E9 208 Digital Video: Perception and Algorithms
Assignment 1

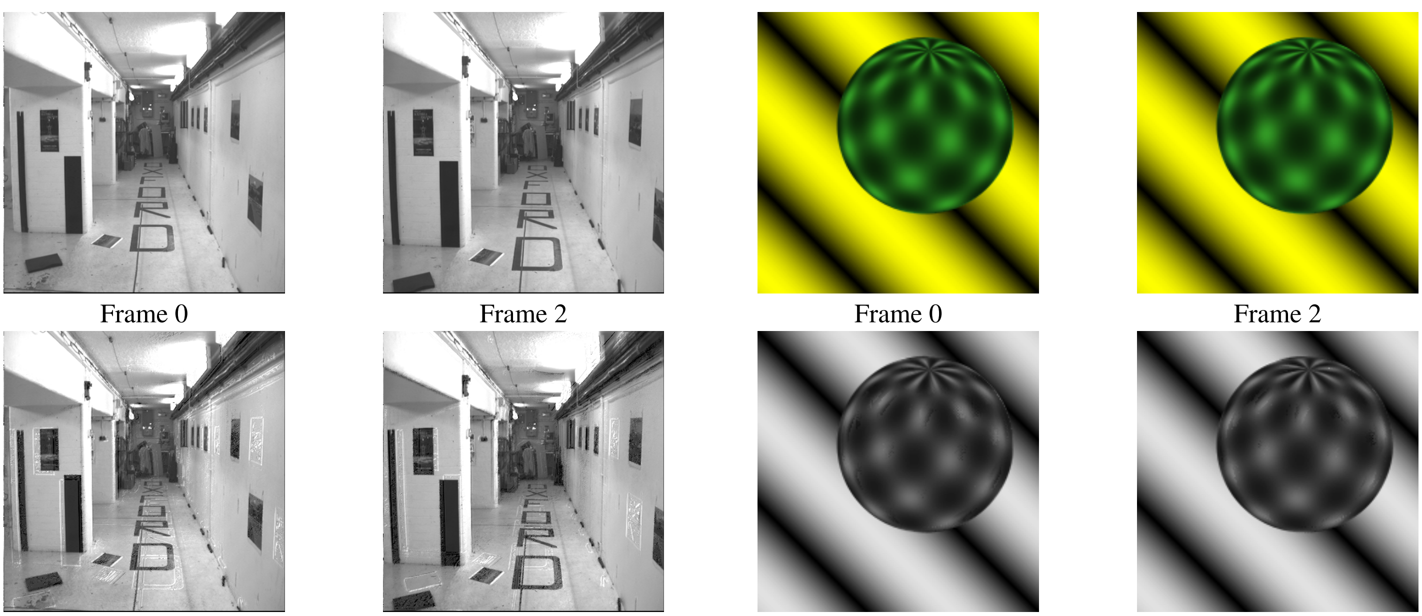
Vineeth S
M. Tech Artificial Intelligence
SR. No. 16543

Abstract

In this assignment we explore the idea of interpolating video frames using optical flow on Corridor dataset and Sphere dataset. To estimate the optical flow we use three popular methods — Lucas-Kanade algorithm, Multiscale Lucas-Kanade algorithm (with iterative tuning) and Discrete Horn-Schunk algorithm. We interpolate the video frames by two methods — by overlapping the predicted forward image & backward image at an intermediate time instant and with a method similar to one proposed in [1]. In all the methods, we observe that the quality of interpolated frames is better with the Sphere data compared to that of Corridor data. Documented code will also be available at `https://github.com/vineeths96/Video-Interpolation-using-Optical-Flow`.

Problem 1: Lucas Kanade Optical Flow Algorithm

In this section, we discuss about the performance of frame interpolation based on Lucas-Kanade algorithm. The standard algorithm is implemented along with a small modification. When we solve for the least squares solution for $Ax = b$, we add an additional constraint that we find the solution to the linear systems only if the smallest eigenvalues of $A^T A$ is greater than a threshold τ. This was introduced so that there are no large values (noise) for the optical flow and showed better empirical performance.



Frame 0 | Frame 2 | Frame 0 | Frame 2

Frame 0 Predicted from Frame 2 | Frame 2 Predicted from Frame 0 | Frame 0 Predicted from Frame 2 | Frame 2 Predicted from Frame 0

Figure 1: Optical flow frame prediction

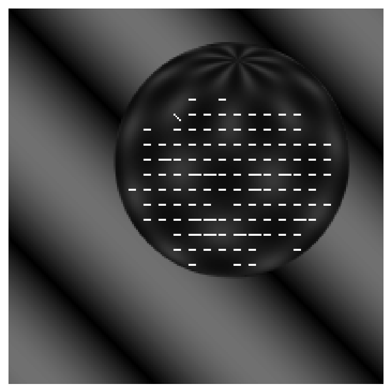
Corridor In Figure 1, we can see that we are able to reconstruct an frame back with a base frame and the calculated optical flow (with $N = 5$). This indeed confirms that we the optical flow calculated are reasonably good. We can also note that there is an error of reconstruction at the solid edges such as rectangular boundaries. This is due to the aperture problem where at the edges, where we have a single gradient direction. One possible way to overcome this issue is to smudge the edges so they are no longer solid, using a filters such as Gaussian Blurring filter. Using filters might improve the quality of optical flow estimation.

Sphere In Figure 1, we can see that we are able to reconstruct an frame back with a base frame and the calculated optical flow (with $N = 5$). The motion of the sphere is slow and have no occlusions, which are the ideal conditions for the luminous constancy assumption and hence for the Lucas-Kanade Algorithm. We can also notice that there are no significant error in the reconstruction, unlike the corridor dataset.

