16-385 Computer Vision, Fall 2019

Homework 4
Augmented Reality with Planar Homographies

Due Date: Monday November 4, 2019 23:59

1 Homograhies

1.1 Homography

As the figure of pdf, x_π is all points lying in plane π. And x and x' is the camera views in camera C and C' whose camera matrix is P_1 and P_2 each. Then we can denote $X = P_1 X_\pi$ and $X' = P_2 X_\pi$. For combining two equations, we can use pseudo inverse of P_2 as P_2'. Then X_1 is represented as $X_1 = P_1 P_2' X'$

1.2 Correspondences

1.2.1 How many degrees of freedom does h have?

8

1.2.2 How many point pairs are required to solve h?

4

1.2.3 Derive A_i

$$\lambda \mathbf{x_1^i} = \mathbf{H}\mathbf{x_2^i}$$

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

$$then, \mathbf{x_1^i} \times \mathbf{H}\mathbf{x_2^i} = 0$$

$$\begin{bmatrix} x_1^i \\ y_1^i \\ 1 \end{bmatrix} \times \begin{bmatrix} \mathbf{h_1^T x_2^i} \\ \mathbf{h_2^T x_2^i} \\ \mathbf{h_3^T x_2^i} \end{bmatrix} = \begin{bmatrix} y_1^i \mathbf{h_3^T x_2^i} - \mathbf{h_2^T x_2^i} \\ \mathbf{h_1^T x_2^i} - x_1^i \mathbf{h_3^T x_2^i} \\ x_1^i \mathbf{h_2^T x_2^i} - y_1^i \mathbf{h_1^T x_2^i} = \end{bmatrix} = 0$$

$$\begin{bmatrix} 0^T & -\mathbf{x_2^i} & y_1^i \mathbf{x_2^i} \\ \mathbf{x_2^i} & 0^T & -x_1^i \mathbf{x_2^i} \\ -y_1^i \mathbf{x_2^i} & x_1^i \mathbf{x_2^i} & 0^T \end{bmatrix} \begin{pmatrix} \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \end{pmatrix} = 0$$

Therefore, we can define the matrix as A_i

1.2.4 When solving $\mathbf{Ah} = 0$, 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 eigen values? What impact will it have on the eigen vectors?

The trivial solution for $\mathbf{h}$ is all zero solution.
The matrix $\mathbf{A}$ is not a full rank.
Because it is not singular, so it has inverse transform.
In this case, rank of $\mathbf{A}$ is 8, and the dimension of $\mathbf{A^T A}$ is 9×9. It has one zero eigen value and each corresponding value is the eigen vector. It will be a solution of h. '

1.3 Homography under rotation

In this case, $\mathbf{x_1} = \mathbf{K_1}\left[\mathbf{I0}\right]\mathbf{X}$ and $\mathbf{x_2} = \mathbf{K_2}\left[\mathbf{R0}\right]\mathbf{X}$.

Then, we can denote $\mathbf{x_1} = \mathbf{K}\left[\mathbf{I0}\right]\mathbf{X} = \mathbf{K}[\mathbf{I0}]\begin{bmatrix}\mathbf{X}\\\mathbf{Y}\\\mathbf{Z}\\1\end{bmatrix} = \mathbf{K_1}\begin{bmatrix}\mathbf{X}\\\mathbf{Y}\\\mathbf{Z}\end{bmatrix}$.

So the point $\mathbf{X}$ can be represented as

$$\mathbf{X} = \begin{bmatrix}\mathbf{X}\\\mathbf{Y}\\\mathbf{Z}\\1\end{bmatrix} = \begin{bmatrix}\mathbf{K_1}^{-1}\mathbf{x_1}\\1\end{bmatrix}$$

Therefore, the point $\mathbf{x_2}$ is

$$\mathbf{x_2} = \mathbf{K_2}[\mathbf{R0}]\mathbf{X} = \mathbf{K_2}[\mathbf{R0}]\begin{bmatrix}\mathbf{X}\\\mathbf{Y}\\\mathbf{Z}\\1\end{bmatrix} = \mathbf{K_2}[\mathbf{R0}]\begin{bmatrix}\mathbf{K_1}^{-1}\mathbf{x_1}\\1\end{bmatrix} = \mathbf{K_2}\mathbf{R}\mathbf{K_1}^{-1}\mathbf{x_1}$$

