VISION: Practical work 1 – implementation of Horn-Schunck and Lucas-Kanade methods

Francesco Zattoni, Angelo Prete, and Linnea Andersen

January 12, 2026

1 Introduction

In this assignment, we implement the Horn-Shunck and Lucas-Kanade methods for basic optical flow estimation. We report on the result achieved on 7 different images: 4 images with the ground truth provided and 3 images without. We test for optimal parameters of each method α for the Horn-Shunck method, and window size $n \times n$, along with different window types (Square, Gaussian, Circular), for the Lucas-Kanade algorithm.

2 `mysine`

(a) `I1` (b) `I2`

Figure 1: `mysine` images

For the `mysine` images, we use the provided ground truth (Figure 2a) to calculate the optimal parameters for each method, minimizing the angular error. The error values are shown in Table 1, and the velocity maps in Figure 2b-2c for the Horn-Shunck and Lucas-Kanade method, respectively.

Horn-Shunck We run the algorithm using $\alpha \in \{0.01, 0.1, 1.0, 10.0\}$. We find that the optimal value of $\alpha = 0.1$ with very low error on all measures (order of 10^{-5} or less), indicating that we can find the near-perfect flow, using the Horn-Shunck method.

Lucas-Kanade We apply this method using a window size $n \times n$ with $n \in \{3, 5, 7, 9, 11\}$, and find that the optimal window size is 3×3, with the error increasing with increase in window size. Using the optimal window size, we achieve errors in the order of magnitude of 10^{-2}.

Comparison and interpretation Since the mysine images are simple and noise-free, the Horn-Shunck method can provide a virtually perfect estimation: The global smoothness constraint (α) aligns perfectly with the sine wave. The Lucas-Kanade method has significantly higher error because it uses a local window, and at the border, the flow cannot be computed. Thus, the larger the window, the more boundary pixels are affected, leading to larger error.

Table 1: Comparison of error values for `mysine` using Horn-Schunck and Lucas-Kanade.

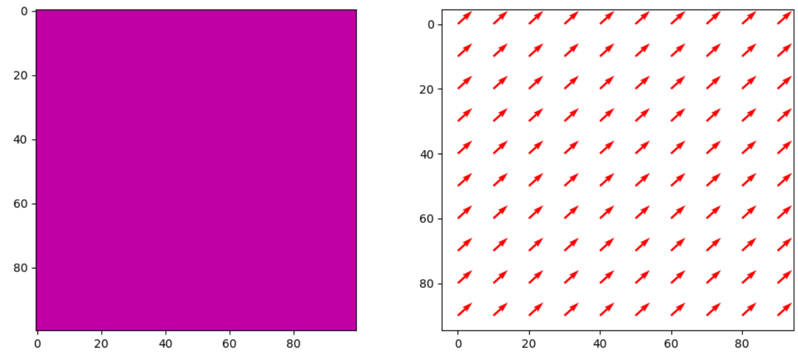
Param	EPE	Angular	Norm	RelNorm	Ang. SpaceTime
Horn-Schunck *(Param: α)*					
0.01	0.00001 ± 0.00019	0.00001 ± 0.00013	0.00000 ± 0.00007	0.00000 ± 0.00005	0.00001 ± 0.00011
0.1	0.00001 ± 0.00010	0.00000 ± 0.00007	0.00000 ± 0.00003	0.00000 ± 0.00002	0.00000 ± 0.00005
1.0	0.01318 ± 0.00118	0.00066 ± 0.00060	0.01312 ± 0.00119	0.00928 ± 0.00084	0.00446 ± 0.00040
10.0	0.86145 ± 0.00967	0.01227 ± 0.01044	0.86133 ± 0.00973	0.60905 ± 0.00688	0.45041 ± 0.00737
Lucas-Kanade *(Param: Window Size)*					
3	10.63551 ± 2.50062	0.10760 ± 0.39679	10.63551 ± 2.50062	7.52044 ± 1.76820	0.56572 ± 0.10565
5	31.39090 ± 8.74320	0.12315 ± 0.42223	31.39090 ± 8.74320	22.19672 ± 6.18238	0.61606 ± 0.09895
7	60.14529 ± 21.31653	0.18284 ± 0.50376	60.14529 ± 21.31653	42.52914 ± 15.07307	0.64229 ± 0.11361
9	95.97639 ± 40.28332	0.24128 ± 0.56637	95.97639 ± 40.28332	67.86556 ± 28.48461	0.66029 ± 0.12568
11	137.73018 ± 66.02084	0.29845 ± 0.61622	137.73018 ± 66.02084	97.38995 ± 46.68378	0.67532 ± 0.13561

Green *indicates performance for the optimal parameter*

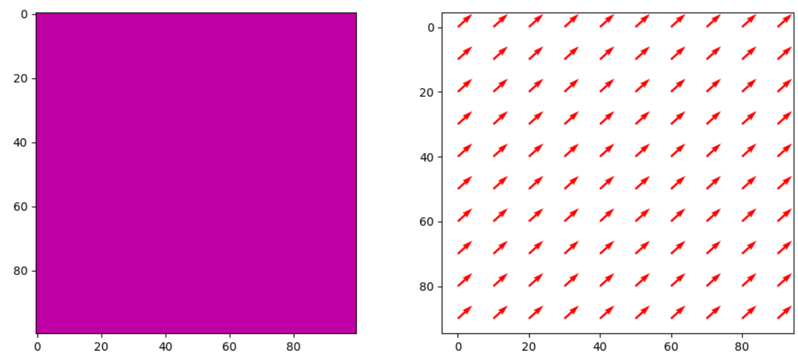
Effect of window type As 2 shows, there is no improvement when using a Gaussian or circular window. This is due to how the algorithm works: It calculates an average over the window size, and with the movement being constant all over the image, the average will be the same no matter which window type is used.

