Optical flow

Ivan Nikolov

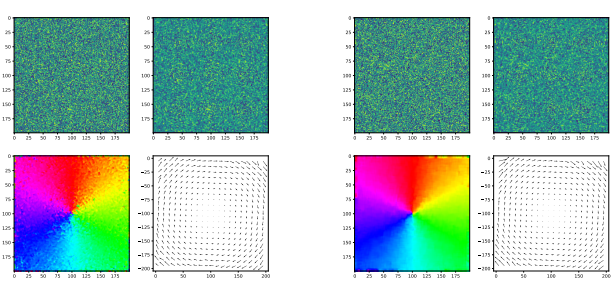
I. Introduction

Optical flow is used to represent motion in a sequence of images. It can be interpreted as a velocity field that transforms the image to the next one in the sequence. For this assignment, we implemented two algorithms that are used for optical flow estimation: Lucas-Kanade and Horn-Schunck. Both of them follow the small motion assumption and brightness constancy constraint. Lucas-Kandade also makes an assumption that the flow is constant in a local neighborhood and solves the flow equations for that pixel with that local information and the least squares criterion. Horn-Schunck represents the flow estimation as an energy minimization problem. It introduces an energy smoothness function. The flow estimation is done for the whole image iteratively.

II. Experiments

A. Lucas-Kanade

As a benchmark and reference, we tested our basic implementation on an image with random noise which was rotated by 1 degree. Our results shown on Figure 1 were comparable and similar to the ones given as a reference, which suggests that the algorithm computes the optical flow correctly.



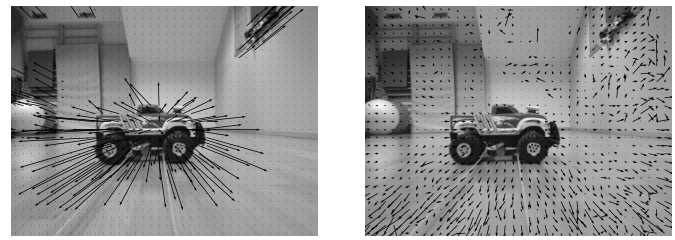
(a) $\sigma = 1, N = 3$ (b) $\sigma = 1, N = 9$

Figure 1: Optical flow response of Lucas-Kanade on a random noise image rotated by 1 °.

The two main parameters that this algorithm has are the neighborhood size N and the standard deviation of the Gaussian kernel σ. Depending on the use-case, small N and σ can result in a noisy flow field. This is visible on Figure 1. Using larger σ values will result in oversmoothing the images and gradient responses and loss in information that could represent small motions.

For our experiments we only tuned N depending on the use-case and σ was left to a value of 1.

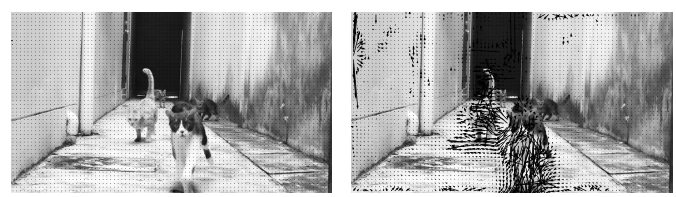
1) Harris response for flow reliability: Lucas-Kande uses the covariance matrix of local gradients to solve the optical flow equation, and it might fail in regions where the eigenvalues of the matrix are too small, or the ratio of the eigenvalues is too large. Regions with these features typically have low texture or are near the edges. On Figure 2 we can observe that by using Harris response, we filter out the unreliable flow estimates on regions with low textures and only keep the ones for our main object of interest (the toy truck).



(a) With Harris response (b) Without Harris response

Figure 2: Comparison of Lucas-Kanade algorithm with and without using Harris response thresholding (N=11, $\sigma = 1$).

2) Lucas-Kandade with pyramids: In many real-world scenarios, the small motion assumption often gets violated. To solve this problem, we used a Gaussian pyramid and estimated the flow at lower resolutions, and gradually improved the flow estimates while increasing the resolution.

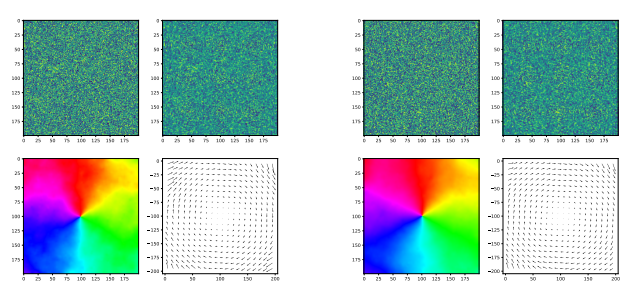


(a) No Gaussian pyramids (b) Gaussian pyramids(levels=4)

Figure 3: Comparison of Lucas-Kanade algorithm with and without using Gaussian pyramids (N=11, $\sigma = 1$).

This is evident in Figure 3 where the basic Lucas-Kandade algorithm fails to detect the fast movement of the cats in the video [1]. The pyramidal implementation on the other hand uses down scaled version of the images and successfully detects the running motion as optical flow.

