IIIT Hyderabad

Computer Vision (CS7.505) **Assignment 2**
Instructor: Anoop Namboodiri
Name:Yash Motwani (2020122002)

Question 1.1

1.

Optical flow can be used to create slow motion videos using interpolation. Interpolation is a type of estimation, it estimates new data points based on the range of a discrete set of known data points .We find the intermediate animation frames generated between existing ones by means of interpolation, in an attempt to make animation more fluid and for fake slow motion effects.

2.

The optical flow is used to show slow-motion effect of the bullet.Firstly, a set of cameras were arranged around the subject. The camera path was pre-designed using computer generated visualizations. Now, there were frames being introduced in between two consecutive frames i.e., interpolating the frames captured via optical flow.

3.

The painterly effect occurs due to the optical flow.Initially, we have the velocity vectors of every pixel.Now, we track the movement of the pixels and also find the motion vector of the actor. We use this information to attach the individual brushstrokes to the existing natural footage that would then move in sync with the live elements.

4.

(i)In this case,there wont be optical any optical flow since no local image motion can be measured but motion field will exist.
(ii)In this case, there will be optical flow pointing in the direction where the illumination is moving, but there won't be any motion field.

Question 1.2

1.

Assumption 1:The apparent intensity value at a physical point does not change across the image sequence –brightness constancy assumption.
Assumption 2:Across consecutive frames, every point undergoes only a small displacement.

2.

I(x,y,t) is the data term and I(x+u,y+v,t+1) is the spatial term.
Using brightness constancy constraint

$$\text{I(x,y,t)=I(x+}\triangle\text{x,y+}\triangle\text{y,t+}\triangle\text{t)}$$

$$\implies I(x,y,t) = I(x,y,t) + \triangle\text{x}\ \frac{\partial I(x,y,t)}{\partial x} + \triangle\text{y}\ \frac{\partial I(x,y,t)}{\partial y} + \triangle\text{t}\ \frac{\partial I(x,y,t)}{\partial t} \ \text{.....[From assumption 2]}$$

$$\implies 0 = \frac{\triangle x}{\triangle t}\frac{\partial I(x,y,t)}{\partial x} + \frac{\triangle y}{\triangle t}\frac{\partial I(x,y,t)}{\partial y} + \frac{\partial I(x,y,t)}{\partial t}$$

$$\implies u(x,y,t)\frac{\partial I(x,y,t)}{\partial x} + v(x,y,t)\frac{\partial I(x,y,t)}{\partial y} + \frac{\partial I(x,y,t)}{\partial t} = 0 \tag{1}$$

Equation 1 is called as Brightness constancy equation.It says that only the component of the optical flow along the direction of the gradient can be determined.The component along the edge direction or the component perpendicular to the gradient cannot be determined.
We notice that equation 1 has 2 unknowns.So we have 2 unknowns but only 1 equation per pixel.This problem is Aperture Problem.It is the motion of an edge as seen through an aperture is essentially ambiguous.

Lucas-Kanade(Solution of aperture problem):
The basic idea is to impose the additional constraints.The assumption is that the flow field is smooth locally i.e., the optical flow is constant in a small window.Then we use least squares approach to solve for the optical flow by combining together several brightness constancy equations.
Ex: If we use 5x5 window, we get 25 equations per pixel.

Let I_x=$\frac{\partial I}{\partial x}$
I_y=$\frac{\partial I}{\partial y}$ and
I_t=$\frac{\partial I}{\partial t}$

Consider a NxN window. From equation (1), we get:

$$\begin{pmatrix} I_{x_1} & I_{y_1} \\ I_{x_2} & I_{y_2} \\ I_{x_3} & I_{y_3} \\ . & \\ . & \\ I_{x_{N^2}} & I_{y_{N^2}} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} I_{t_1} \\ I_{t_2} \\ I_{t_3} \\ . \\ . \\ I_{t_{N^2}} \end{pmatrix}$$

$$\implies A.d = b$$

The goal is to minimise $||Ad - b||^2$. To minimise , we use least squares problem.

$$(A^TA)d{=}A^Tb$$

$$\implies \begin{bmatrix} \sum I_xI_x & \sum I_xI_y \\ \sum I_xI_y & \sum I_yI_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_xI_t \\ \sum I_xI_t \end{bmatrix}$$

