Exercise 1: Optical flow

Žan Pušenjak, *ACVM 2024/25, FRI, UL*

I. Introduction

In this exercise, we were given the task of computing the optical flow in a sequence of images. We compared different computational approaches using different methods and their improvements. We looked at how the algorithms act when ran with different parameters and selected the best ones.

II. Experiments

We implemented two algorithms. Lukas-Kanade (LK) and Horn-Schnuck (HS) optical flows. To further improve the speed and results we improved LK with the pyramidal approach and pre-initialized HS with the LK method. We computed the optical flow (OF) on a random noise rotated by one degree and three pairs of images. Pairs can be seen on Figure 1 and will be referenced throughout the report.

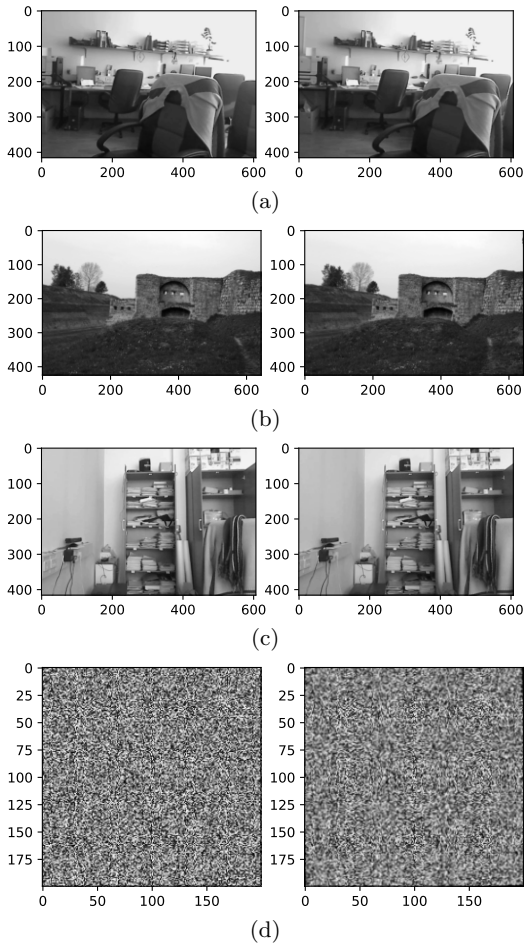

Figure 1: Images used for evaluation

A. Lukas-Kanade

LK algorithm depends on two variables. N that denotes the size of the neighborhood for computing the flow at each pixel and σ that is used in gaussian smoothing and derivation. We fixed $N = 6$ since it gave the best results in general, but the variation in N did not change the results that drastically. On the other hand σ does have a different effect based on the image size. Taking that into account we set the parameter dynamically with the formula $\sigma = min(w, h) * 0.005$ where w and h are the width and height of the image. This gave us an algorithm that worked on different sizes of images out of the box with no parameter tweaking.

The results on all the test images can be seen on Figure 2

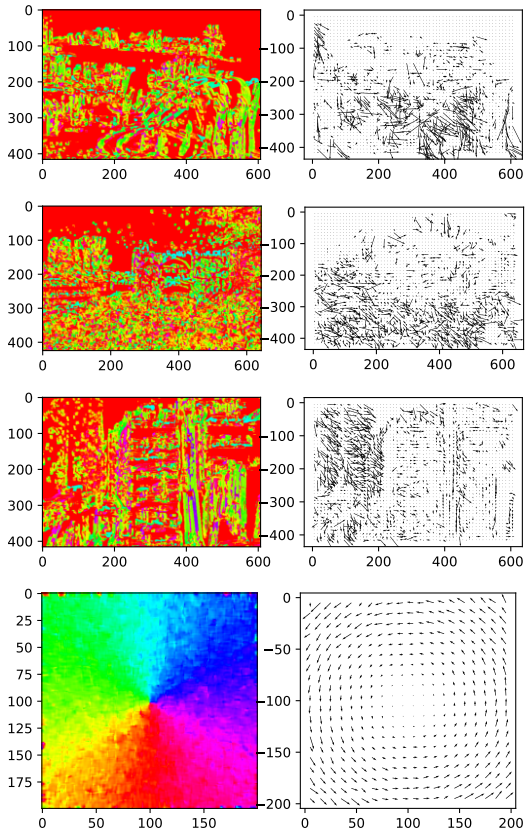

Figure 2: Lukas-Kanade optical flow computations.

Because of our assumptions while deriving the OF this LK method is not stably in areas with similar texture. That is why we used Harris corner detection[1] method to show the areas where the algorithm is stable (an example of the stability regions on image 1c can be seen on Figure 3, where the unstable regions are black).

B. Pyramidal Lukas-Kanade

To improve the classic LK algorithm we used the pyramidal approach that can detect larger movements on the flow. We observed that when building full pyramids some monotone images, like the rotating random noise would get a worse OF computation. This happened, because the very top of the pyramid that detects the large movements should not detect a movement, but because the noise is similar throughout there is

[1] https://docs.opencv.org/4.x/dc/d0d/tutorial_py_features_harris.html

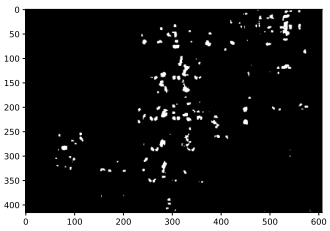

Figure 3: Stable areas on image 1c

a chance that a movement will be detected. To avoid that we limited pyramid build process to not go further if the image at current level is not at least $50x50$ pixels.

