<div align="center">

Homework 3

Zongwen Mu, Andrew ID: zongwenm

</div>

1 Lucas-Kanade Tracking

Q1.1

For a warp function:

$$\mathcal{W}(\mathbf{x};\mathbf{p}) = \begin{bmatrix} \mathcal{W}_x \\ \mathcal{W}_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_1 \\ 0 & 1 & p_2 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{1}$$

$$= \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} \tag{2}$$

Therefore:

$$\frac{\partial \mathcal{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T} = \begin{bmatrix} \frac{\partial \mathcal{W}_x(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}_1^T} & \frac{\partial \mathcal{W}_x(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}_2^T} \\ \frac{\partial \mathcal{W}_y(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}_1^T} & \frac{\partial \mathcal{W}_y(\mathbf{x},\mathbf{p})}{\partial \mathbf{p}_2^T} \end{bmatrix} \tag{3}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4}$$

Since we are trying to obtain:

$$\arg\min_{\Delta p} \sum_{\mathbf{x}\in\mathbb{N}} \left\| \mathcal{I}_{t+1}\left(\mathbf{x}' + \Delta\mathbf{p}\right) - \mathcal{I}_t\left(\mathbf{x}\right) \right\|_2^2 \tag{5}$$

$$= \arg\min_{\Delta p} \sum_{\mathbf{x}\in\mathbb{N}} \left\| \mathcal{I}_{t+1}\left(\mathbf{x}'\right) + \frac{\partial\mathcal{I}_{t+1}\left(\mathbf{x}'\right)}{\partial\mathbf{x}'^T}\frac{\partial\mathcal{W}\left(\mathbf{x};\mathbf{p}\right)}{\partial\mathbf{p}^T}\Delta\mathbf{p} - \mathcal{I}_t\left(\mathbf{x}\right) \right\|_2^2 \tag{6}$$

$$= \arg\min_{\Delta p} \sum_{\mathbf{x}\in\mathbb{N}} \left\| \frac{\partial\mathcal{I}_{t+1}\left(\mathbf{x}'\right)}{\partial\mathbf{x}'^T}\frac{\partial\mathcal{W}\left(\mathbf{x};\mathbf{p}\right)}{\partial\mathbf{p}^T}\Delta\mathbf{p} - \left[\mathcal{I}_t\left(\mathbf{x}\right) - \mathcal{I}_{t+1}\left(\mathbf{x}'\right)\right] \right\|_2^2 \tag{7}$$

Where $\mathbf{x}' = \mathbf{x} + \mathbf{p}$.

Therefore, we have:

$$\mathbf{A} = \frac{\partial\mathcal{I}_{t+1}\left(\mathbf{x}'\right)}{\partial\mathbf{x}'^T}\frac{\partial\mathcal{W}\left(\mathbf{x};\mathbf{p}\right)}{\partial\mathbf{p}^T} \tag{8}$$

$$= \begin{bmatrix} \frac{\partial\mathcal{I}_{t+1}\left(\mathbf{x}_1'\right)}{\partial\mathbf{x}_1'^T} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{\partial\mathcal{I}_{t+1}\left(\mathbf{x}_D'\right)}{\partial\mathbf{x}_D'^T} \end{bmatrix} \begin{bmatrix} \frac{\partial\mathcal{W}(\mathbf{x_1};\mathbf{p})}{\partial\mathbf{p}^T} \\ \vdots \\ \frac{\partial\mathcal{W}(\mathbf{x_D};\mathbf{p})}{\partial\mathbf{p}^T} \end{bmatrix} \tag{9}$$

$$\mathbf{b} = \mathcal{I}_t\left(\mathbf{x}\right) - \mathcal{I}_{t+1}\left(\mathbf{x}'\right) \tag{10}$$

$$= \begin{bmatrix} \mathcal{I}_t\left(\mathbf{x_1}\right) - \mathcal{I}_{t+1}\left(\mathbf{x_1'}\right) \\ \vdots \\ \mathcal{I}_t\left(\mathbf{x_D}\right) - \mathcal{I}_{t+1}\left(\mathbf{x_D'}\right) \end{bmatrix} \tag{11}$$

To guarantee a unique solution for $\Delta\mathbf{p}$, $\mathbf{A^T A}$ needs to be full rank, in other words, $det\left(\mathbf{A^T A}\right) \neq 0$

2. How many point pairs are required to solve h?

4 point pairs are required.

3. Derive A_i

In part 1, we acquired equations:

$$x_1 = h_{11}x_2 + h_{12}y_2 + h_{13} \tag{12}$$
$$y_1 = h_{21}x_2 + h_{22}y_2 + h_{23} \tag{13}$$
$$1 = h_{31}x_2 + h_{32}y_2 + h_{33} \tag{14}$$

Since $\mathbf{A_i h = 0}$, we could derive $\mathbf{A_i}$ as:

$$A_i = \begin{bmatrix} -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_1 x_2 & x_1 y_2 & x_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & y_1 x_2 & y_1 y_2 & y_1 \end{bmatrix} \tag{15}$$

4.

This could be solved in the least squares fashion, by finding the eigenvector of $\mathbf{A}$, corresponding to the smallest eigenvalue of $\mathbf{A}$.