So, $\mathbf{H} = \mathbf{K_2}\mathbf{R}\mathbf{K_1}^{-1}$

1.4 Understanding homographies under rotation

If the case of rotation $\mathbf{R}$ is on xy plane, then the rotation matrix $\mathbf{R}$ is

$$\mathbf{R} = \begin{bmatrix}-cos\theta & -sin\theta & 0\\ sin\theta & cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

We can compute

$$
\begin{aligned}
\mathbf{H} &= \mathbf{K}\mathbf{R}_\theta\mathbf{K}^{-1}\mathbf{K}\mathbf{R}_\theta\mathbf{K}^{-1} \\
&= \mathbf{K}\mathbf{R}_\theta\mathbf{R}_\theta\mathbf{K}^{-1} \\
&= \mathbf{K}\begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{K}^{-1} \\
&= \mathbf{K}\begin{bmatrix} cos^2\theta - sin^2\theta & -2sin\theta cos\theta & 0 \\ 2sin\theta cos\theta & cos^2\theta - sin^2\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{K}^{-1} \\
&= \mathbf{K}\begin{bmatrix} cos(2\theta) & -sin(2\theta) & 0 \\ sin(2\theta) & cos(2\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}\mathbf{K}^{-1} \\
&= \mathbf{K}\mathbf{R}_{\mathbf{2}\theta}\mathbf{K}^{-1}
\end{aligned}
$$

The case of rotation $\mathbf{R}$ on another plane is similar.

1.5 Limitations of the planar homography

In the case of between the subregions of the two images, there exists different homographies which can correspond on the viewpoints of the subregions of the same planar.
Therefore, it is not completely sufficient for the planar homographies to map any arbitrary image to another viewpoint.

1.6 Behavior of lines under perspective projections

Let's denote the 3D points as $\mathbf{X_1}, \mathbf{X_2} and \mathbf{X_3}$. Then, we can represent the difference of two points as $\mathbf{X_1} - \mathbf{X_2}$ and $\mathbf{X_2} - \mathbf{X_3}$ and they is proportionate. So it can represent as $\mathbf{X_1} - \mathbf{X_2} = K(\mathbf{X_2} - \mathbf{X_3})$. For projection, apply $\mathbf{P}$ to each side. $\mathbf{P}(\mathbf{X_1} - \mathbf{X_2}) = \mathbf{P}K(\mathbf{X_2} - \mathbf{X_3})$. After calculating this, $\mathbf{x_1} - \mathbf{x_2} = K(\mathbf{x_2} - \mathbf{x_3})$. We can check that 2D points are also in line.

2 Computing Planar Homographies

2.1 Feature Detection and Matching

2.1.1 FAST Detector

Harris corner detector is method that shifting small window in image, and find corner which has significant change in all directions. However, Fast algorithm uses decision tree rather than use counting similar consecutive points for judging the point is corner or not. FAST method determines whether the point p is a corner by looking at 16 pixels in a circle with radius of 3 centered on p. So if there are more than n consecutive pixels which are brighter than p, or vise versa, then we can judge the point p as a corner point.
FAST method is optimized method for speed, so its computational performance is so fast.

2.1.2 BRIEF Descriptor

We can detect any points which get meaningful features by detector and extract meaningful features which has different formation of information by using descriptor.
In our lecture, the filter banks are applied to whole images. So we can get the meaningful information of whole images or patch of images. We can represent that image patch by using multiple filtered images. However, BRIEF is applied to specific key points. BRIEF just extracts meaningful local image feature information. And BRIEF represents random pixel location of that patch by binarized vector.

2.1.3 Matching Methods

Hamming distance is a metric for comparing two binary data strings. While comparing two binary strings of equal length, Hamming distance is the number of bit positions in which the two bits are different.
Nearest Neighbor is a method used for classification and regression. In this case, the input consists of the closest training example in the feature space. Then the output depends on whether NN is used for classification or regression.
At BRIEF, it using hamming distance for metric and detect the highest nearest neighbor example for matching points. It is better than euclidean distance, it is used for error detection or error correction when data is composed of binary data.