The results on all the test images can be seen on Figure 4

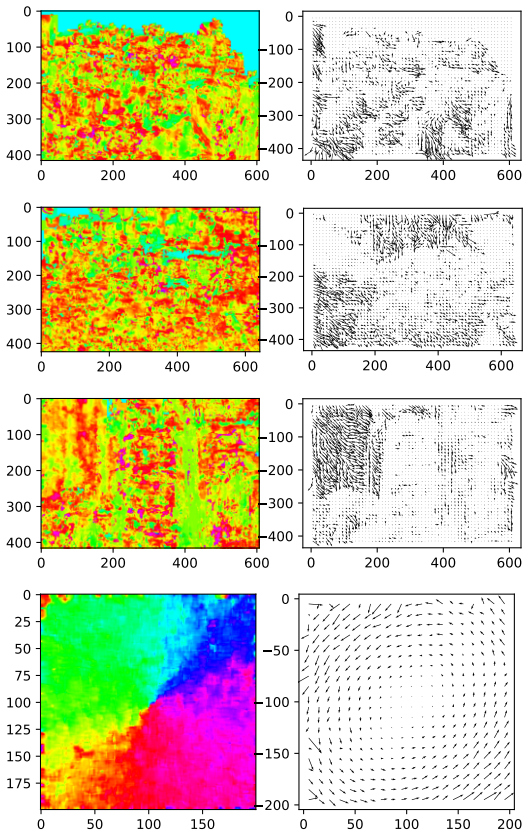

Figure 4: Lukas-Kanade Pyramidal optical flow computations.

When comparing results with the normal LK we can observe that this approach gives better results that are much less noisy and work better when the depths of objects vary.

C. Horn-Shnuck

With the HS algorithm there are more parameters to be determined, prior to the computation. Similarly as with LK we used gauss smoothing and derivation so we determined N and σ the same way as with LK. Since HS is a iterative algorithm it needs a stopping scenario. We determine that with parameters n_iter that represents the maximum number of iterations and Δ_{tr} that represents the threshold for change. If the change in current iteration is too small the algorithm will stop. With experimentation we decided on the values $n\_iter = 1000$ and $\Delta_{tr} = 10^{-5}$ Additionally HS has a λ parameter that determines the trade-off between smoothness and accuracy of the OF. Since each image has a different optimal λ we fixed its value to 0.5 and gave the user an option of controlling the parameter. It would be interesting to find a way to find the optimal lambda value by using a metric to describe the accuracy of the computed OF and a method like grid search or to set the parameter based on the images properties (texture complexity, noise level...).

The results on all the test images can be seen on Figure 5

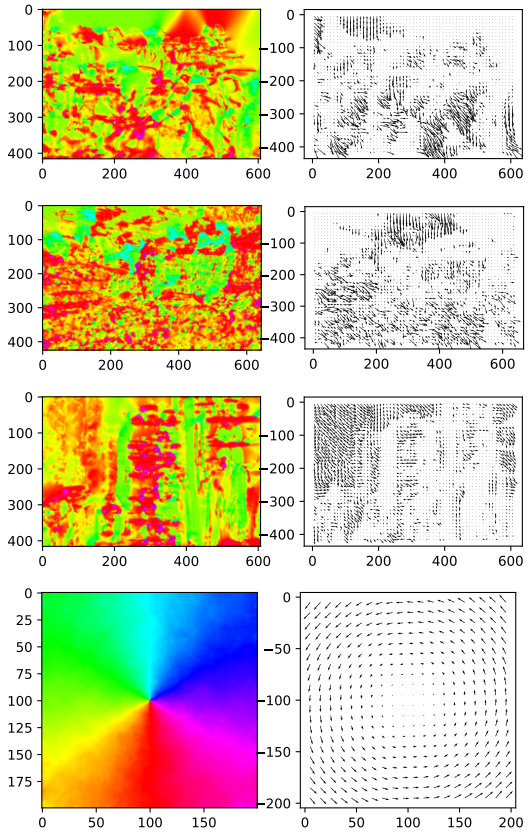

Figure 5: Horn-Shnuck optical flow computations.

D. Preinitialized Horm-Shnuck

HS algorithm takes much longer to compute than LK due to its iteration based nature. To battle that we tried to first compute the OF with LK and than ran the HS computation of the output of LK hoping for improved speed and maybe even improved results. In Table I we can observe that the time did not improve drastically and the results stayed the same.

Table I: Times of different algoritms on image 1c in seconds

Algorithm	LH	LK pyramid	HS	HS initialized
Time	0.02	0.04	11.02	10.49

III. Conclusion

When comparing the final results we can see that the LK-pyramidal and HS gave very similar results (although LK-pyramidal was noisier with larger variance), while the normal LK gave a very noisy OF compared to the previous two and only worked perfectly on the rotated random noise. We can observe that normal LK had difficulty with depth, as the perception of movement changes with depth and the algorithm does not account for that.