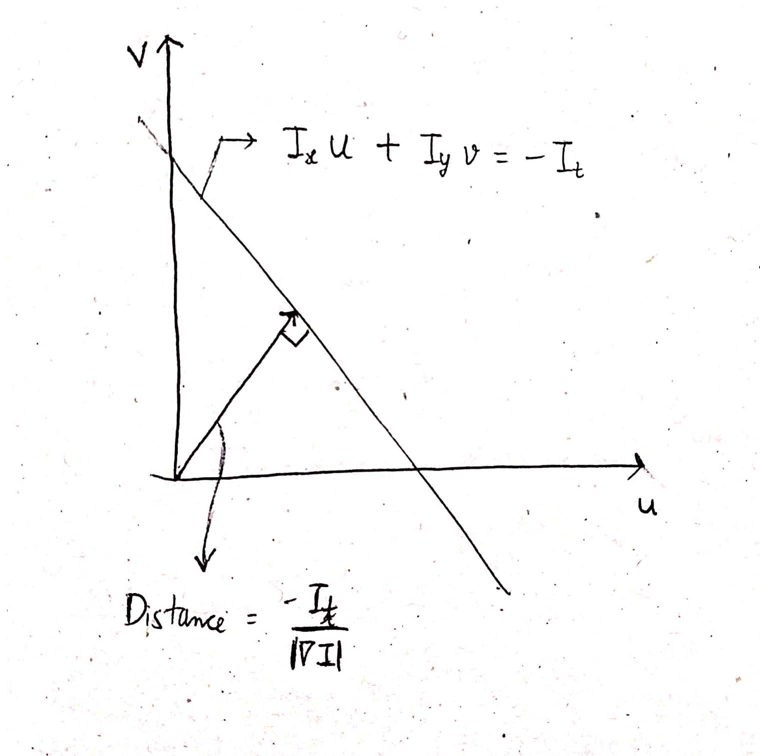
Optimal (u,v) satisfies the above equation.

3.

The Taylor series provides an approximation or series expansion for a function. This is useful to evaluate numerically certain functions which don't have a simple formula.Regarding why first order instead of higher order, the answer in this case is most likely the same as the answer in every other application of Taylor polynomials: In order to minimize computation time, use the smallest order Taylor polynomial for which the approximation error is acceptably small.

4.

$$\nabla I.\begin{bmatrix} u & v \end{bmatrix} = -I_t$$
$$\implies I_xu + I_yv + I_t = 0$$

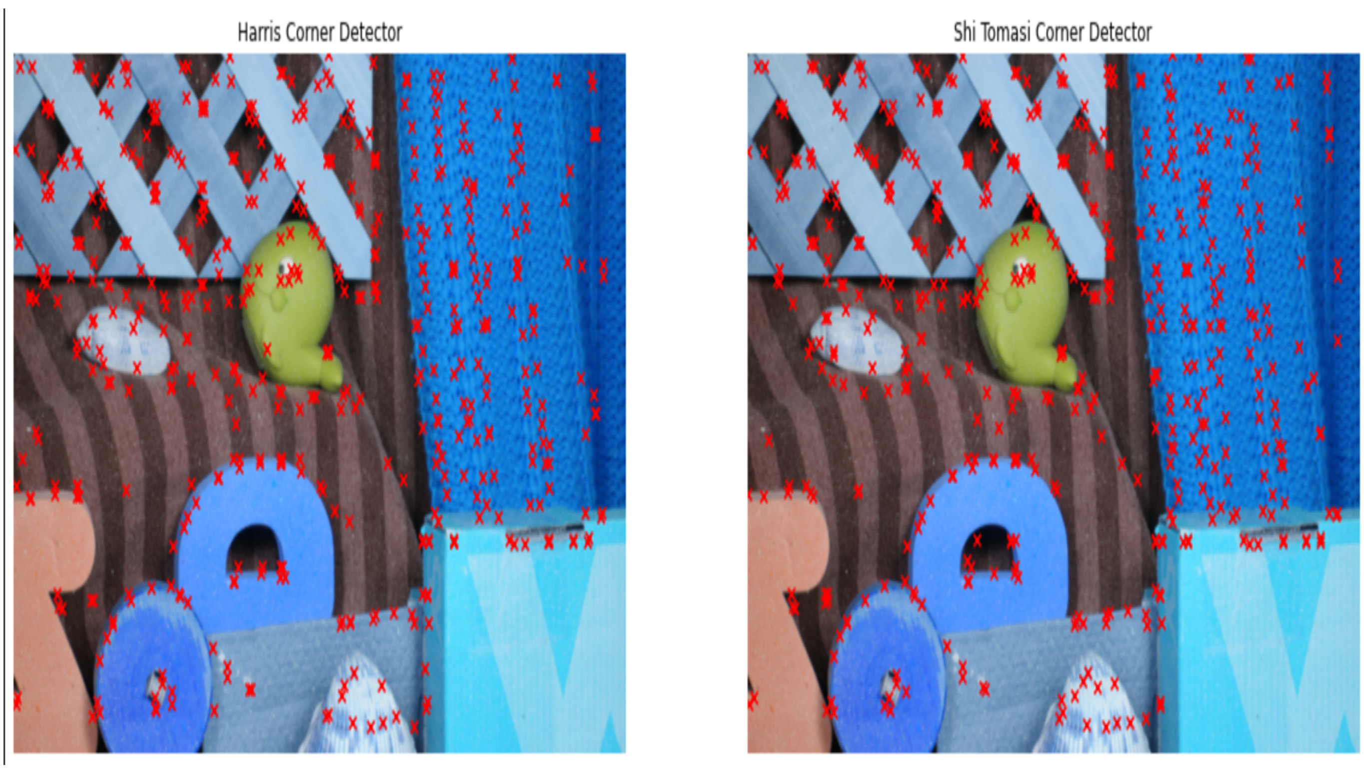


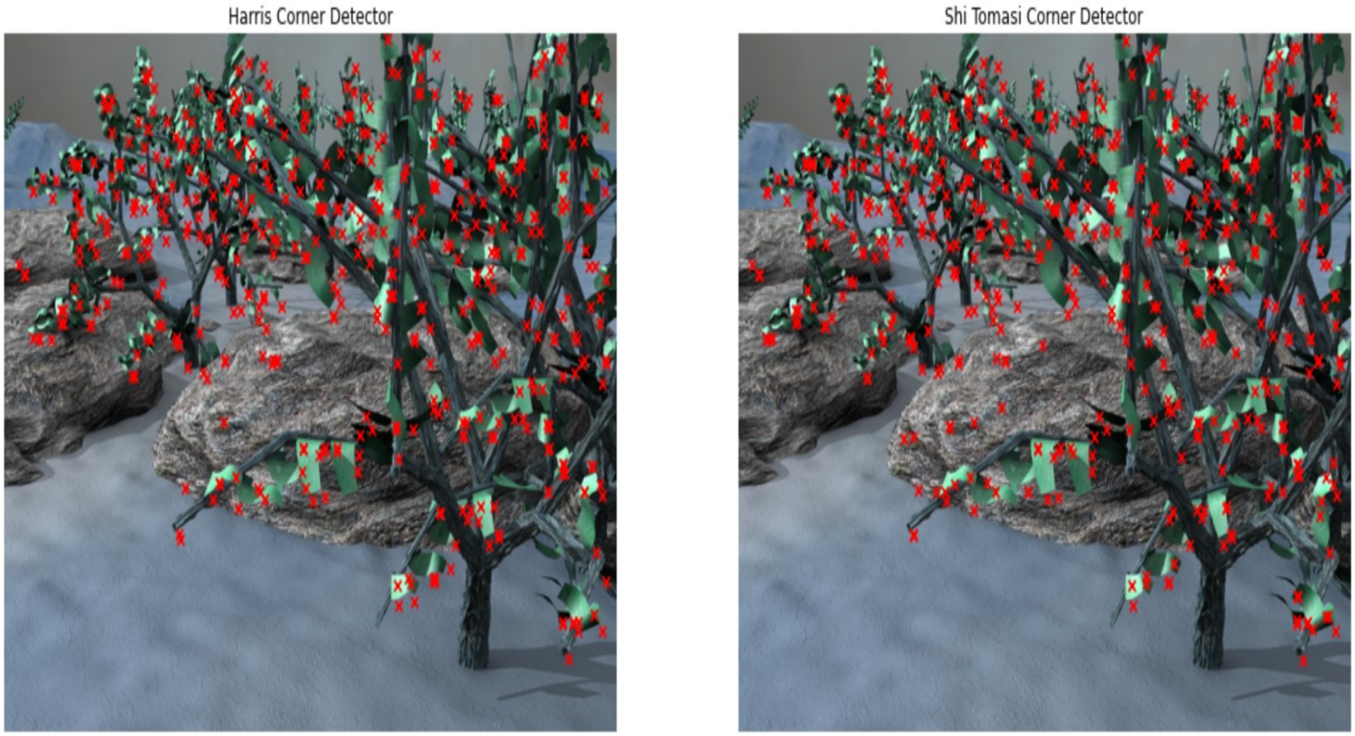
where the intensity gradient, ∇I and I_t is measurable and image flow (u, v) needs to be derived. The intensity gradient, I, always points in the direction of maximum increase of intensity, while the vector (u , v) lies somewhere along a line perpendicular to the intensity gradient. Since there are two unknowns in one equation, solutions to optical flow constraint equation is non-unique and problem is ill posed.

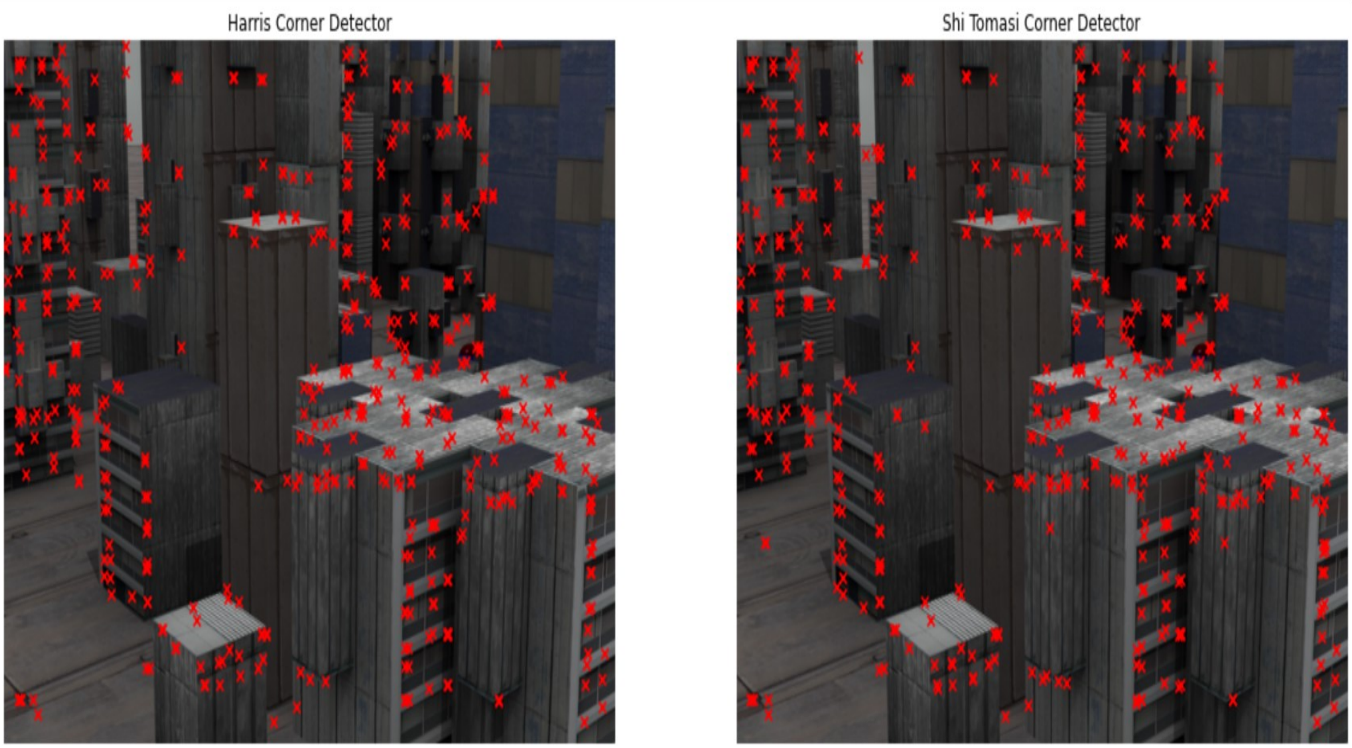
Question 2.1

(i)Harris corner detector algorithm is:
1. Color to grayscale
2. Spatial derivative calculation
3. Structure tensor setup
4. Harris response calculation
5. Non-maximum suppression .

(ii)Shi-Tomasi corner detector algorithm is:
1. We consider a small window on the image then scan the whole image, looking for corners.
2. Shifting this small window in any direction would result in a large change in appearance, if that particular window happens to be located on a corner.
3. Flat regions will have no change in any direction.
4. If there's an edge, then there will be no major change along the edge direction.

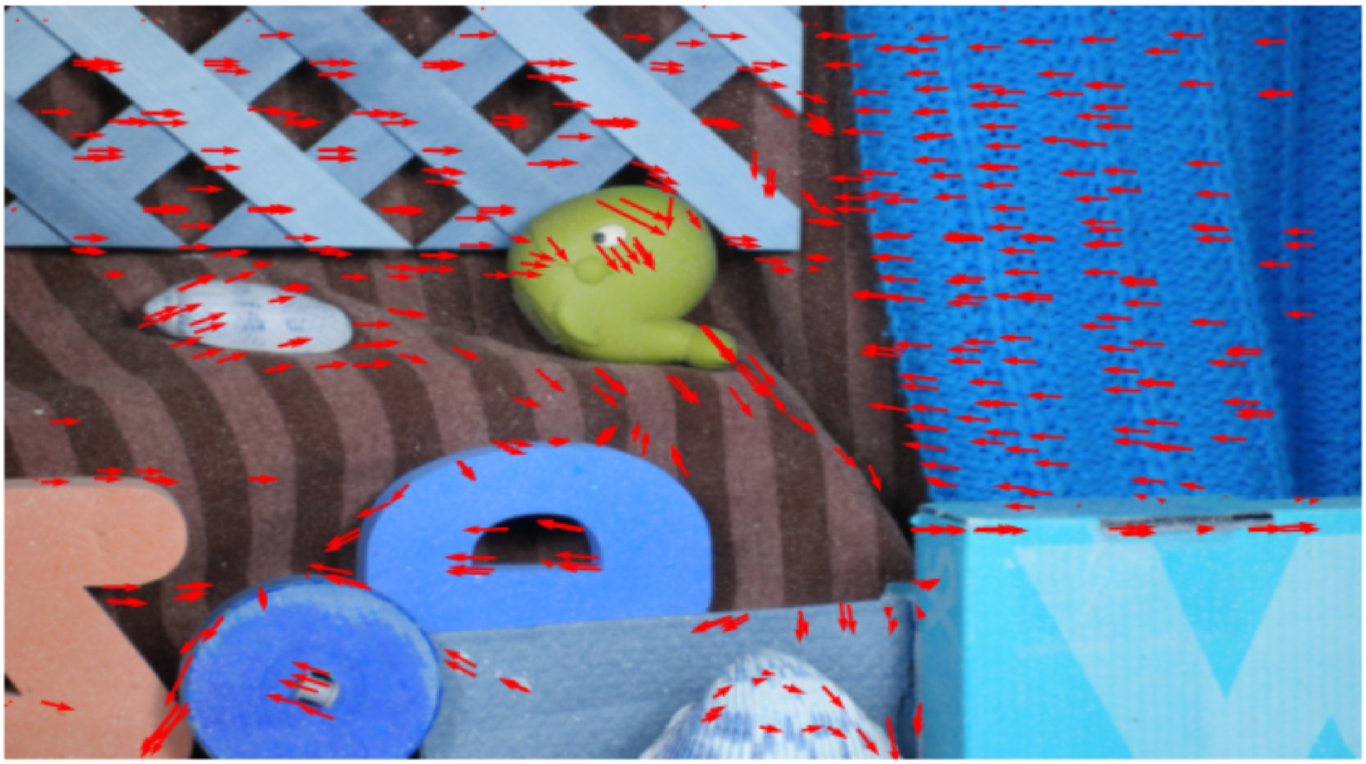


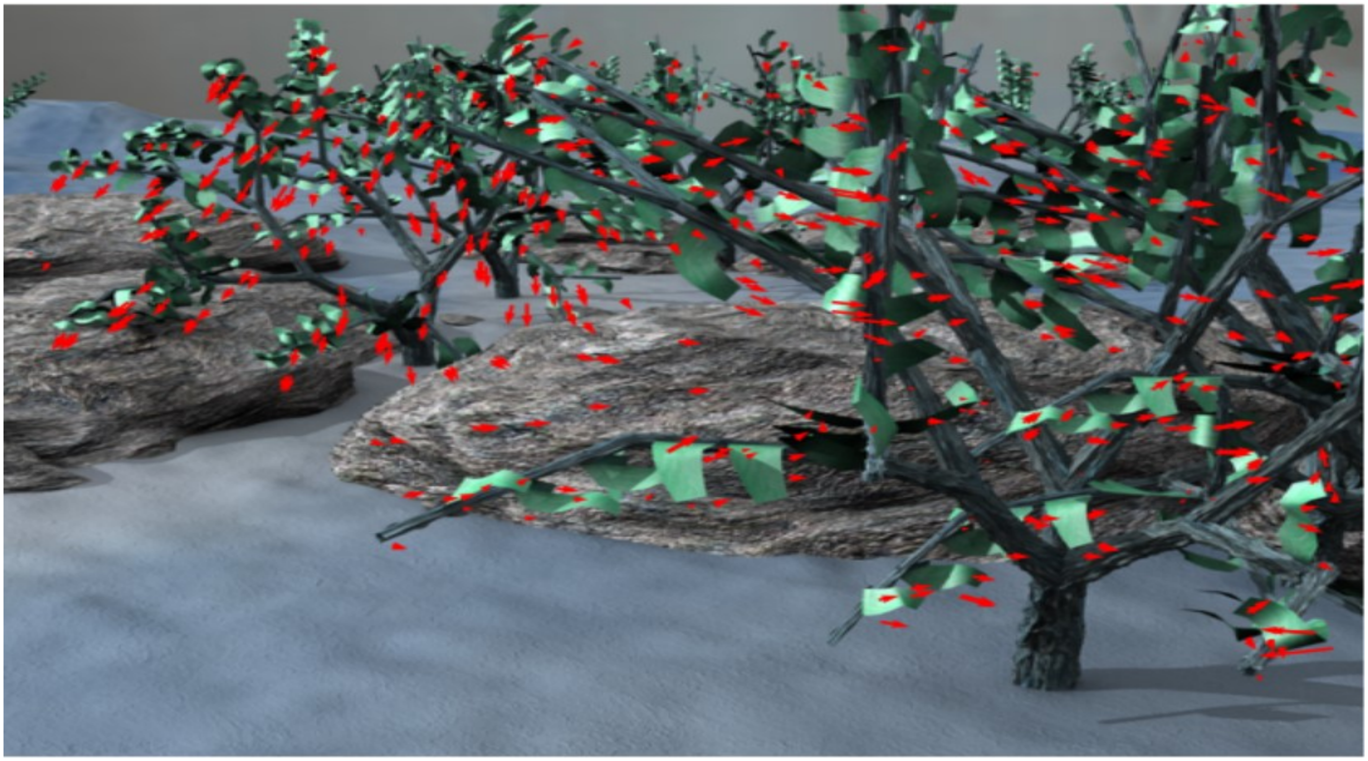

Harris Corner Detector
Shi Tomasi Corner Detector

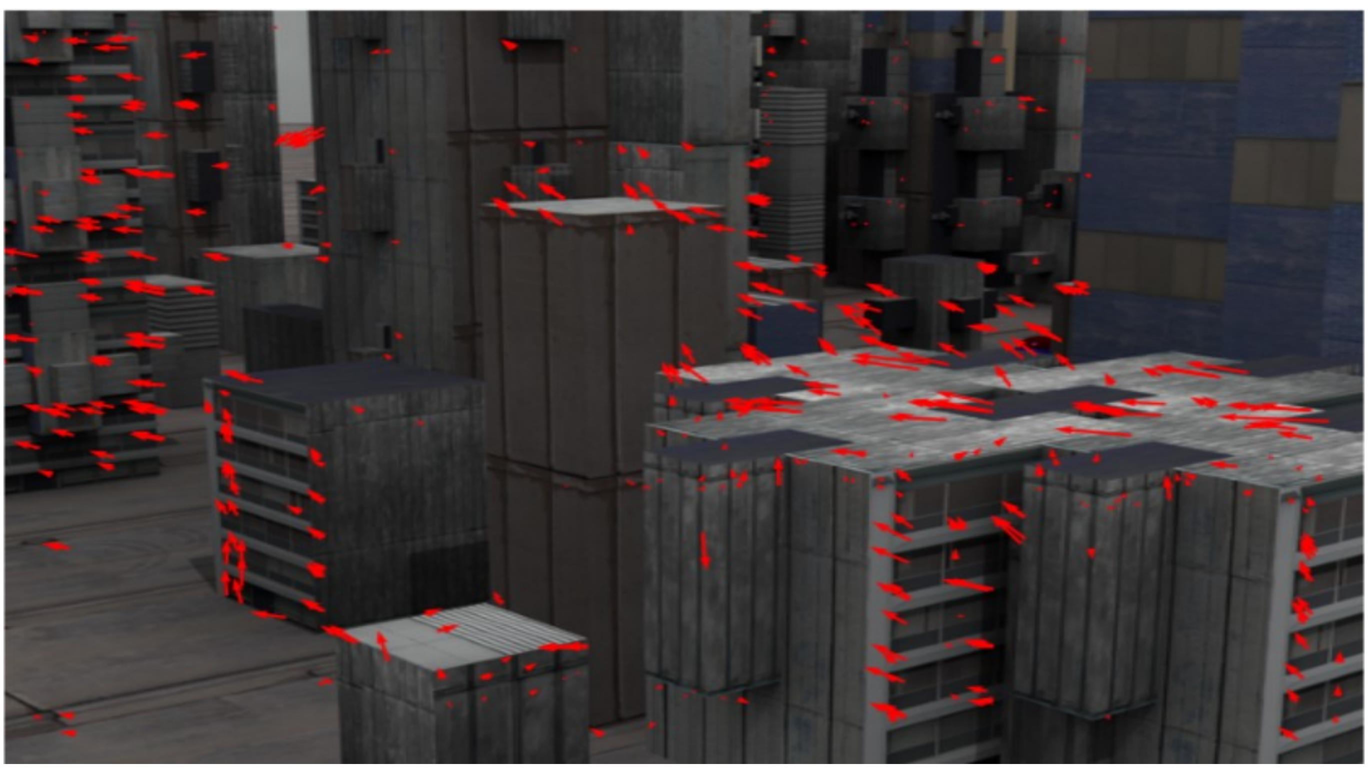

Harris Corner Detector
Shi Tomasi Corner Detector

Question 2.2

The main idea of this method based on a local motion constancy assumption, where nearby pixels have the same displacement direction. This assumption helps to get the approximated solution for the equation with two variables. Let us assume that the neighboring pixels have the same motion vector $(\Delta x, \Delta y)$. We can take a fixed-size window to create a system of equations. Let $p_i = (x_i, y_i)$ be the pixel coordinates in the chosen window of N elements .