Table 2: Comparison of best error values for `mysine` using Lucas-Kanade with different window types

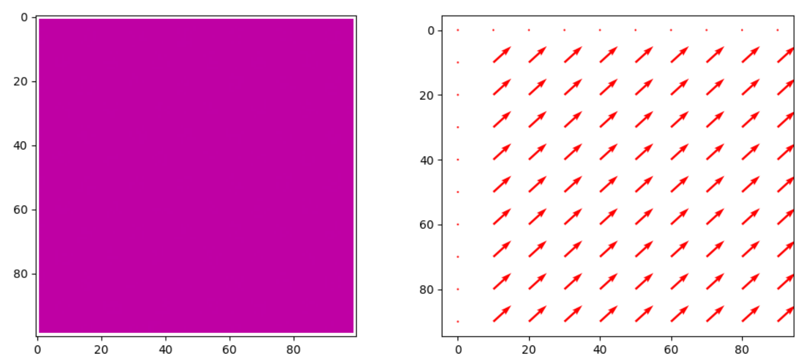
Window Type	EPE	Angular	Norm	RelNorm	Ang. SpaceTime
Square (3×3)	0.05602 ± 0.27579	0.06221 ± 0.30633	0.05602 ± 0.27579	0.03961 ± 0.19502	0.03784 ± 0.18630
Gaussian (3×3)	0.05602 ± 0.27579	0.06221 ± 0.30633	0.05602 ± 0.27579	0.03961 ± 0.19502	0.03784 ± 0.18630
Circular (3×3)	0.05602 ± 0.27579	0.06221 ± 0.30633	0.05602 ± 0.27579	0.03961 ± 0.19502	0.03784 ± 0.18630



(a) Ground truth



(b) Horn-Schunck method with $\alpha = 0.1, \quad error = 0.00000$



(c) Lucas-Kanade method with $n = 3, \quad error = 0.06222$

Figure 2: Optical flow results for `mysine`

3 `rubberwhale`

(a) `I1`

(b) `I2`

Figure 3: `rubberwhale` images

The `rubberwhale` images appear much more complex. With the ground truth available (see Figure 4a), we can calculate the optimal parameters for each method, minimizing the angular error. The error values are shown in Table 3, and the velocity maps in Figure 4b-4c for the Horn-Shunck and Lucas-Kanade method, respectively.

Table 3: Comparison of error values for `rubberwhale` using Horn-Schunck and Lucas-Kanade.

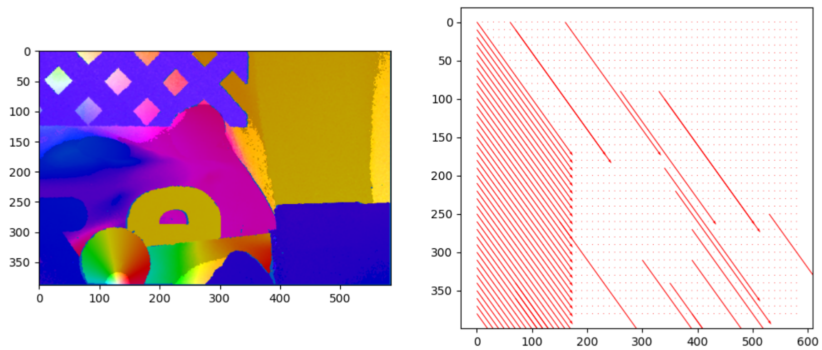
Param	**EPE**	**Angular**	**Norm**	**RelNorm**	**Ang. SpaceTime**
Horn-Schunck *(Param: α)*					
1.0	$37676248.17 \pm 295608308.80$	0.32014 ± 0.54172	$37676247.87 \pm 295608308.06$	0.24309 ± 0.37962	0.27938 ± 0.38963
50	$37676247.99 \pm 295608308.82$	0.21225 ± 0.49003	$37676247.80 \pm 295608308.10$	0.17803 ± 0.30792	0.19179 ± 0.34308
100	$37676247.98 \pm 295608308.82$	0.20467 ± 0.48939	$37676247.80 \pm 295608308.10$	0.17468 ± 0.30772	0.18518 ± 0.33927
150	$37676247.97 \pm 295608308.82$	0.20313 ± 0.49144	$37676247.80 \pm 295608308.10$	0.17441 ± 0.30850	0.18349 ± 0.33793
200	$37676247.97 \pm 295608308.82$	0.20316 ± 0.49325	$37676247.80 \pm 295608308.10$	0.17525 ± 0.30932	0.18348 ± 0.33737
Lucas-Kanade *(Param: Window Size)*					
3	$37676244.00 \pm 295608288.00$	0.29597 ± 0.54562	$37676244.00 \pm 295608288.00$	0.23033 ± 0.32634	0.26014 ± 0.38775
5	$37676244.00 \pm 295608288.00$	0.24187 ± 0.52567	$37676244.00 \pm 295608288.00$	0.19093 ± 0.28585	0.20992 ± 0.36505
7	$37676244.00 \pm 295608288.00$	0.24111 ± 0.54472	$37676244.00 \pm 295608288.00$	0.18542 ± 0.30055	0.20113 ± 0.36616
9	$37676244.00 \pm 295608288.00$	0.25317 ± 0.56852	$37676244.00 \pm 295608288.00$	0.18857 ± 0.31881	0.20385 ± 0.37270
11	$37676244.00 \pm 295608288.00$	0.27043 ± 0.59191	$37676244.00 \pm 295608288.00$	0.19484 ± 0.33506	0.21117 ± 0.38138

Green *indicates performance for the optimal parameter*

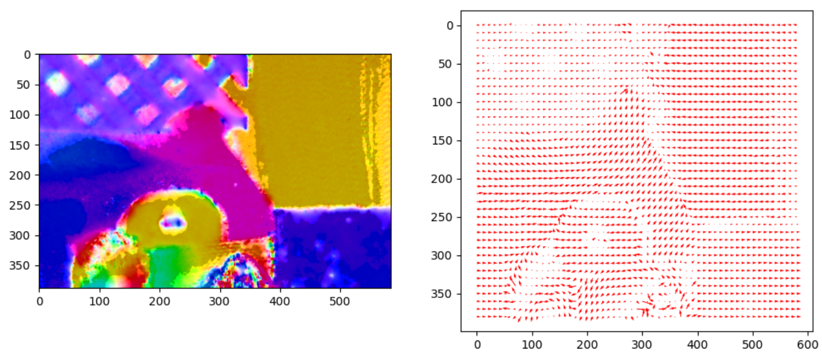
Horn-Shunck We run the algorithm using $\alpha \in \{1.0, 50, 100, 150, 200\}$. The best result is obtained with $\alpha = 150$, which gives an angular error of 0.20313 ± 0.49144. This high regularization weight is necessary because the whale is textureless.

Lucas-Kanade We apply this method using a window size $n \times n$ with $n \in \{3, 5, 7, 9, 11\}$, achieving best performance with $n = 7$ (angular error $= 0.241110.54472$).

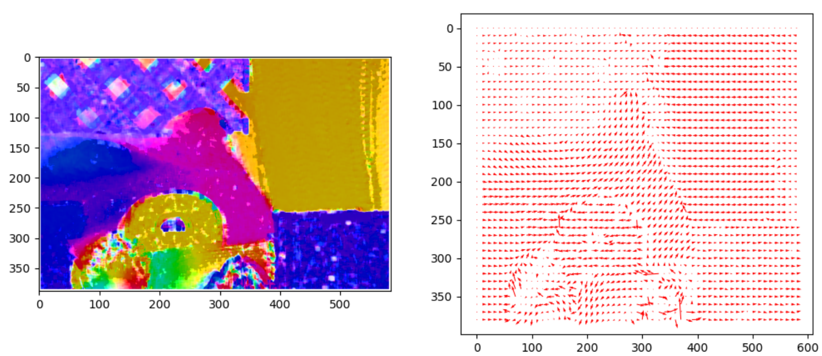
Comparison and interpretation From the high error values, the `rubberwhale` images are much more difficult to estimate for both methods. Furthermore, the EPE and Norm error values explode because the ground truth has pixels for which the optical flow is unknown, which is marked with a large value $1.6666668e + 09$.



(a) Ground truth



(b) Horn-Schunck method with $\alpha = 150, \quad error = 0.20313$



(c) Lucas-Kanade method with $n = 7, \quad error = 0.24111$

Figure 4: Optical flow results for `rubberwhale`

Effect of window type We achieve a slightly lower error using the square 7×7 window. The Gaussian window performed worse likely because of the smoothness of the whale: In such a textureless region, the optical flow solution needs to capture gradient information from distant neighbors at the edges of the window. But the Gaussian window assigns low weights to these pixels. Consequently, the Gaussian window must be physically larger to capture the same amount of distinct texture information that a Square window captures.

Table 4: Comparison of best error values for `rubberwhale` using Lucas-Kanade with different window types.

Window Type	EPE	Angular	Norm	RelNorm	Ang. SpaceTime
Square (7×7)	$37676248.00 \pm 295608308.84$	0.24111 ± 0.54472	$37676247.81 \pm 295608308.15$	0.18542 ± 0.30055	0.20113 ± 0.36616
Gaussian (9×9)	$37676248.02 \pm 295608308.84$	0.25867 ± 0.55376	$37676247.82 \pm 295608308.15$	0.19933 ± 0.30427	0.21362 ± 0.37068
Circular (7×7)	$37676248.01 \pm 295608308.84$	0.24149 ± 0.53616	$37676247.81 \pm 295608308.15$	0.18809 ± 0.29163	0.20399 ± 0.36498

4 square

(a) I1 (b) I2

Figure 5: square images

Appearing simple, the square images do prove challenging for our algorithms. We have the ground truth available (Figure 6a), so we can calculate the optimal parameters for each method, minimizing the angular error. The error values are shown in Table 5, and the velocity maps in Figure 6b-6c for the Horn-Shunck and Lucas-Kanade method, respectively.