Taking the SVD of $\mathbf{A}$:

$$h = argmin\left(\left(h^T V\right)\Sigma^2\left(V^T h\right)\right) \tag{16}$$

Therefore, $\mathbf{A}$ should be full rank, and $\mathbf{h}$ is the last column of $\mathbf{V}$.

1.1 Homography Under Rotation

Since there's no translation between world frame and two cameras, and there's no rotation for the first camera, we could find a inverse projection matrix for camera 1, $M_1^{-1} = \left[\,I\ 0\,\right]^T K_1^{-1}$, and we could acquire $\mathbf{H}$ by $\boldsymbol{H} = M_1^{-1} K_2\left[\,R\ 0\,\right]$. Therefore, there exists a homography $\mathbf{H}$ between $\mathbf{x_1}$ and $\mathbf{x_2}$.

1.2 Understanding Homographies Under Rotation

Since $\mathbf{K}$ is constant, so the matrix $\mathbf{H^2}$ is corresponding to $\mathbf{R^2}$. For instance, if the camera is rotating around $\mathbf{z}$ axis:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

$$\mathbf{R^2} = \begin{bmatrix} \cos 2\theta & -\sin 2\theta & 0 \\ \sin 2\theta & \cos 2\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

Therefore, $\mathbf{H^2}$ is corresponding to 2θ.

1.3 Limitations of the Planar Homography

The motion of camera may not be a pure rotation around its center or the points are not on a plane, thus planar homography is insufficient.

1.4 Behavior of Lines Under Perspective Projections

Point in world frame $\begin{bmatrix} x\ y\ z \end{bmatrix}^T$ to point in projection $\begin{bmatrix} x'\ y' \end{bmatrix}^T$. Since for a line function in $3D$:

$$l \begin{bmatrix} x \\ y \\ z \end{bmatrix} = 0 \tag{19}$$

In 2d coordinate, we have:

$$l \begin{bmatrix} \frac{x}{z} \\ \frac{y}{z} \\ 1 \end{bmatrix} = 0 \tag{20}$$

Thus let $\frac{x}{z} = x'$ and $\frac{y}{z} = y'$, we have:

$$l \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = 0 \tag{21}$$

Therefore, the projection $\mathbf{P}$ preserve lines.

2 Computing Planar Homographies

2.1 Feature Detection and Matching

2.1.1 FAST Detector

FAST detector works by selecting pixel rather than using sliding window, and FAST detector would compare those pixels around selected pixel, and if there is enough pixels brighter or darker than selected pixel, this pixel is considered as a corner. Compared with Harris detector, FAST do not need to calculate λ for whole sliding window, which would save great energy and time in computation.

2.1.2 BRIEF Descriptor

BRIEF would create a patch around selected point and implement Gaussian Filter to this region. Then it will randomly select **N** point pairs and compare each pair to acquire a binary code in length **N** as descriptor of this patch.

2.1.3 Matching Methods

When BRIEF descriptor acquired binary codes for two interest points, Hamming distance is calculating the XOR between two descriptors. Hamming distance benefits since it would be much easier for CPU to perform XOR and bit count operations.

2.1.4 Feature Matching

Result for default parameters is shown below:

2.1.5 Feature Matching Parameter Tuning

Firstly, I used fixed $sigma = 0.15$ and tuning $ratio$, the results are shown below:

Then using fixed $ration = 0.7$, tuning sigma:

As observed in the results. with the increase of ratio, matched points number increased drastically and when the ratio dropped, there would be fewer matched points, this is due to fewer point pairs would be considered matched with the drop of ratio. And increasing sigma would decrease matched points. That is, sigma is the threshold for corner detection and increase sigma would lead to less detected corner so that there would be fewer interest points for descriptor.

2.1.6 BRIEF and Rotations

In this case, with BRIEF feature descriptor ratio $ratio = 0.7$, I visualized match result for rotation angle $60, 180, 270$, images are shown below:

The histogram is shown below:

As was observed in results, the number of matches dropped drastically as the rotation angle increased. This is because BRIEF descriptors construct a bitset descriptor for each patch by comparing randomly chosen pixels location. When rotating the image, the keypoints are bound to give different descriptors for the same patch, and the comparison would occur at different pixel intensities.

2.2 Homography Computation, Normalization and RANSAC

2.2.4 Putting it together

Warped image is shown below:

Noticed that the hp_cover image was placed in the right place but with a wrong size. The way to solve this problem could be increase iteration number to find a better homography, resize hp_cover to a more precise shape, and decrease tolerance.

2.2.5   RANSAC Parameter Tuning

Since increase iteration number could increase the possibility to find optimized homography, so first I used default tolerance 2 and changed numbers of iteration, the composited image became:

Noticed that as the iteration number increased, the warping image fitted slightly better because with more iteration we could acquire a better homography. So choose iteration number 1500, the next step is to tune tolerance. Results are shown below:

As observed in results, when tolerance increase, the template would have a better fit of image, since for tolerance larger than 3 the results do not differ much, I would choose a tolerance as 3.

3   Creating your Augmented Reality Application

3.1   Incorporating Video

4   Extra Credit

4.2   Create a Simple Paranoma

Created panaroma is shown below: