1.1

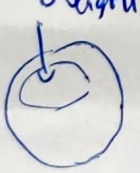
1. Consider a movie at 24 fps now if we reduce the fps by half i.e make it 12. we have got a slow motion movie albeit without smoothness. Then we use optical flow to convert this movie back to 24 fps and we can do so on for creating slow motion footage take a lot of frames in a split second and spred it out using optical flow.

2. 

Consider the following setup we have cameras at regular angular distance surrounding the object we then take calculated pictures using these cameras at an offset. We then stich together these frames using optical flow to create a full movie. This is how the bullet time scene was shot. The actor just fall down and cameras took multiple photos at diffrent angles and these photos we stiched using optical flow.

3. The generation of painting effects was done in the following way. The One can generate a paint stroke for every pixel and can transform the paint stroke to the next image and to the next image after that. One can use the optical flow calculation to compute the transformation needed.

4.) The lambertian ball is completely smooth therefore a ~~moving~~ rotating ball in a constant global Illumination would not produce any optical flow. However a moving light would cause the shadows and lighting to change on the surface of the lambertian ball thus producing non-zero optical flow



2.3

1. A^TA is a 2×2 matrix. Therefore we can write the equation for least squares as

$$V = (A^T A)^{-1} A^T b$$

for A^TA to be invertible it should have rank 2 then only the above equation would be solvable.

The threshold tau is checking this only if τ is large than the smallest eigen value of A^TA then the flow should not be computed. Keeping τ to be a small quantity we can eliminate responses whose eigen value is too close to zero i.e rank < 2.

2. In the ipynb itself. See report.

3. With a small window size the algorithm captures subtle motions but not large motions. With large size it happens the other way.

(for output see ipynb) This is due to the fact that the motion is written as

$$I_1(x,y) = I_2(x+u, y+v)$$

This is due to this implicit assumption that the motion is small. If the motion is large this assumption is violated.

4.

(1) It cannot provide flow information in uniform regions of the image- like take lambertian ball example

(2) It can also happen the object gets occluded in the next frame. ~~Then~~ Here Lucas Kanade would not be able to ~~go~~ provide accurate results.

5. Not just flow HSV can be used to color code any 2D vector field. because HSV has a cyclic colormap therefore any angle and a radius corresponds to a unique color this allows us to map vectors in 2D space to unique color values depending upon their magnitude and direction

1.2.

1: 1) Nearby objects in the image plane move in a similar manner. (Spatial coherence) Since they usually belong to same surface.

2) The Observed brightness of any object point is constant over time. (~~Temporal persistance~~) (Brightness constancy)

3. Temporal Persistance: The image motion of surface patch changes gradually over time.

2: The brightness constancy constrait is given as

$$I(x,y,t) = I(x+\Delta x,\ y+\Delta y,\ t+\Delta t)$$

$$I(x+\Delta x,\ y+\Delta y,\ t+\Delta t) = I(x,y,t) + \frac{dI}{dx}\Delta x + \frac{dI}{dy}\Delta y + \frac{dI}{dt}\Delta t$$

$$\frac{dI}{dx}\Delta x + \frac{dI}{dy}\Delta y + \frac{dI}{dt}\Delta t = 0$$

divide by Δt

$$\frac{dI}{dx}\frac{\Delta x}{\Delta t} + \frac{dI}{dy}\frac{\Delta y}{\Delta t} + \frac{dI}{dt} = 0$$

$$I_x V_x + I_y V_y = -I_t$$

V_x & V_y are x & y components of velo

$$\therefore \nabla I \cdot \vec{V} = -I_t$$

∇I: Data term, $\vec{V}$: Spatial term.

3- It is done to T linearization. The accuracy of this
do a
approximation depends on displacement magnitude and
Since we assume to have low displacement b/w two
frames this optimization holds.