Table 5: Comparison of error values for square using Horn-Schunck and Lucas-Kanade.

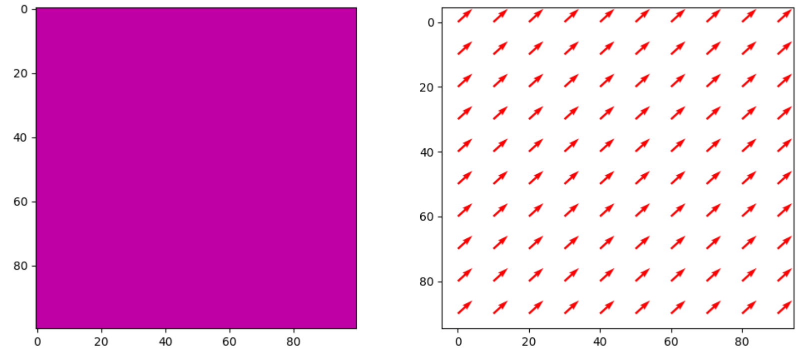
Param	EPE	Angular	Norm	RelNorm	Ang. SpaceTime
Horn-Schunck *(Param: α)*					
0.001	1.24877 ± 0.31437	0.18136 ± 0.11828	1.21417 ± 0.34558	0.64392 ± 0.18328	0.55449 ± 0.20492
0.01	1.22169 ± 0.32464	0.18180 ± 0.11684	1.18489 ± 0.35720	0.62840 ± 0.18944	0.53863 ± 0.20580
0.1	1.13923 ± 0.36300	0.18183 ± 0.11131	1.09261 ± 0.40576	0.57946 ± 0.21519	0.49494 ± 0.20930
1.0	1.19367 ± 0.35752	0.19093 ± 0.10918	1.15594 ± 0.38490	0.61305 ± 0.20413	0.52925 ± 0.21661
Lucas-Kanade *(Param: Window Size)*					
33	1.18000 ± 0.87188	0.95730 ± 0.75668	1.17314 ± 0.87924	0.62217 ± 0.46630	0.66627 ± 0.51443
35	1.17810 ± 0.87415	0.95592 ± 0.75836	1.17284 ± 0.87954	0.62201 ± 0.46646	0.66526 ± 0.51566
37	1.17766 ± 0.87467	0.95532 ± 0.75911	1.17290 ± 0.87944	0.62204 ± 0.46641	0.66492 ± 0.51607
39	1.19668 ± 0.87337	0.97303 ± 0.75679	1.19318 ± 0.87692	0.63279 ± 0.46507	0.67669 ± 0.51478
41	1.23454 ± 0.86813	1.00861 ± 0.74962	1.23295 ± 0.87023	0.65388 ± 0.46152	0.70033 ± 0.51048
43	1.27686 ± 0.85502	1.04526 ± 0.73815	1.27544 ± 0.85700	0.67642 ± 0.45450	0.72529 ± 0.50267
45	1.31810 ± 0.83963	1.08080 ± 0.72495	1.31679 ± 0.84154	0.69835 ± 0.44631	0.74952 ± 0.49365
47	1.35791 ± 0.82251	1.11513 ± 0.71026	1.35670 ± 0.82438	0.71952 ± 0.43720	0.77291 ± 0.48360

Green *indicates performance for the optimal parameter*

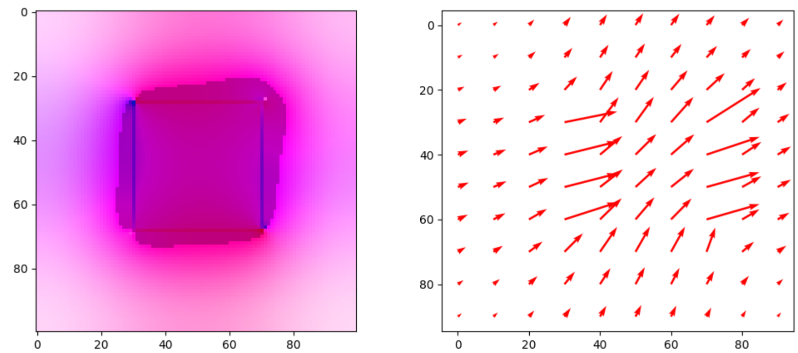
Horn-Shunck We run the algorithm using $\alpha \in \{0.001, 0.01, 0.1, 1.0\}$, finding the optimal $\alpha = 0.001$.

Lucas-Kanade We apply this method using a window size $n \times n$ with $n \in \{33, 35, 37, 39, 41, 43, 45, 47\}$. We find that the optimal window size is 37.

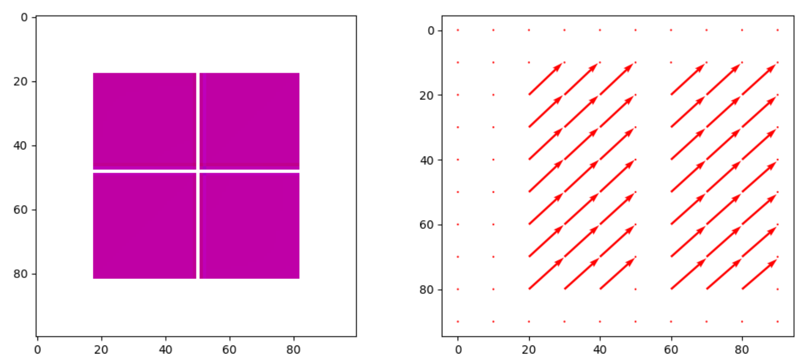
Comparison and interpretation Neither algorithm has a good performance, but the Horn-Shunck method is much better than the Lucas-Kanade method. The issue is the aperture problem caused by the uniform black background and the textureless white square. To the Lucas-Kanade window, the black pixels look as if they are not moving, resulting in large error. This is also likely to explain why the window size has to be so big: With that large of a window size, most windows will include a part of the square. The aperture problem also impacts the Horn-Shunck method: On the velocity map, it appears as if the square itself moves much faster than its surroundings. Here, we need a small α because otherwise, the motion of the square would smear out into the background, increasing the error.



(a) Ground truth



(b) Horn-Schunck method with $\alpha = 0.001, \quad error = 0.18136$



(c) Lucas-Kanade method with $n = 37, \quad error = 0.95532$

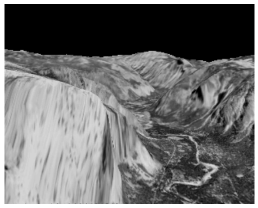
Figure 6: Optical flow results for `square`

Effect of window type Table 6 show that there is little effect of changing the window type, with the square type being the best, and the other window types performing slightly worse. For a pixel located along a straight edge to correctly estimate the global motion, its local window must be large enough to include a corner. A **Square window** assigns uniform weight to all pixels, allowing a distant corner at the window's border to contribute fully to the solution. In contrast, a **Gaussian window** attenuates the influence of pixels at the periphery, so, even if a Gaussian window physically overlaps a corner, the low weight assigned to that corner means the algorithm fails to incorporate it when calculating the solution.

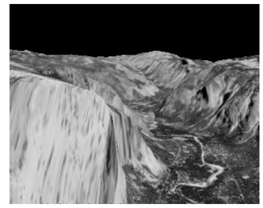
Table 6: Comparison of best error values for `square` using Lucas-Kanade with different window types.

Window Type	**EPE**	**Angular**	**Norm**	**RelNorm**	**Ang. SpaceTime**
Square (37×37)	1.17766 ± 0.87467	0.95532 ± 0.75911	1.17290 ± 0.87944	0.62204 ± 0.46641	0.66492 ± 0.51607
Gaussian (37×37)	1.18399 ± 0.86724	0.96033 ± 0.75292	1.17478 ± 0.87768	0.62304 ± 0.46547	0.66854 ± 0.51164
Circular (37×37)	1.21129 ± 0.87386	0.97990 ± 0.75054	1.20002 ± 0.87889	0.63642 ± 0.46611	0.68238 ± 0.51055

5 yosemite



(a) I1



(b) I2

Figure 7: yosemite images

The yosemite images, a real-life scenario, are also much more complex than the mysine and square. Due to the depth of the scene, the mountain in the bottom left moves significantly faster than the rest of the landscape, as it is closer to the camera. Using the provided ground truth, we calculate the optimal parameters, minimizing the angular error. The error values are shown in Table 7, and the velocity maps in Figure 8b-8c for the Horn-Shunck and Lucas-Kanade method, respectively.

Table 7: Comparison of error values for yosemite using Horn-Schunck and Lucas-Kanade.

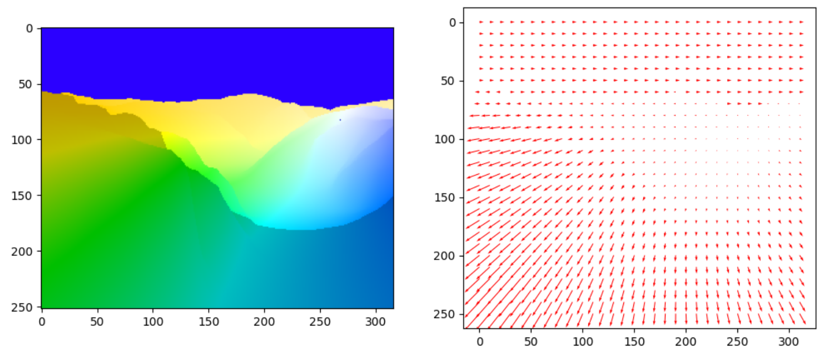
Param	EPE	Angular	Norm	RelNorm	Ang. SpaceTime
Horn-Schunck *(Param: α)*					
0.01	1.03143 ± 1.03972	1.02831 ± 1.28861	0.81986 ± 0.88098	0.45715 ± 0.33058	0.49093 ± 0.43654
0.05	1.00540 ± 1.00543	0.98129 ± 1.28509	0.84440 ± 0.89698	0.46316 ± 0.33343	0.46281 ± 0.40598
0.1	1.01795 ± 1.00357	0.98127 ± 1.27731	0.86326 ± 0.89460	0.47221 ± 0.33234	0.46362 ± 0.39459
0.2	1.04206 ± 1.01088	0.98884 ± 1.26569	0.89181 ± 0.89987	0.48746 ± 0.33134	0.47198 ± 0.38435
Lucas-Kanade *(Param: Window Size)*					
7	18.43317 ± 25.10246	1.00622 ± 0.77020	18.36308 ± 25.06750	17.50700 ± 20.58061	0.94800 ± 0.32310
9	31.35613 ± 42.36900	0.97627 ± 0.79251	31.28866 ± 42.34131	30.23103 ± 34.69344	0.95858 ± 0.33616
11	48.12948 ± 64.01443	0.95943 ± 0.81050	48.06084 ± 63.99081	46.68060 ± 52.44167	0.96665 ± 0.34982
13	68.99738 ± 90.45105	0.94978 ± 0.82556	68.92507 ± 90.42983	66.95603 ± 73.77659	0.97376 ± 0.36214
15	93.60778 ± 121.51872	0.94783 ± 0.83818	93.53026 ± 121.49900	90.58004 ± 98.49095	0.97942 ± 0.37384

Green *indicates performance for the optimal parameter*

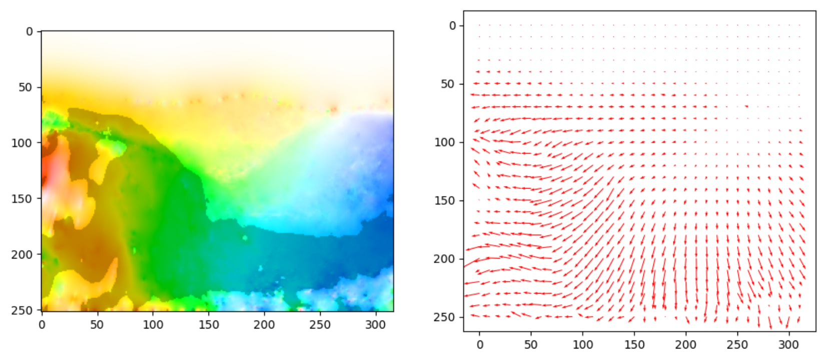
Horn-Shunck We run the algorithm using $\alpha \in \{0.01, 0.05, 0.1, 0.2\}$. The best α turns out to be 0.1.

Lucas-Kanade We apply this method using a window size $n \times n$ with $n \in \{7, 9, 11, 13, 15\}$. The best n is 15.

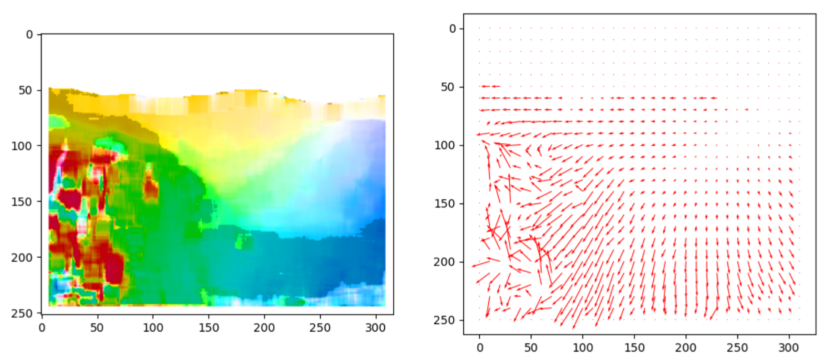
Comparison and interpretation The two methods have a similar bad performance on this image sequence. The methods are not able to pick up the lack of movement in the sky, and fail especially at the bottom left corner, with both methods estimating more chaotic movement.



(a) Ground truth



(b) Horn-Schunck method with $\alpha = 0.1$, $\quad error = 0.98127$



(c) Lucas-Kanade method with $n = 15$, $\quad error = 0.94783$

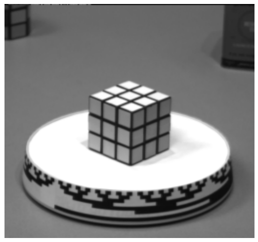
Figure 8: Optical flow results for `yosemite`

Effect of window type The choice of window type on the `yosemite` sequence does not improve the result much, but the circular window type has very similar error rates.

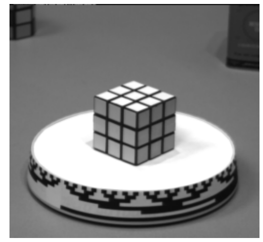
Table 8: Comparison of best error values for `yosemite` using Lucas-Kanade with different window types.

Window Type	EPE	Angular	Norm	RelNorm	Ang. SpaceTime
Square (11×11)	1.02341 ± 1.19948	0.71941 ± 0.81334	0.80787 ± 0.90526	0.47656 ± 0.38852	0.48031 ± 0.46389
Gaussian (13×13)	1.06615 ± 1.25290	0.75551 ± 0.82522	0.82090 ± 0.91874	0.48310 ± 0.38861	0.50301 ± 0.47781
Circular (11×11)	1.03501 ± 1.21666	0.72571 ± 0.80866	0.81081 ± 0.90610	0.47880 ± 0.38838	0.48588 ± 0.46670