2.1.4 Feature Matching

2.1.5 BRIEF and Rotations

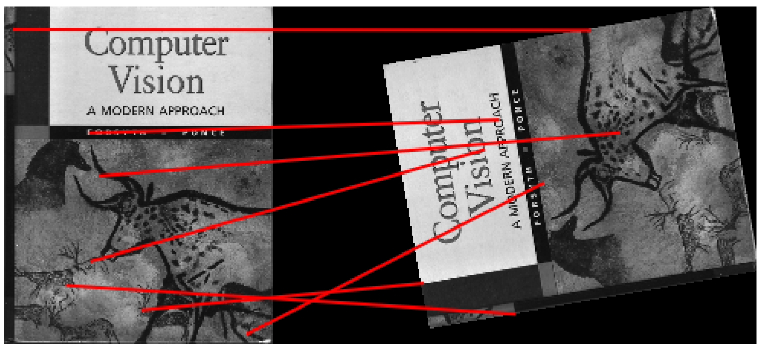

Figure 1: Rotate image for 100 degrees

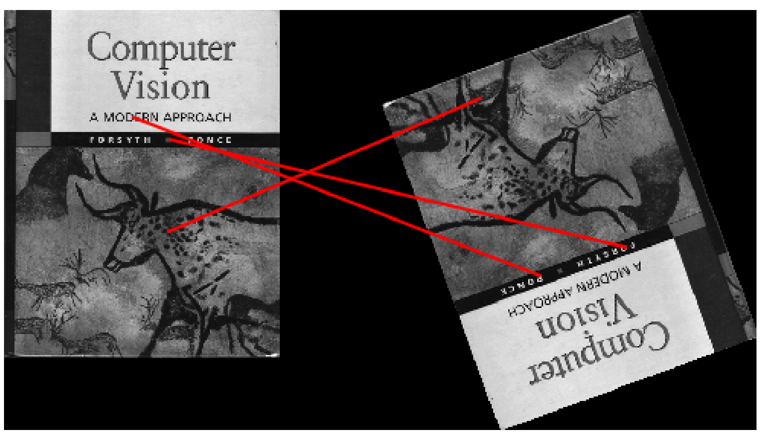

Figure 2: Rotate image for 200 degrees

The above figures are the case of different orientations. BRIEF make 1D random index for location. So if the image is rotated, the index completely changes. Therefore, the BRIEF descriptor can not catch the matches if the image is rotated.

The above figure is the histogram of the counts of matches for each orientation.

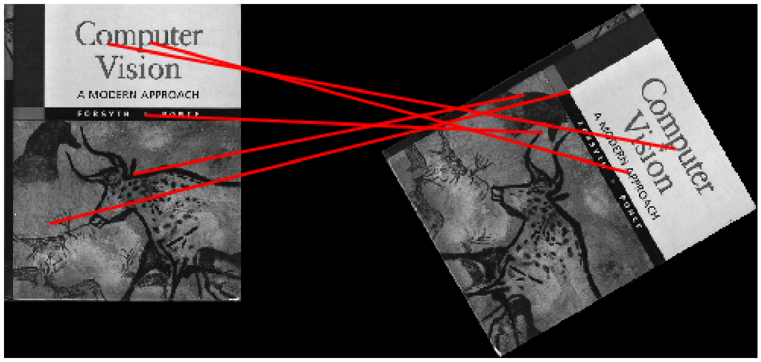

Figure 3: Rotate image for 300 degree

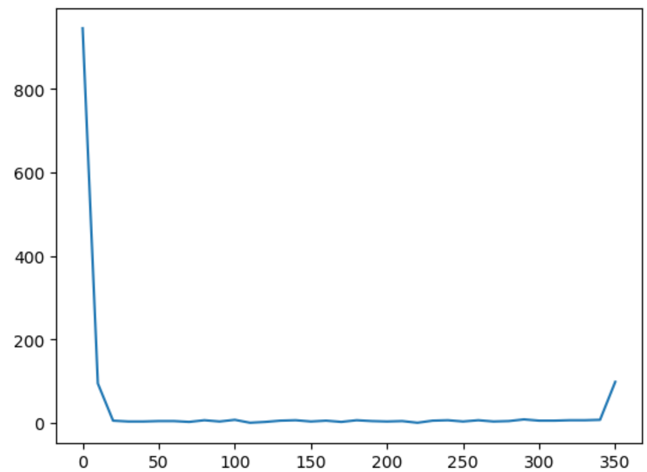

Figure 4: histogram of the count of matches for each orientation

2.2 Homography Computation

2.2.1 Computing the Homography

2.3 Homography Normalization

2.3.1 Homography with normalization

2.4 RANSAC

2.4.1 Implement RANSAC for computing a homography

2.5 Automated Homography Estimation and Warping

2.5.1 Putting it together

3 Creating your Augmented Reality application

3.1 Incorporating video