Name: Balaji Chundi
SR No: 19728

E9 208 Digital Video: Perception and Algorithms

Assignment 1

Abstract

This assignment is about extrapolating the video frames using optical flow. For the estimation of forward optical flow I have used two methods, namely, Lucas-Kanade algorithm and Discrete Horn-Schunck algorihtm. These algorithms have been compared in terms of quality of the extrapolated frame, where the motion is assumed to be linear. I have extrapolated video frames using the method proposed in [1].
Datasets used for this analysis are Corridor and Sphere. The hyperparameters of both these algorithms have been optimized for the given task.

Approach

The equation to be solved for u and v is shown below,

$$I_x u + I_y v + I_t = 0$$

where u and v represent the velocity of x and y respectively and I_x, I_y and I_t are partial derivatives of pixel function of coordinates x and y.
So, for the two unknowns we only have a single equation, this cannot be solved. The algorithms Lucas-Kanade and Discrete Horn-Schunck are two of the many methods that help us solve the above equation for u and v.

Lucas-Kanade Algorithm

Standard algorithm from openCV library, *calcOpticalFlowPyrLK* is used to calculate flow of the desired points. The desired points were also calculated using the *goodFeaturesToTrack* function from openCV. Parameter *maxLevel* is kept 0, a constant value, as we are not considering multiscale Lucas-Kanade optical flow algorithm. Criteria for the optical flow is also considered to be constant.

The hyperparameter tuned is *window size* for both the video frames. A range of window sizes from 11 to 21 are considered and the best in terms of performance for both the video frames are:

- For Sphere: (12, 12)
- For Corridor: (13, 13)

After the calculation of the forward flow from images and time t and t+1, I tried to extrapolate the next frame using the frame t+1 and the estimated forward optical flow from frame at time t and frame at time t+1.
Now by multiplying some coefficient k to the motion vector, we can get following equation:

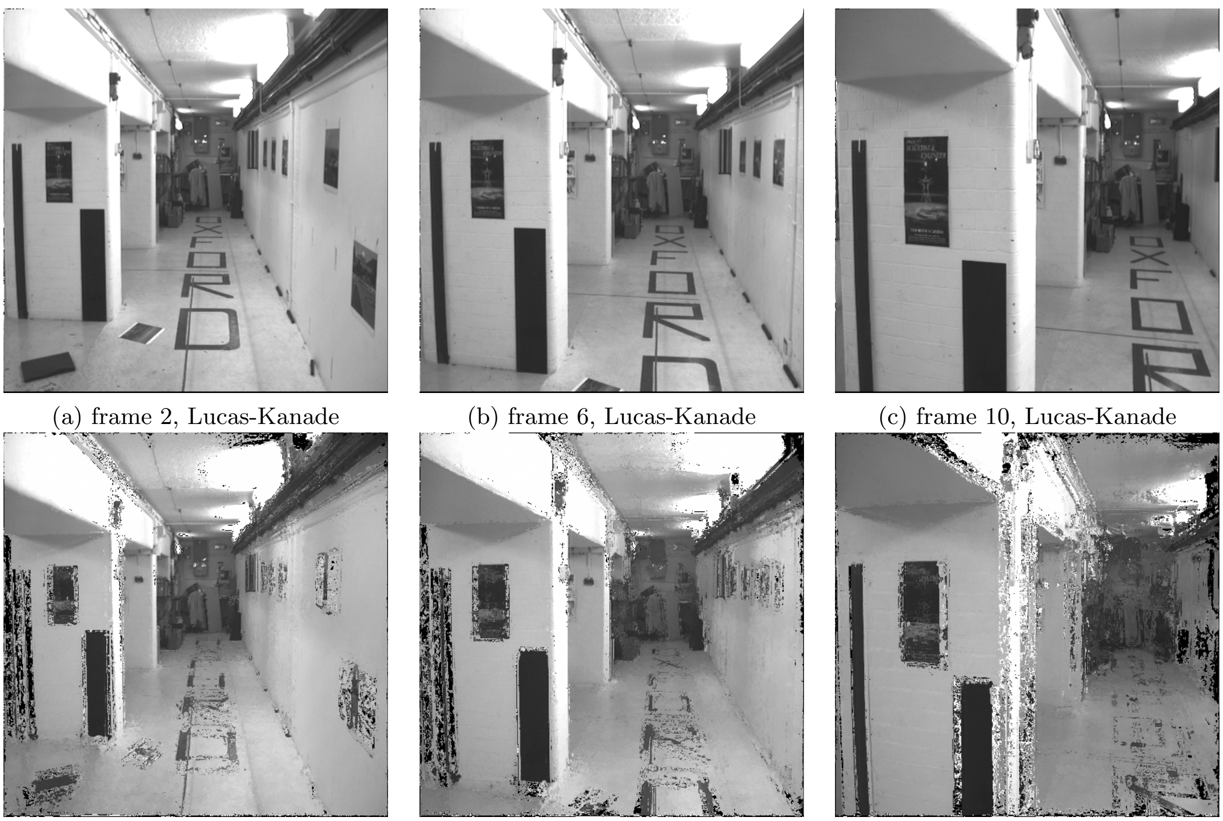
$$I_{new}(x_0 + ku, y_0 + kv, t_0 + k\delta t) = I_0(x_0, y_0, t_0)$$