6 rubic



(a) I1

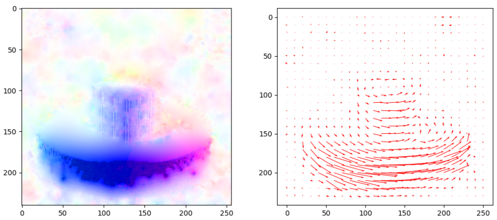


(b) I2

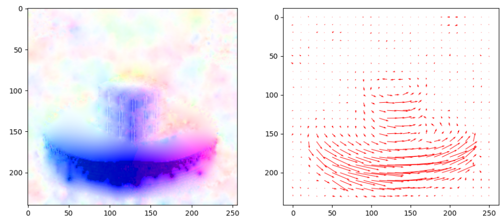
Figure 9: rubic images

The rubic sequence depicts a rubic cube located on a disk that spins counter-clockwise between images, causing the rubic's cube to move, while the background remains static. We do not have any reference flow here, so we will inspect flows for some value of the parameters, shown in Figure 10.

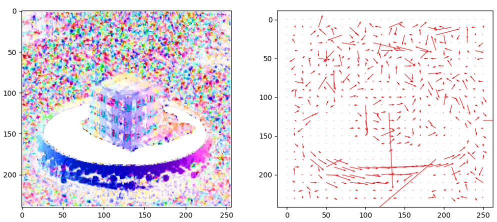
Comparison and interpretation The Horn-Shunck method captures the movement well, both the static background and the difference in speed between the disk and cube, although these appear blurry. It is hard to tell which α has a better performance. The Lucas-Kanade method struggles much more with this image sequence. The velocity map with a small window size shows a lot of noise in the background, while a larger window size proves a better result for the background, but also blurs the objects slightly, although they are still sharper than for the Horn-Shunck method.



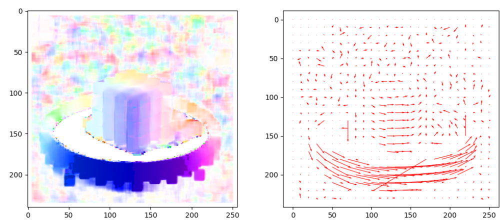
(a) Horn-Schunck method with $\alpha = 0.001$



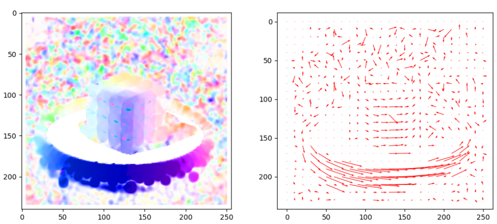
(b) Horn-Schunck method with $\alpha = 10.0$



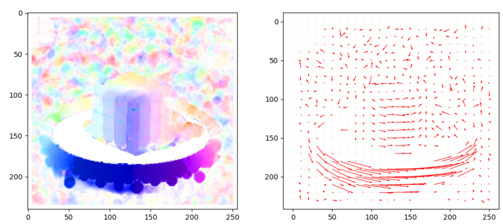
(c) Lucas-Kanade method with $n = 3$



(d) Lucas-Kanade method with $n = 11$



(e) Lucas-Kanade method with Gaussian window $n = 11$



(f) Lucas-Kanade method with Circular window $n = 11$

Figure 10: Optical flow results for `rubic`

7 `taxi`

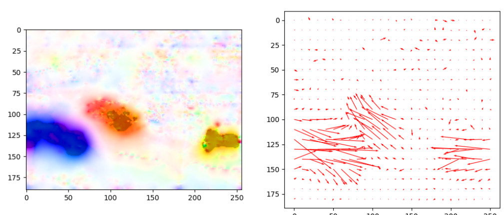
(a) `I1`

(b) `I2`

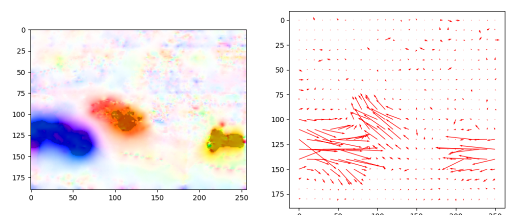
Figure 11: `taxi` images

The `taxi` sequence shows movement of three cars in different directions, while the background remains static. With no reference flows, we inspect flows for some value of the parameters, shown in Figure 10.

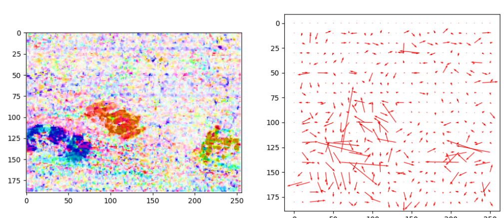
Comparison and interpretation We see a similar result to the `rubic` imges: The Horn-Shunck methods perform relatively well, but the objects appear blurry. The small window size used in Lucas-Kanade gives a lot of background noise, while increasing the window size decreases the noise level and renders sharper objects than for the Horn-Shunck method. Qualitatively, it is hard to judge whether a different window type decreases error: All three window types pick up on the motion of the three cars.