Interpolation Next, we explore the primary motive of frame interpolation. We calculate the forward optical flow from Frame n to Frame $n+2$ and the backward optical flow from Frame $n+2$ to Frame n. We use them both to interpolate the intermediate Frame n. We explore this method for different block sizes ($N \times N$), for $N = 5, 8, 11, 14, 20$.

Figure 3 shows the interpolated intermediate frames for the corridor dataset using Lucas-Kanade for different values of N. Figure 4 shows the interpolated intermediate frames for the spheres dataset using Lucas-Kanade for different values of N. The frames for $N = 20$ is not shown due to space constraints. Only first nine interpolated frames for the sphere data is shown. From the figure, we can observe that there is no significant difference in the quality of reconstructed frames with variation in N. The quality of the interpolated images in the corridor data is really poor. This could be due to the fact that the motion is fast and pixels are moving out of/moving into the frame. The quality of interpolated images for the sphere dataset is comparable to original frame and is much better compared to that of corridor dataset. Again this can be attributed to the slow motion and no occlusions.

(a) Corridor



(b) Sphere

Figure 2: Sample of estimated optical flows

Figure 2 shows the estimated optical flow for select few pixels in a random frame for the corridor and sphere data.

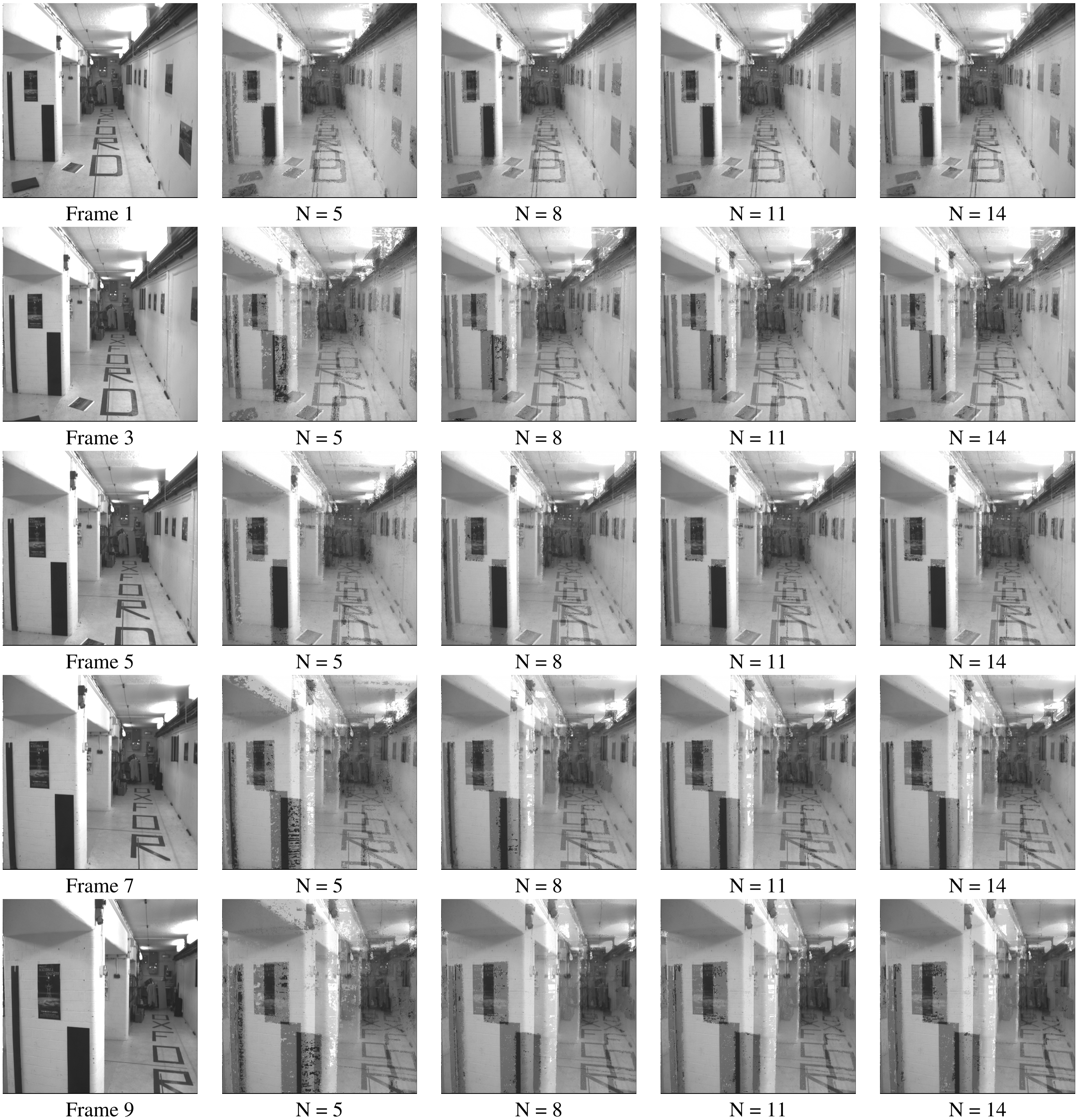


Figure 3: Lucas-Kanade Optical flow frame interpolation: Corridor

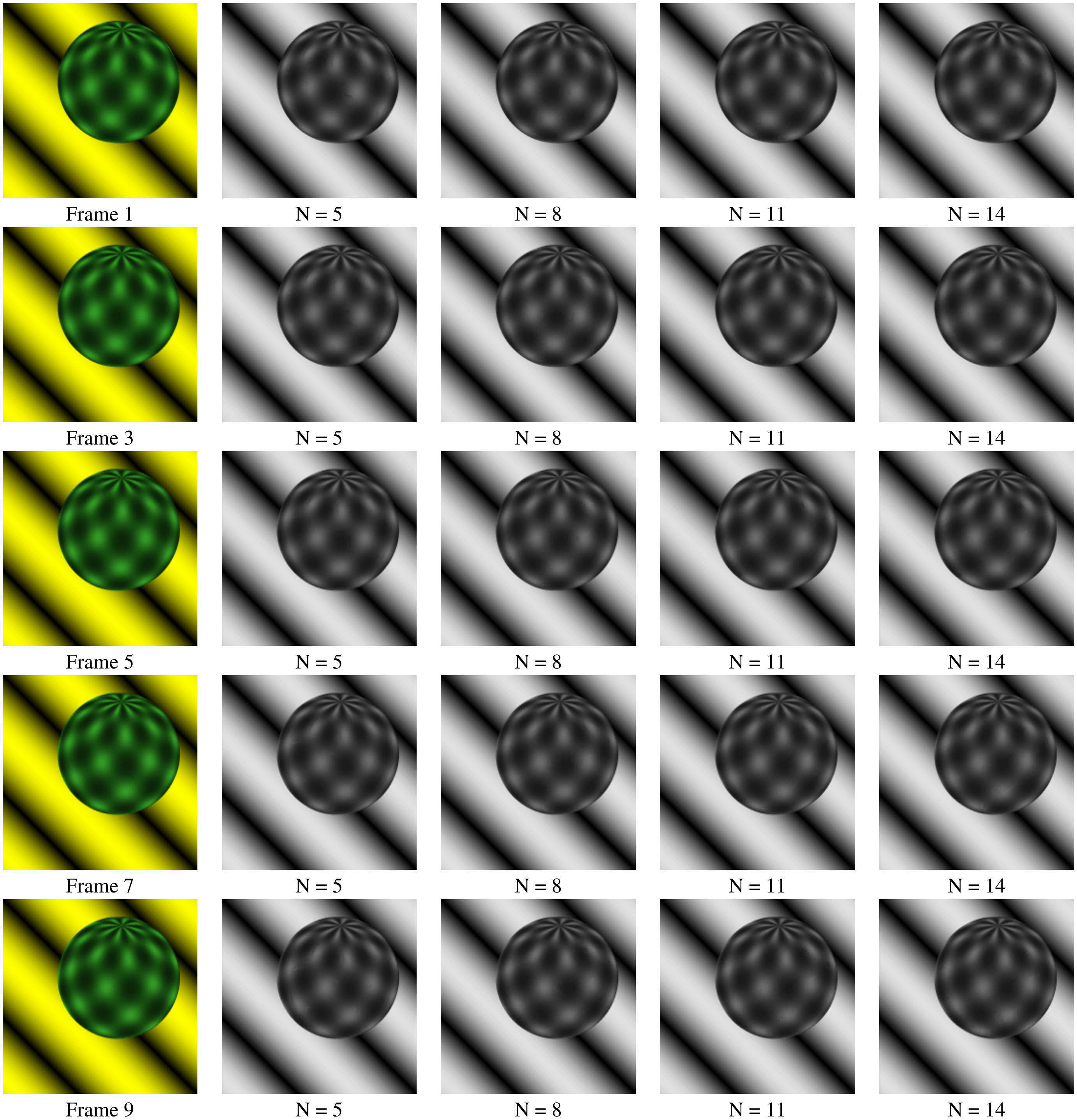


Figure 4: Lucas-Kanade Optical flow frame interpolation: Sphere

Problem 2: Multiscale Lucas Kanade Optical Flow Algorithm

In this section, we discuss about the performance of frame interpolation based on Multiscale Lucas-Kanade algorithm. We implement the algorithm and create image pyramids. At the lowest level (where image size is smallest), we estimate the optical flow using iterative Lucas-Kanade algorithm. When we move up to the next level of the pyramid, we scale the optical flow and then use iterative Lucas-Kanade algorithm to obtain the optical flow at the next level. We continue use the modification proposed in Problem 1 in Lucas-Kanade algorithm.

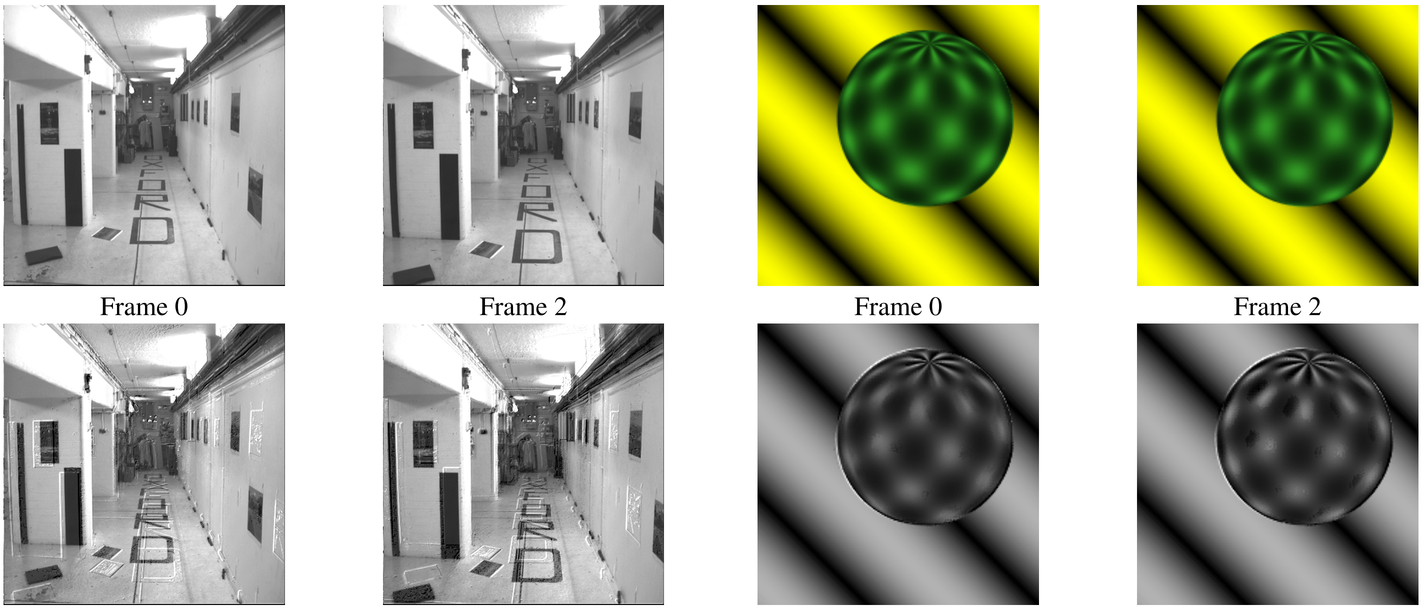


Figure 5: Optical flow frame prediction

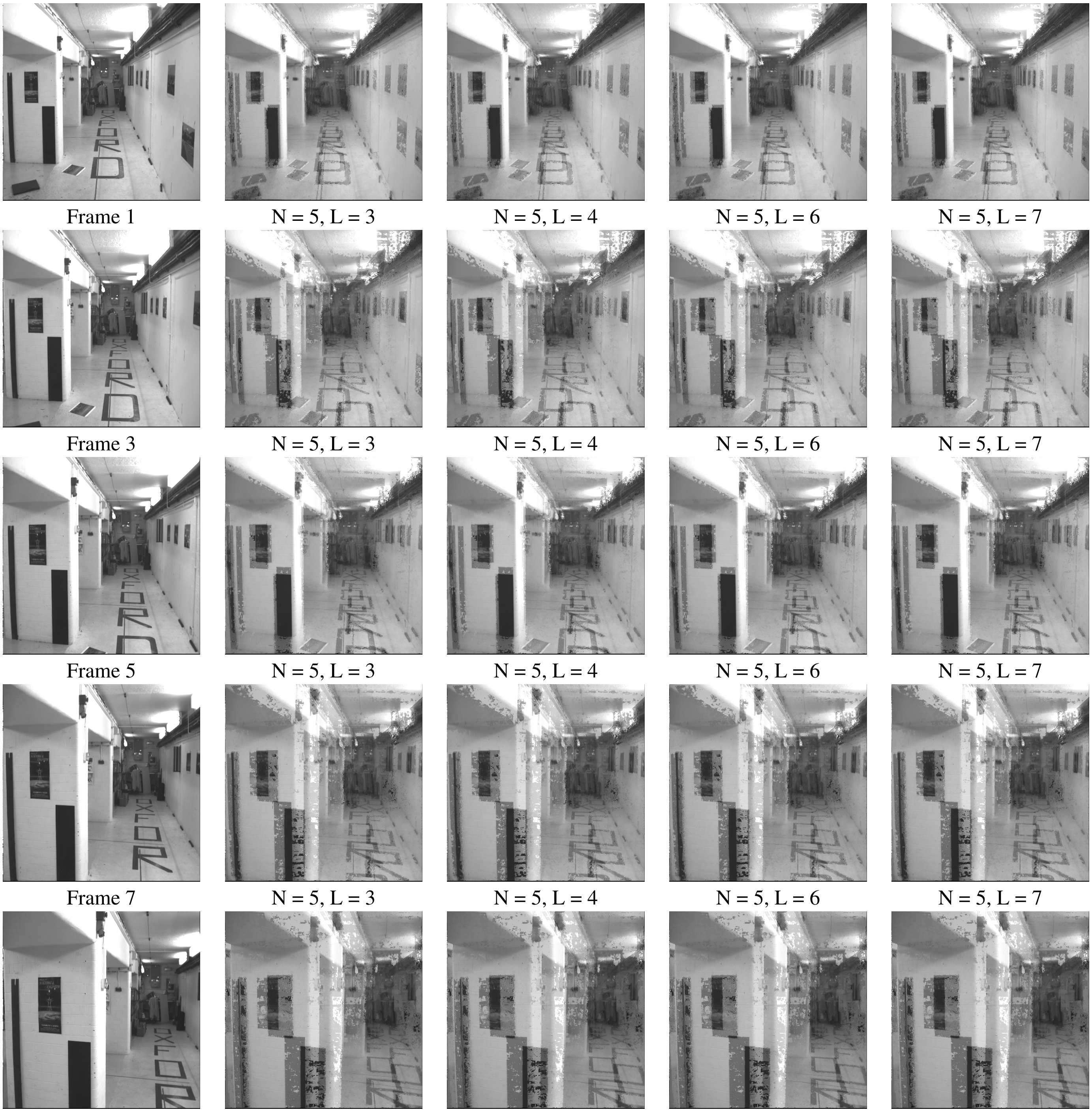
Corridor In Figure 5, we can see that we are able to reconstruct an frame back with a base frame and the calculated optical flow (with $N = 5$ and $L = 3$). This indeed confirms that we the optical flow calculated are reasonably good. The general observations regarding the reconstruction error from Problem 1 follows here as well. But by visual inspection the quality of this method slightly outperforms the previous method.

Sphere In Figure 5, we can see that we are able to reconstruct an frame back with a base frame and the calculated optical flow (with $N = 5$ and $L = 3$). The general observations regarding the reconstruction error from Problem 1 follows here as well.

Interpolation Next, we explore the primary motive of frame interpolation. We calculate the forward optical flow from Frame n to Frame $n + 2$ and the backward optical flow from Frame $n + 2$ to Frame n. We use them both to interpolate the intermediate Frame n. We explore this method for different block sizes ($N \times N$), for $N = 5, 11$ and different number of pyramid (scale) levels $L = 3, 4, 6, 7$. This methods takes higher compute power than the previous method so a large range of N was not explored.

Figure 6 shows the interpolated intermediate frames for the corridor dataset using Multiscale Lucas-Kanade for different values of N and L. Figure 7 shows the interpolated intermediate frames for the spheres dataset using Multiscale Lucas-Kanade for different values of N and L. From the figure, we can observe that there is no significant difference in the quality of reconstructed frames with variation in N and L. Only first nine interpolated frames for the sphere data is shown. Similar to previous algorithm, the interpolated frames of corridor data is of low quality. The quality of interpolated images for the sphere dataset is comparable to original frame and is much better compared to that of corridor dataset. Observations are similar to that of Problem 1.

Multiscale Lucas-Kanade algorithm tends to perform better when we have many pixels having the same brightness. Also, they perform better for large motion (pixels with large optical flows). In this case, multiscale estimates did not have any improvements over single scale estimates for the sphere data, where the motion was small.



Frame 1 | N = 5, L = 3 | N = 5, L = 4 | N = 5, L = 6 | N = 5, L = 7
Frame 3 | N = 5, L = 3 | N = 5, L = 4 | N = 5, L = 6 | N = 5, L = 7
Frame 5 | N = 5, L = 3 | N = 5, L = 4 | N = 5, L = 6 | N = 5, L = 7
Frame 7 | N = 5, L = 3 | N = 5, L = 4 | N = 5, L = 6 | N = 5, L = 7
Frame 9 | N = 5, L = 3 | N = 5, L = 4 | N = 5, L = 6 | N = 5, L = 7

Figure 6: Multiscale Lucas-Kanade Optical flow frame interpolation: Corridor

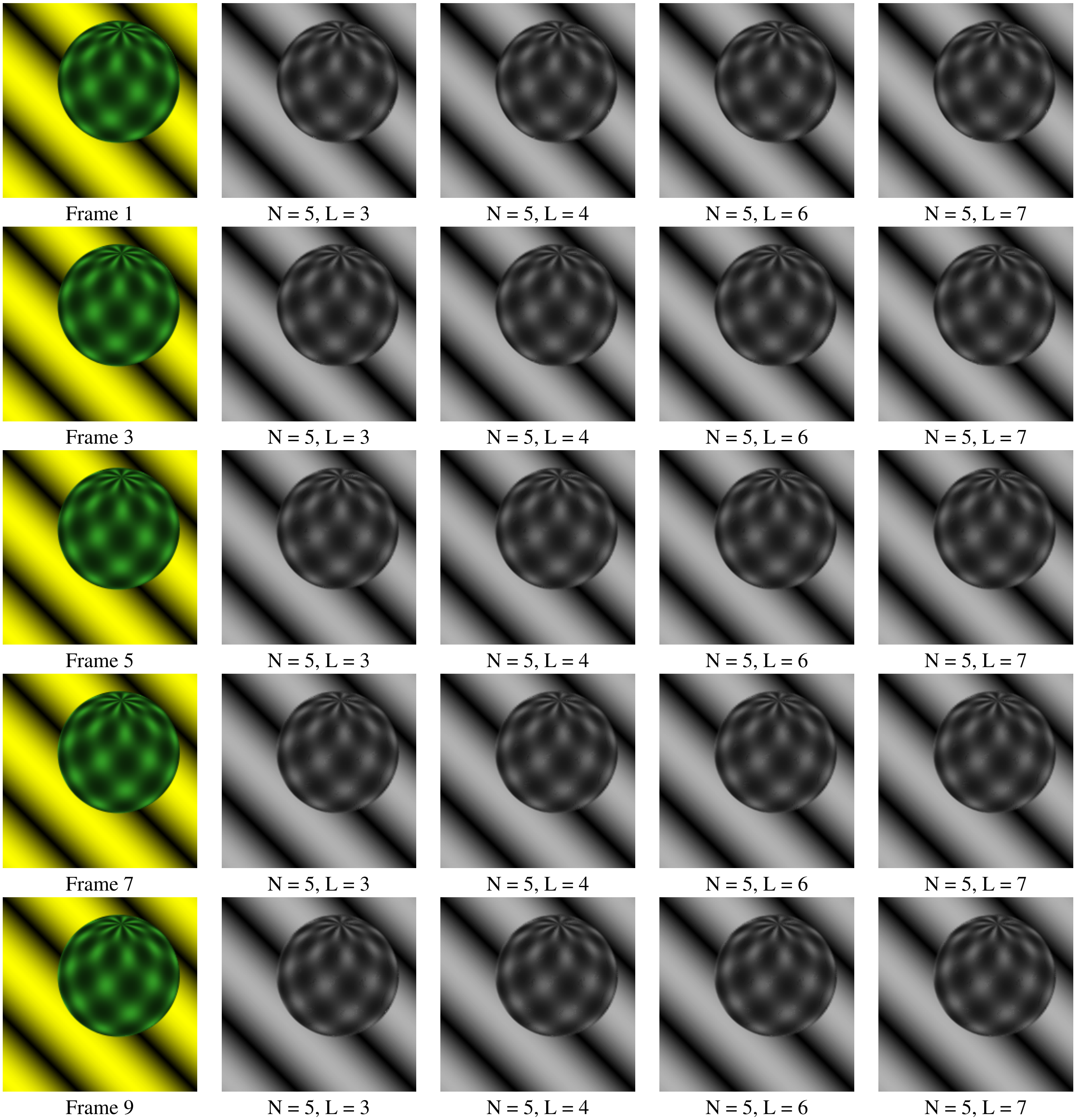


Figure 7: Multiscale Lucas-Kanade Optical flow frame interpolation: Sphere

Problem 3: Discrete Horn Schunck Optical Flow Algorithm

In this section we discuss about the performance of Frame interpolation based on Discrete Horn-Schunck algorithm.

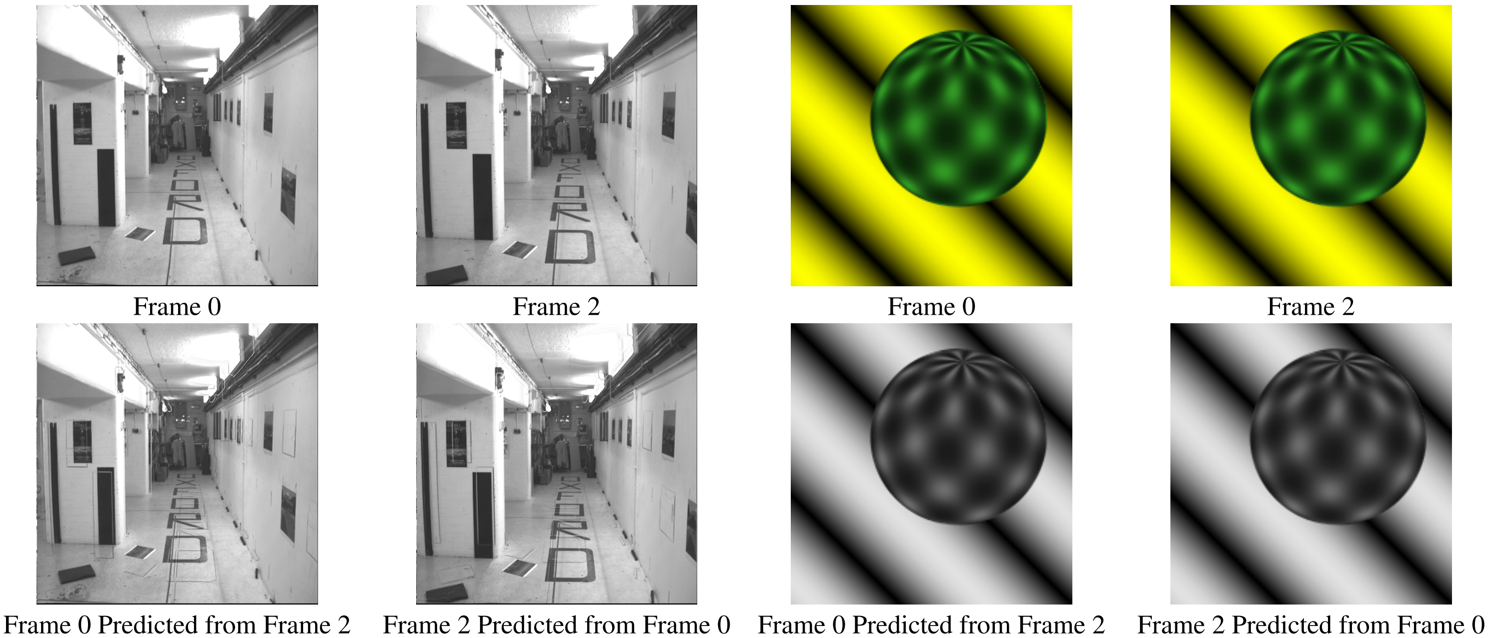


Figure 8: Optical flow frame prediction

Corridor In Figure 8, we can see that we are able to reconstruct an frame back with a base frame and the calculated optical flow (with $\lambda = 1$). This indeed confirms that we the optical flow calculated are reasonably good. The general observations regarding the reconstruction error from Problem 1 follows here as well. But by visual inspection the quality of this method outperforms both the previous method.

Sphere In Figure 8, we can see that we are able to reconstruct an frame back with a base frame and the calculated optical flow (with $\lambda = 1$). The general observations regarding the reconstruction error from Problem 1 follows here as well.

Interpolation Next, we explore the primary motive of frame interpolation. We calculate the forward optical flow from Frame n to Frame $n + 2$ and the backward optical flow from Frame $n + 2$ to Frame n. We use them both to interpolate the intermediate Frame n. We explore this method for different values of regularization parameter $\lambda = 0.01, 0.25, 0.5, 1, 1.5$.

Figure 9 shows the interpolated intermediate frames for the corridor dataset using Discrete Horn-Schunck for different values of λ. Figure 10 shows the interpolated intermediate frames for the spheres dataset using Discrete Horn-Schunck for different values of λ. The frames for $\lambda = 1.5$ is not shown due to space constraints. Only first nine interpolated frames for the sphere data is shown. From the figure, we can observe that there is difference in the quality of some reconstructed frames with variation in λ. Similar to previous algorithms, the interpolated frames of corridor data is of low quality. The quality of interpolated images for the sphere dataset is comparable to original frame and is much better compared to that of corridor dataset. Observations are similar to that of Problem 1.

Computationally this method is much cheaper compared to the previous methods. This method took much lesser time to compute while maintaining decent performance in calculating the optical flow.