B. Horn-Schunck



(a) $\lambda = 0.5, iter = 500$ (b) $\lambda = 0.5, iter = 1000$

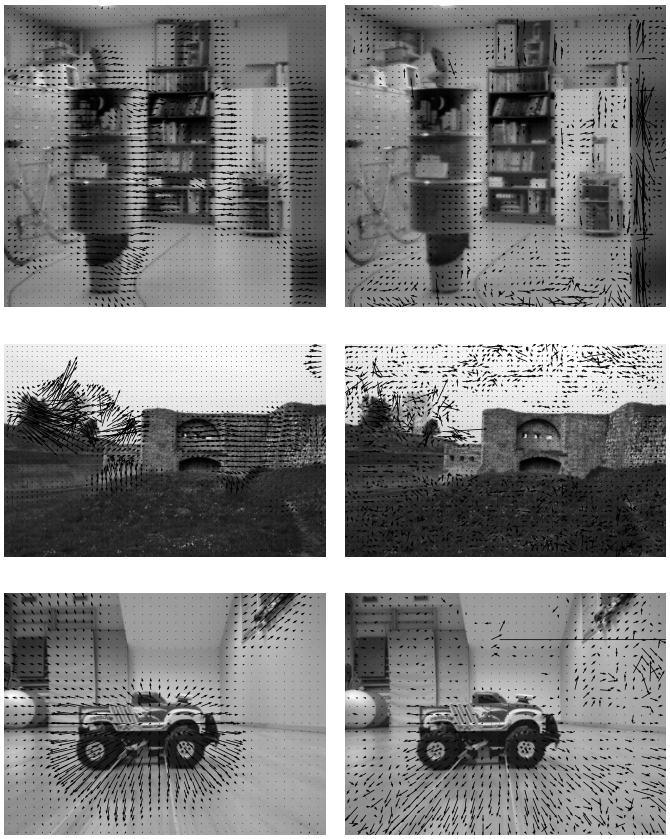
Figure 4: Optical flow response of Lucas-Kanade on a random noise image rotated by 1 °.

As for the Lucas-Kanade algorithm, we tested Horn-Schunck on a random noise image rotated by a 1 °. On Figure 4 we can see that the Horn-Schunck optical flow is smoother than the Lucas-Kanade flow shown on Figure 1. This is due to the fact that Horn-Schunck uses an additional smoothness cost and optimizes the flow in the whole image as a whole.

In our experiments, we found out that the number of iterations can greatly affect the algorithm output. Larger values result in smoother and more accurate optical flow shown on Figure 4b, as opposed to lower values where the noise is visible in the HSV plot at Figure 4a. Larger number of iterations also results in slower execution time.

C. Algorithms comparison

In order to compare the strengths and weaknesses of both algorithms, we compared the flow estimates on 3 additional images.



(a) Horn-Schunck (b) Lucas-Kanade

Figure 5: Comparison of optical flow fields from both algorithms.

On Figure 5 we can see that in general, Horn-Schunck performs better than Lucas-Kanade in terms of field accuracy and smoothness. Both algorithms struggle with optical flow estimation in regions with no or little texture or areas around the edges of the objects.

We compared the execution time of both algorithms under different parameters. On Table I the execution times on an image size of 288×384 are presented. The algorithm abbreviations are the following:

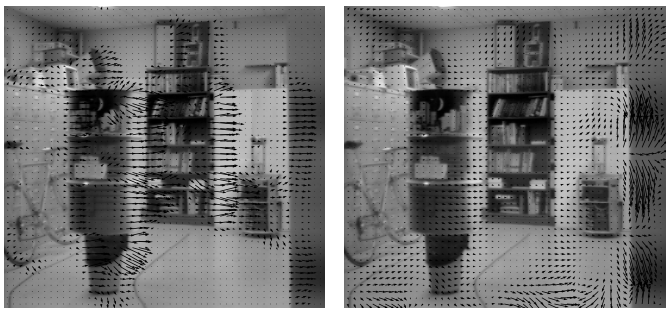
- HSLK - Horn-Schunck (initalized with Lucas-Kanade);

Method	Iterations	N	Time (seconds)
HSLK	1000	/	3.832
HS	1000	/	1.688
HS	500	/	0.901
HS	500	/	0.888
LKP	/	9	0.039
LK	/	9	0.031
LK	/	3	0.009

Table I: Comparison execution of Different Optical Flow Methods

- HS - Horn-Schunck;
- LKP - Lucas-Kanade with pyramids;
- LK - Lucas-Kanade.

Horn-Schunck initialized with Lucas-Kanade performed worse, because the initialization did not bring faster convergence and the flow estimation was more noisy and inaccurate.



(a) Horn-Schunck with 0 initialization (b) Horn-Schunck with Lucas-Kanade

Figure 6: Comparison of Horn-Schunck initialization techniques ($iterations = 500$)

Although Lucas-Kanade on our examples had worse flow estimation, the faster execution time could be crucial in application where the time constraint is low.

III. Conclusion

For this assignment, we implemented two of the most used optical flow methods (Lucas-Kanade and Horn-Schunck). From our experiments Horn-Schunck is more accurate, with the main drawback being the longer execution time. Both algorithms suffer from scenarios where the small motion assumption is violated, Lucas-Kanade solves this with Gaussian pyramids. The instability of Lucas-Kanade can be solved with using the response from the Harris detector. Initializing Horn-Schunck with Lucas-Kanade did not decrease iterations needed for convergence.

References

[1] "Agile Feline Runs and Jumps Over Group of Cats - 988915," https://www.youtube.com/watch?v=1AacxhpOm3w&ab_channel=RMVideos, accessed: March 15, 2024.