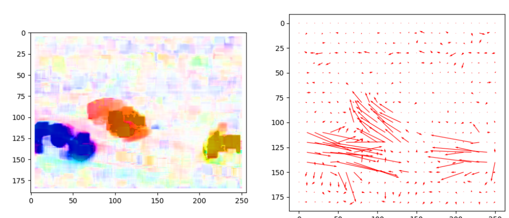
(a) Horn-Schunck method with $\alpha = 0.01$



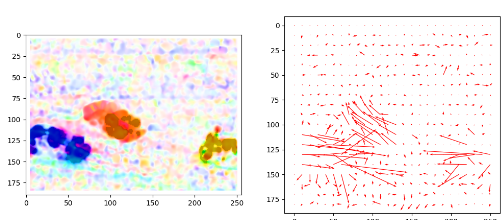
(b) Horn-Schunck method with $\alpha = 10.0$



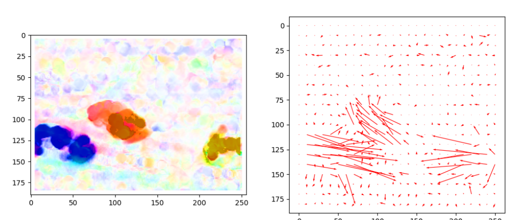
(c) Lucas-Kanade method with $n = 3$



(d) Lucas-Kanade method with $n = 11$



(e) Lucas-Kanade method with Gaussian window $n = 11$



(f) Lucas-Kanade method with Circular window $n = 11$

Figure 12: Optical flow results for `taxi`

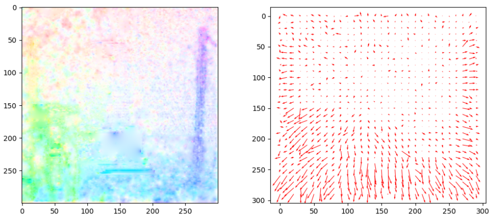
8 nasa

(a) I1 (b) I2

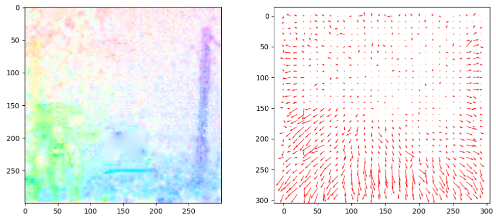
Figure 13: nasa images

The nasa images show a tiny forward movement of the camera, resulting in expansion from the center. Again we don't have the ground truth, so we simply plot the velocity maps for some parameters in Figure 14.

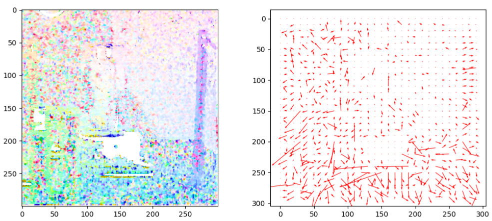
Comparison and interpretation It appears that Horn-Shunck performs better, at least than Lucas-Kanade with a small window size. All runs pick up that the pole is moving slightly faster than the remaining objects (since the pole object is closer to the camera), but it is hard to make out any of the other objects, likely because the shirt in the background has a knitted texture, and the Coca Cola can has a smooth surface. Using Lucas-Kanade with a Gaussian or Circular window doesn't appear to affect the resulting velocity maps much.



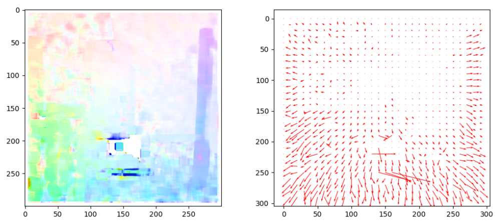
(a) Horn-Schunck method with $\alpha = 0.001$



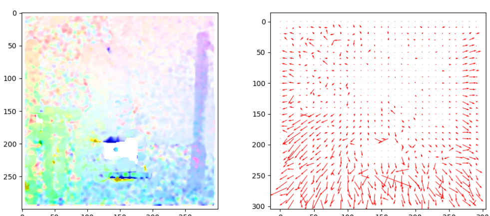
(b) Horn-Schunck method with $\alpha = 10.0$



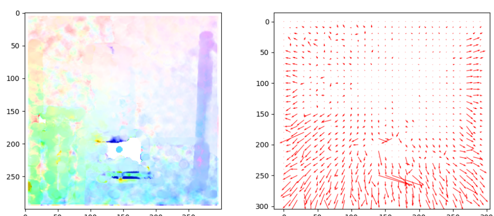
(c) Lucas-Kanade method with $n = 3$



(d) Lucas-Kanade method with $n = 11$



(e) Lucas-Kanade method with Gaussian window $n = 11$



(f) Lucas-Kanade method with Circular window $n = 11$

Figure 14: Optical flow results for `nasa`

9 Conclusion

We implemented and ran two different optical flow algorithms on 7 different types of images. Judging by the angular error for the images with ground truths provided, Horn-Shunck performed best for `mysine`, `rubberwhale`, `square`, while Lucas-Kanade performed best for `yosemite`. These results suggest that the optimal choice between these algorithms depends heavily on the scene structure. While the Horn-Schunck method imposes a global smoothness constraint, which allows it to successfully estimate motion in textureless regions where local information is ambiguous, the Lucas-Kanade method is more computationally efficient and robust to noise due to its use of a local window. This makes it excel in texture-rich areas, such as the Yosemite mountains.

In general, we saw little improvement using different window types, although in theory, the Gaussian window should help with noise reduction, and a Circular window with bias towards corner pixels of a square window.