as we are extrapolating the value of k, should be greater than 1 ($k > 1$). I have tried different values for k and found that $k = 2$ works better for Corridor frames. and k = 4 works better for Sphere frames. Intuitively, the k coefficient the position of new frame.
To estimate the performance of these algoithms in predicting future frames, I have used simple mean-square error as the metric.

Discrete Horn-Schunck Algorithm

The hyperparameter tuned in the case of Discrete Horn-Schunck algorithm is 'alpha', the relative weight of the cost function or simply put, regularization parameter. A range of values 0.001, 0.01, 0.1, 0.5, 1, 0.75, 1.5 for alpha are considered. For both the video frames $\alpha = 1.5$ turned out to be the best. The number of iterations were limited to 100 in both the cases too, because not much development has been observed in the extrapolated frames even if I increase the iterations to 300.

1 Question 1



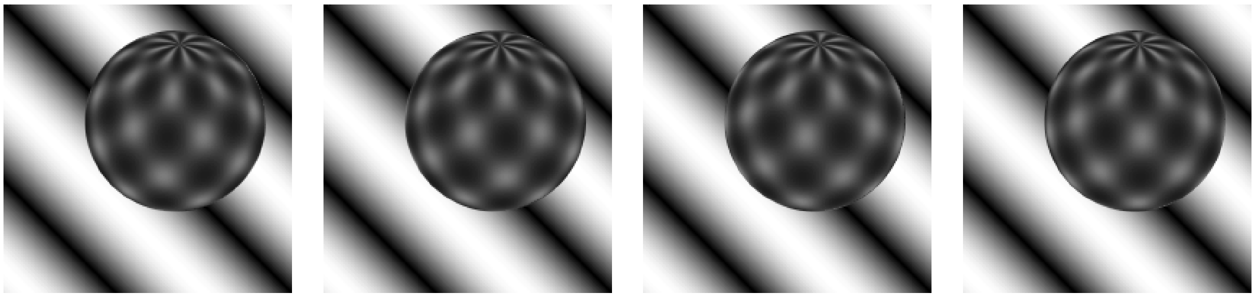
(a) frame 2, Lucas-Kanade (b) frame 6, Lucas-Kanade (c) frame 10, Lucas-Kanade

(d) frame 2, Horn Schunck (e) frame 6, Horn Schunck (f) frame 10, Horn Schunck

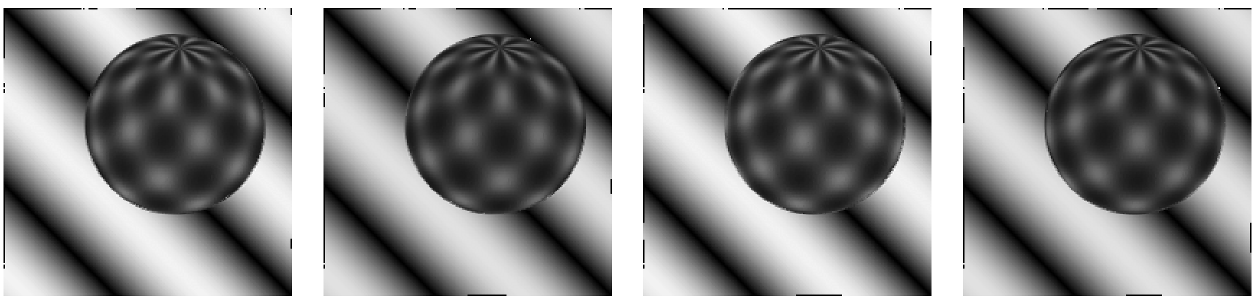
Figure 1: Predicted Frames for Corridor Video Frames