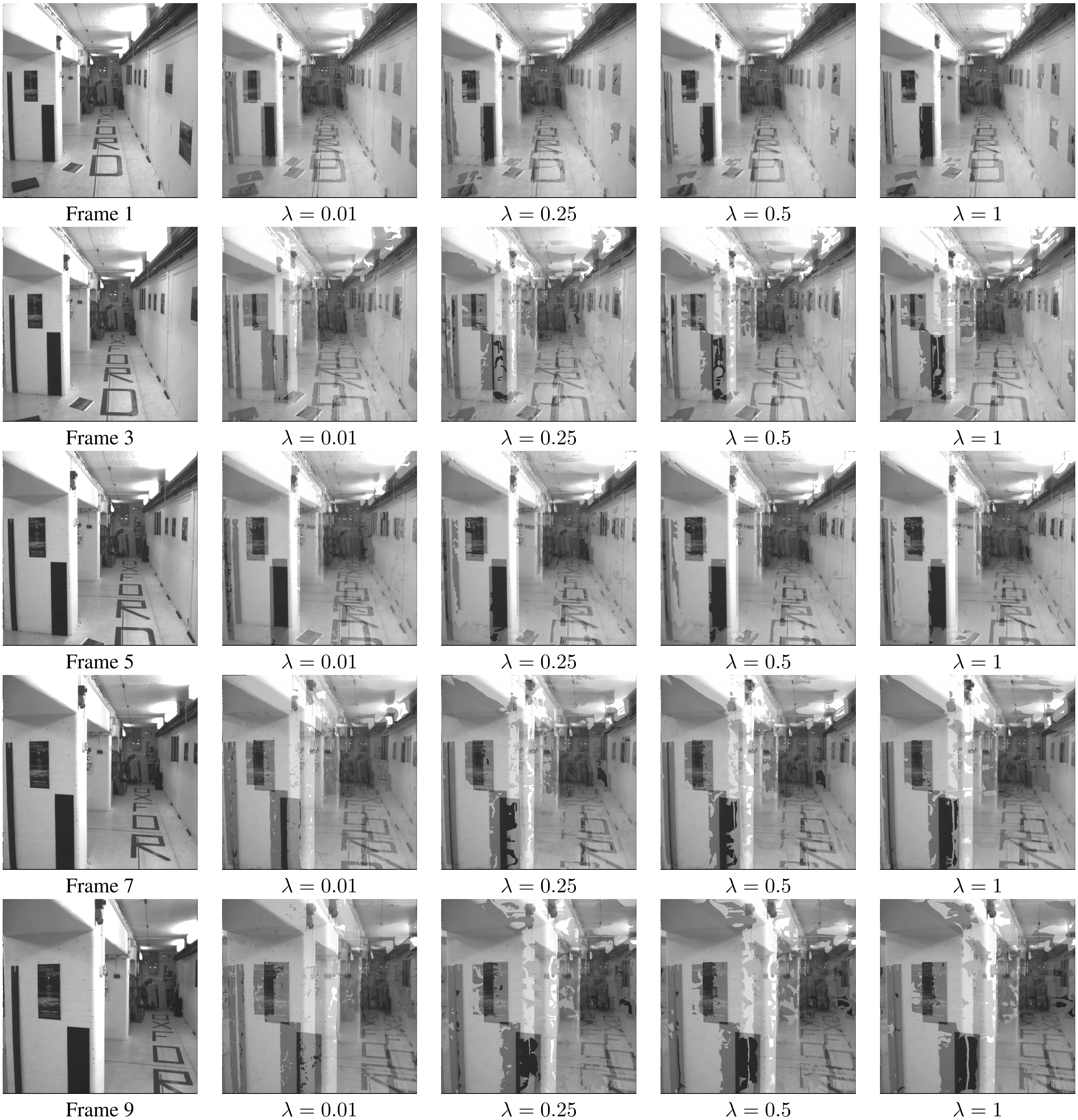


Figure 9: Horn-Schunck Optical flow frame interpolation: Corridor

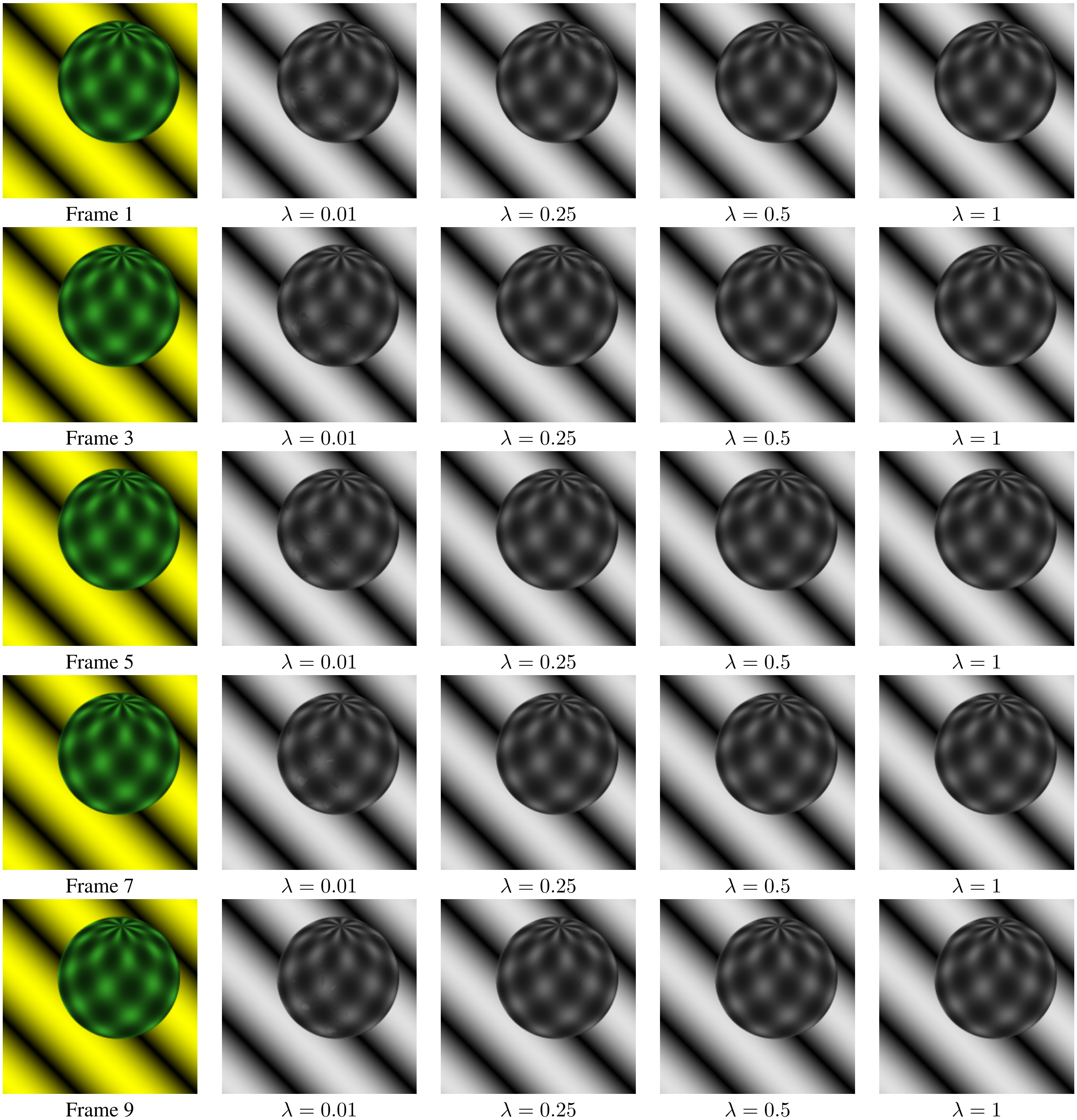
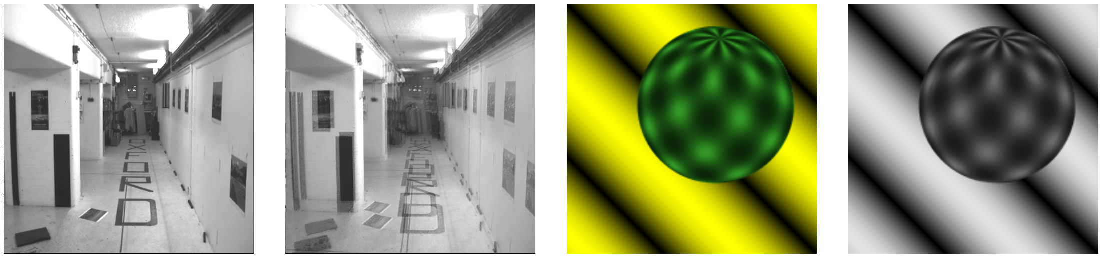


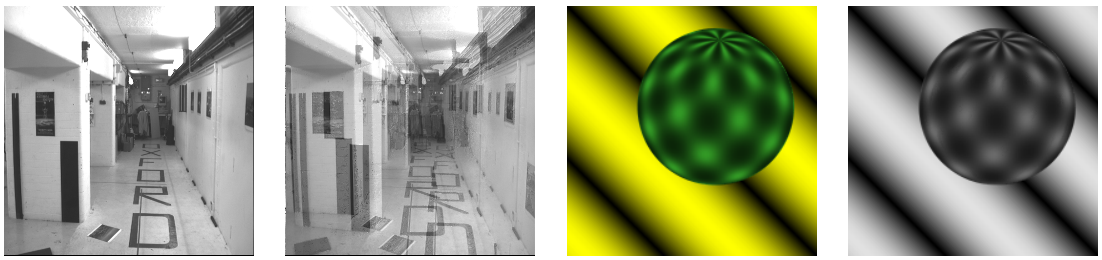
Figure 10: Horn-Schunck Optical flow frame interpolation: Sphere

Overlap Interpolation

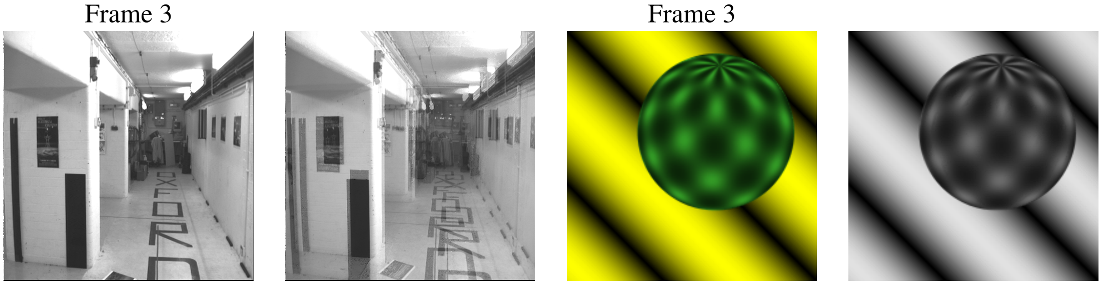
In this section, frame interpolation based on overlap of forward predicted frame and backward predicted frame is shown. Again, this works reasonably good with spheres data and poor with corridor data. Only few frames are shown below since this is not a standard way of interpolating.



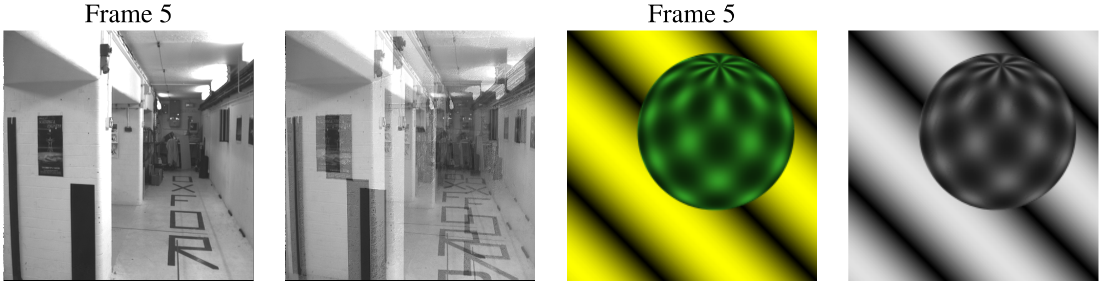
Frame 1 Frame 1



Frame 3 Frame 3



Frame 5 Frame 5



Frame 7 Frame 7

Pixel Mapping

The interpolation method proposed in [1] was used in the above algorithms, with slight modifications. The method was tested on Middlebury frame interpolation sample and it produced good results. We use bivariate spline approximation (scipy library) to sample intensity from the frame at non-integer indices. We also identify occlusions in frames and sample only from the non occluded frame.

References

[1] S. Baker et al. "A Database and Evaluation Methodology for Optical Flow". In: *2007 IEEE 11th International Conference on Computer Vision*. 2007, pp. 1–8. DOI: 10.1109/ICCV.2007.4408903.