Assignment 2

Ashwin Rao (2019101049)

1.1: Thought experiments

1. Optical flow can be used to create slow-motion videos. This is done by synthesizing intermediate frames of the video. By estimating the flow between 2 consecutive frames of the original video, we can interpolate the optical flow for intermediate frames. This interpolated flow can be used to generate intermediate frames. Hence, we can create a slow-motion effect by synthesizing new frames that capture intermediate information (such as movement). This is done without any effect on the FPS of the video. Further, this makes the output video smoother than simply changing the FPS. Instead of changing the time between frames, we insert new frames which add more information between original frames.

2. In the video, multiple cameras are placed around the scene and the actors. The cameras capture images one after the other with some delay between each click, in a specific sequence. The resulting video can be slowed down, or made smoother, by interpolating consecutive frames as described in the previous question.

3. We can recreate the painterly effect seen in the video using optical flow. This is done by estimating the flow between frames as a 2×1 vector. We then apply a paint effect (smearing, stroking, etc) along these flow vectors.

4. When the sphere is rotated, there is no visible change in the intensity or distribution of pixels, since there is constant illumination. Hence, we do not observe any optical flow. The 2D motion field will be in the direction of motion of the sphere.

When the light source is moved, there is a visible change in illumination. Patches of lighter pixels are observed to move across the sphere. Hence, we will observe optical flow in this case. It will be in the direction opposite to the direction of motion of the illumination source.

1.2: Concept review

1. The first assumption we make is that pixels retain the same brightness or intensity between frames. Only their positions change due to the motion of the camera, observer, or objects in the scene. hence, we do not consider changes in lighting, for example. This is the **brightness constancy assumption**.

The second assumption is the **spatial coherence assumption**. We assume that neighbouring pixels of a given pixel have the same flow or shift as the pixel under consideration. This means that groups of pixels for together in a similar motion, and not just individual pixels. This is done to avoid making the problem under-constrained, giving us enough equations to solve for unknowns.

2.

The objective function is :

$$I(x, y, t) = I(x+u, y+v, t+1)$$

where :

- $(x, y) \rightarrow$ original pixel location.
- $(u, v) \rightarrow$ flow for that pixel.

Here, we assume that pixel intensity remains constant and only their locations shift (brightness constancy assumption).

Using Taylor series approximation :

$$\nabla I \, [u \;\; v]^T + I_t = 0$$

where

- $\nabla I = [I_x \;\; I_y]$ (spatial term since it involves derivative at a specific point)
- $I_t = I(t+1) - I(t)$ (data term since it is the difference in data at two times)

Assume the spatial coherence constraint and 5×5 window size. That is, suppose the flow is the same for all pixels around the pixel under consideration, in a 5×5 window.

Then,

$$A d = b$$

where:

$$A = \begin{bmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ \vdots & \vdots \\ I_{x_{25}} & I_{y_{25}} \end{bmatrix}, \quad d = \begin{bmatrix} u \\ v \end{bmatrix}, \quad b = \begin{bmatrix} I_{t_1} \\ I_{t_2} \\ \vdots \\ I_{t_{25}} \end{bmatrix}$$

Multiply with A^T on both sides to get the Lucas-Kanade equation:

$$(A^T A) d = A^T b$$

3. The first-order Taylor series approximation is done to simplify the equation, making it in terms of the derivative of the image and the flow, as separate terms. If this simplification was not done, it would not be possible to estimate the flow, since the original equation consists of a function of the flow: $I(x+u, y+v, t+1)$ rather than the flow itself: (u, v).

The Taylor approximation results in us being able to estimate the flow directly, since we obtain a linear equation which we can solve to obtain the flow. Further, its terms are easy to compute: for example, the derivatives of the first image and the difference between the two images.

The objective function before Taylor Series approximation:

$$I(x, y, t) = I(x+u, y+v, t+1)$$

where:

- $(x, y) \rightarrow$ original pixel location.
- $(u, v) \rightarrow$ flow for that pixel.

After Taylor series approximation:

$$\nabla I \, [u \;\; v]^T + I_t = 0$$

where

- $\nabla I = [I_x \;\; I_y]$ ($I_x \rightarrow$ x-derivative at $[x \;\; y]^T$
  $I_y \rightarrow$ y-derivative at $[x \;\; y]^T$)
- $I_t = I(t+1) - I(t)$

4. Geometrically, the objective function obtained after the Taylor series approximation fails because motion perpendicular to the gradient at any point cannot be captured in the equation.

If $\nabla I \, [u' \;\; v']^T = 0$, this means that $[u' \;\; v']^T$ is perpendicular to the gradient at a point.

This means that

$\nabla I \, [u \;\; v]^T + I_t = 0$ is satisfied for $[u \;\; v]^T$ as well as for $[u+u' \;\; v+v']^T$.

2.3: Analyzing Lucas-Kanade Method

1. This is because if $A^T A$ has a rank less than 2, it is not invertible since it is not full rank. This means that the Lucas-Kanade equation has no solution at these points.

The threshold τ is used on lower eigenvalues. Using τ ensures that the determinant is not very small, since if a matrix has a determinant of zero, it is not invertible. This means that solutions become less reliable as the determinant approaches zero.

2. Using larger thresholds comes with a trade-off. When using a larger threshold, only well-conditioned points like corners satisfy the threshold, and the flow

estimation is best for these points. However, due to less number of points under consideration, the overall flow estimation for the image becomes sparse.

In my implementation, the algorithm worked better for the RubberWhale sequence, while it performed relatively worse on the Grove3 sequence. In the RubberWhale sequence, the flow is more organized, less noisy and similar over larger regions. An image in this sequence can be divided into four parts based on flow, and the algorithm performs well. However, the Grove3 sequence, although having more corners and edges, is highly unorganized, with flow varying from point to point. The estimated flow is hence noisier.

3. In general, I observed that results improved with an increase in window size. This is particularly true for organized sequences like RubberWhale. When using larger windows, we have more equations to contain flow estimates for each point. The trade-off is that we require more computational power since the computations become more complicated and take longer, hence reducing the efficiency of the algorithm. Smaller windows take less time, while larger windows give better results.

4. This method fails for rotations and occlusions. If the motion of the camera or objects in the scene result in certain points, originally visible, being occluded in later frames, this method fails since it requires the positions of the points in later frames to estimate flow.

5. Ground truth visualizations are in HSV colour space. This is because the flow is represented as a 2×1 vector. This means that it can be mapped to a point on a 2D plane, specifically, the 2D colour wheel used in HSV colour representation (where, for a given saturation, the angle determines the hue and magnitude determines the value).