- Visually for both the datasets, Lucas-Kanade has performed better rather than Discrete Horn-Schunck Algorithm
- No significant difference in visual quality of the extrapolated frames can be observed with Lucas-Kanade but this isn't the case with Discrete Horn-Schunck

- In comparision with the Sphere frames, the quality of the extrapolated images of Corridor frames is really very poor.
- Whereas the quality of the extrapolated frames of sphere frames is very much comparable to the original frames and is much better than that of corridor dataset.



(a) frame 2, Lucas-Kanade (b) frame 8, Lucas-Kanade (c) frame 13, Lucas-Kanade (d) frame 18, Lucas-Kanade



(e) frame 2, Horn Schunck (f) frame 8, Horn Schunck (g) frame 13, Horn Schunck (h) frame 18, Horn Schunck

Figure 2: Predicted Frames for Sphere Video Frames

2 Question 2

For numerical comparisions, I have considered a Mean-Square Error as the metric. **Maximum of MSE** For every extrapolated frame, I have calculated the MSE with the corresponding ground truth frame. The maximum of all these errors is considered as the deviation, that is, Maximum of Mean-Square Error.

- Maximum MSE for Lucas-Kanade on Sphere Frames when k = 4 and with an window of size (12, 12) is **11.037225**
- Maximum MSE for Discrete Horn-Schunck on Sphere Frames with $\alpha = 1.5$ is **21.774225**
- Maximum MSE for Lucas-Kanade on Corridor Frames when k = 2 and with an window of size (13, 13) is **80.0526**
- Maximum MSE for Discrete Horn-Schunck on Corridor Frames with $\alpha = 1.5$ is **82.7438**

Clearly, for both the datasets, Lucas-Kanade Method has performed better. Compared to Corridor Dataset, where both algorihtms failed significatly, Sphere dataset results are far more acceptable(although Lucas-Kanade's are better).

3 Question 3

- For Lucas-Kanade, as we consider image patches here patch is of size (12, 12) or (13, 13), most of the pixels show no relative flow. **The main reason for this is Lucas-Kanade tries to estimate the local optical flow by an assumption that all the neighbourhood pixels have the same optical flow and moreover that flow is linear**.
- Conversely, Discrete Horn Schunck is a global optical flow calculator. It has non-zero values at every pixel.
- To summarize, **both the methods differ in smooth regions as Lucas-Kanade gives 0 optical flow vectors in these regions but Discrete Horn-Schunck gives non-zero values at these locations**.

4 Question 4

- Pixels at which the image gradients evaluated to zero are the regions where I couldn't determine the intensities.
- Clearly, these are the regions with smooth neighbourhood where intensity values are equal. The pixel gradients result in zero value, in turn, the calculation of intensities in the extrapolated frame is not feasible.

5 Conclusion

- In Horn-Schunck algorithm, we optimize the function based on residuals from the brightness constancy. Here we improve the global smoothness but to reduce the mathematical complexity, we go for iterative method which is a time taking process. We used $= 1.5$, it produced not so better results in most of the cases.
- In Lucas-Kanade method we have used image patches and windowing methods with least squares technique.
- When compared to Horn-Schunck algorithm, Lucas-Kanade algorithm reduces noise giving more accurate and relatively better looking frames.
- The performance of Discrete Horn-Schunck may be improved by spatio-temporal presmoothing.

6 References

[1] Lyasheva, S Rakhmankulov, R Shleymovich, M. (2020). Frame interpolation in video stream using optical flow methods. Journal of Physics: Conference Series. 1488. 012024. 10.1088/1742-6596/1488/1/012024.

7 Note

Instead of considering AAE (or) PSNR as the performance comparison metric, I chose Mean-Square Error because we are dealing with the construction of a synthetic frame for which we already have a ground-truth. So, considering MSE makes more sense to me for this particular task.