

$$\mathbf{h} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

This is a trivial solution for  $\mathbf{h}$  since it does not change the transformation represented by  $\mathbf{H}$ . The matrix  $\mathbf{A}$  is not full rank because it is a rank-2 matrix. This is because the rank of  $\mathbf{A}$  is at most 2, since the rows of  $\mathbf{A}$  are linearly dependent. This is because the points  $\mathbf{x}_1^i$  and  $\mathbf{x}_2^i$  are related by a homography, which is a linear transformation.

- (b) What impact will it have on the singular values? What impact will it have on the singular vectors?

The singular values of  $\mathbf{A}$  will be 0, since the rank of  $\mathbf{A}$  is 2. The singular vectors corresponding to the singular values will be

$$\text{the null space of } \mathbf{A}, \text{ which is the space spanned by the trivial solution } \mathbf{h} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

## 1.2 Computing Planar Homographies

### 1.2.1 FAST Detector (5 points)

1. 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?

The FAST detector is different from the Harris corner detector in that it is a corner detector that uses a different criterion to determine corners. The FAST detector uses a circle of 16 pixels around the pixel of interest, and checks if there are 12 contiguous pixels that are either brighter or darker than the pixel of interest. If such a circle exists, the pixel is classified as a corner. The Harris corner detector, on the other hand, uses the eigenvalues of the structure tensor to determine corners. The FAST detector is computationally faster than the Harris corner detector because it only requires a simple thresholding operation to determine corners, while the Harris corner detector requires the computation of the structure tensor and the eigenvalues of the structure tensor.

### 1.2.2 BRIEF Descriptor (5 points)

1. 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?

The BRIEF descriptor is different from the filterbanks you've seen in the lectures in that it is a binary descriptor, while the filterbanks are real-valued descriptors. The BRIEF descriptor represents the image region corresponding to the detected feature point as a binary string of 1s and 0s, while the filterbanks represent the image region as a set of real-valued numbers. The BRIEF descriptor is designed to be computationally efficient and memory efficient, while the filterbanks are designed to capture the spatial structure of the image region. You could use the filterbanks as a descriptor, but it would not be as efficient as the BRIEF descriptor in terms of computation and memory.

### 1.2.3 Matching Methods (5 points)

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

### 1.2.4 Feature Matching (10 points)

1. Use the provided helper function `plotMatches` to visualize your matched points and include the resulting image in your write-up.

### 1.2.5 BRIEF and Rotations

Include visualizations of the feature matching results at three different orientations. Explain why you think the BRIEF descriptor behaves this way.

## 1.3 Homography Computation

visualizations

## 1.4 RANSAC

1. At this point you should notice that although the image is being warped to the correct location, it is not filling up the same space as the book. Why do you think this is happening? How would you modify `hp cover.jpg` to fix this issue?
2. Show us your final image, composite img, generated by your script `HarryPotterize.py`.

## 1.5 Creating your Augmented Reality application

visualizations

## 1.6 Extra Credit

Include your original images and the panorama result image