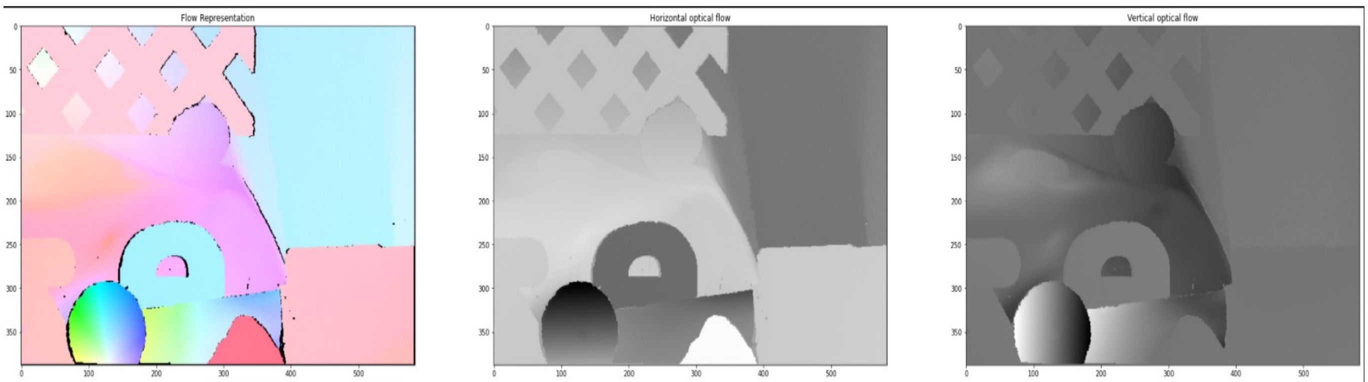


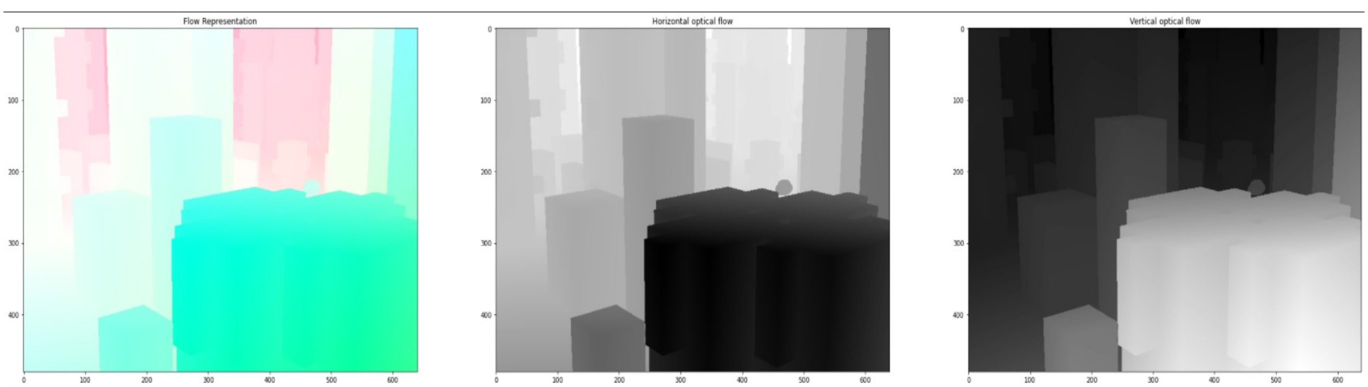


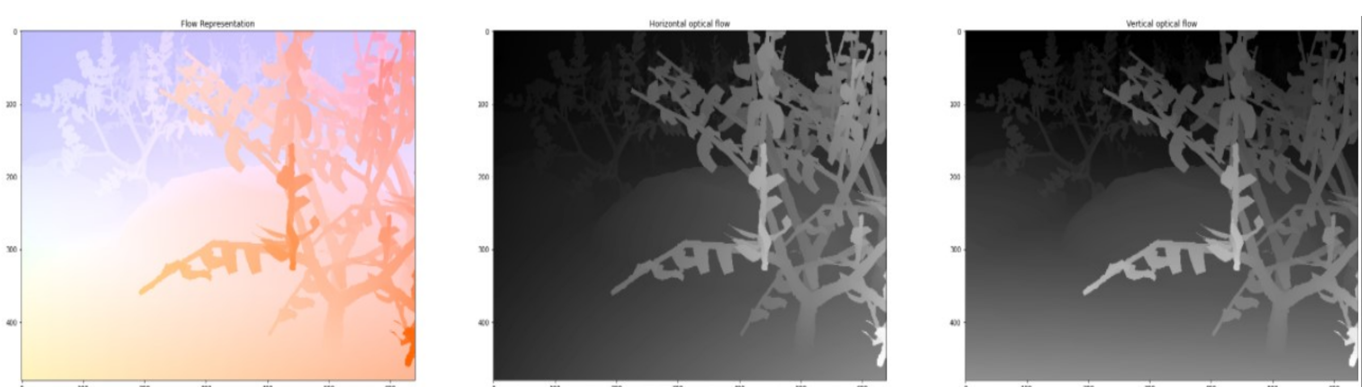
The EPE for different datasets for a particular value of window size for shi-tomasi corner detector and Lucas Kanade method is given as:

Rubber Whale:36.815
Urban2:152.076
Grove3:62.91

The horizontal and vertical flows are given as:







Question 2.3

1.

When the rank of A^TA is 2, then we have two equations and two unknowns. Hence, a single unique solution can be found.If we increase the rank then we can only obtain a least-squares solution which won't be accurate. The threshold (TAU) determines the range till which the eigenvalues are allowed. If the threshold is small, then the rank of matrix can be low. If it is high, then the matrix is close to full rank.

2.

I have used a threshold tau . If the eigen values exceed the threshold tau, those points will be considered.

3.

The algorithm is not working well for Urban. The average end point error seems very high. One of the assumptions of Lucas Kanade is that neighboring pixels have the same displacement. However, this is violated as the buildings on the right side move much more compared to those on the left . Also, Urban has much lesser contrast compared to the rest.

4.

The image above is an example of optical flow failure. This is because Ego motion rotations normally result in homogeneous and uniform image stream flows, whereas independent rotations (bottom) commonly result in the Lucas Kanade assumption of local constant velocity not holding. Another reason for failure is occlusion's reliance on the relationship between picture sampling frequency and occlusion.

Question 3

The main idea of this method is to approximate some neighbors of each pixel with a polynomial. In the sparse Lucas Kanade method, we used the linear approximation since we had only first-order Taylor's expansion. Now, we are increasing the accuracy of approximation with second-order values. Here, the idea leads to observing the differences in the approximated polynomials caused by object displacements. Our goal here is to calculate a displacement d in the equation using the polynomial approximation.