Implementation of two methods of optical flow

- VISION Practical Session Report -

Yanis Chemli, Diana Mandache

24/1/17

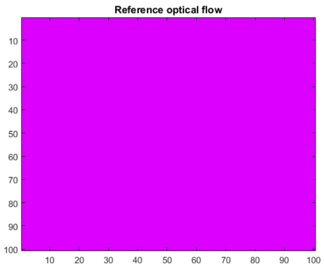
1 Introduction

In this paper we present the results obtained after implementing two methods for optical flow computation i.e. Horn Schunk and Lucas Kanade. These methods serve as a tracking mechanism of objects from video sequences. In this work we have multiple pairs of frames as input data and, for some of them, there is also provided the ground truth for the optical flow; in these cases the mean angular error between the reference and the estimation will be computed.

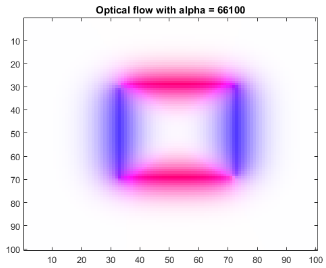
2 Horn and Schunk Method

Horn Schunk is an iterative algorithm that takes as parameters N the number of iterations and α the level of regularization, the latter serves at smoothing the flow in order to ameliorate the aperture problem, as it will be shown in the visual results we obtained. This particularity of the method consists in filling the flow information missing from the flow of the surrounding area (if there is any).

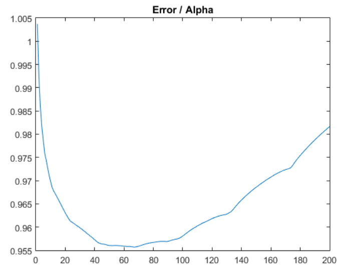
For the *square* data for instance, we notice that even if to the human eye it appears that only the white square moves towards the upper right corner of the image while the black background remains motionless, the ground truth tells us that all the pixels of the image move in the same direction and with the same velocity, giving a uniform optical flow. This is the aperture problem.



(a) Reference optical flow

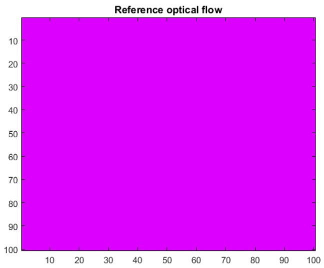


(b) $\alpha = 66.100$ minimizing the error

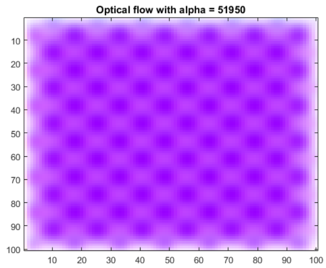


(c) Error evolution related to $\alpha = 100...200.000$

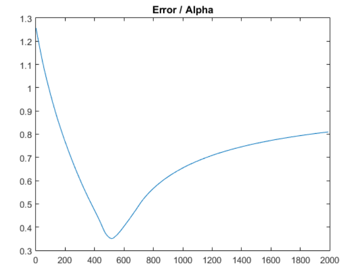
Figure 1: Study of α minimizing the mean angular error on *square* data



(a) Reference optical flow

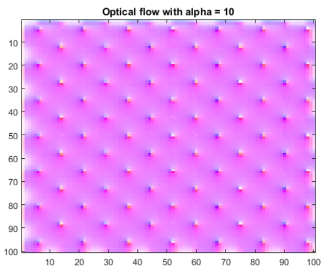


(b) $\alpha = 51.950$ minimizing the error

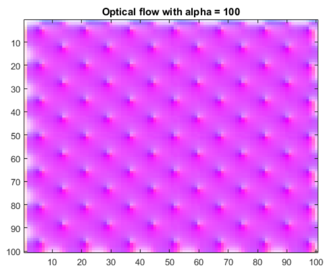


(c) Error evolution related to $\alpha = 100...200.000$

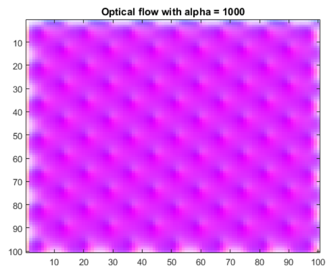
Figure 2: Study of α minimizing the mean angular error on *mysine* data



(a) $\alpha = 10$



(b) $\alpha = 100$



(c) $\alpha = 1000$

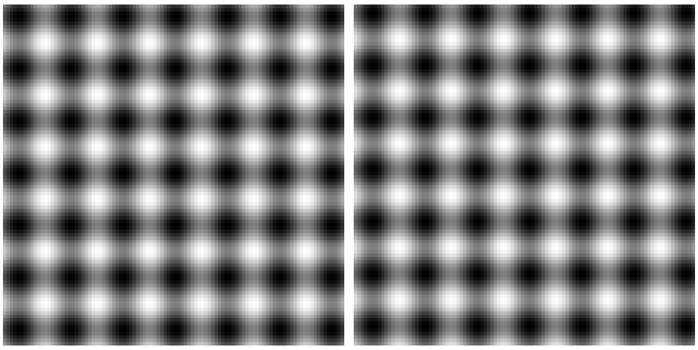
Figure 3: Effect of regularization parameter α on *mysine* data

3 Lucas and Kanade Method

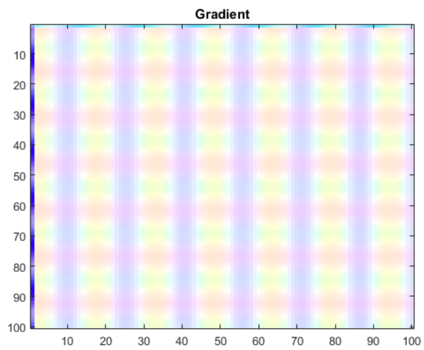
This is a one-step algorithm that solves the optical flow equations for every pixel of the image taking into account its neighboring pixels inside a window of size w. For simplifying the computation we have chosen to ignore the boundaries of the image up to a width of $w/2$.

As expected, the estimation is closest to the reference velocity when the window is larger, consequently we can deduce that this method is very sensible to the aperture problem.

We depict this inconvenience of the method by studying the effects of different window sizes on the *mysine* data. While the ground truth and also the visual observation tell us that there is present a uniform motion of all the pixels of the image, if we observe only a sufficiently small region of the sequence we can be tricked to consider that only some pixels are indeed in motion (like in the case of the *square* sequence). This small region we observe is encoded in the size of the window.

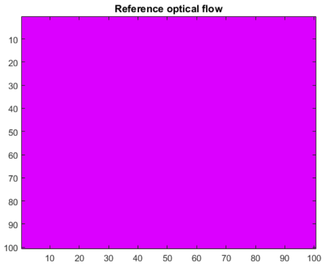


(a) Frame 0 (b) Frame 1

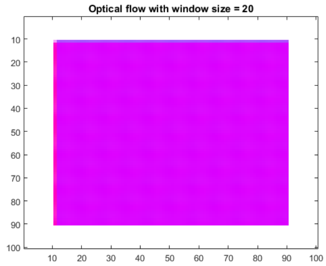


(c) Gradient

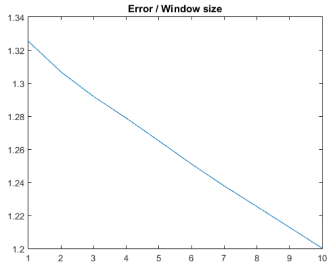
Figure 4: *mysine* input data and gradient



(a) Reference optical flow

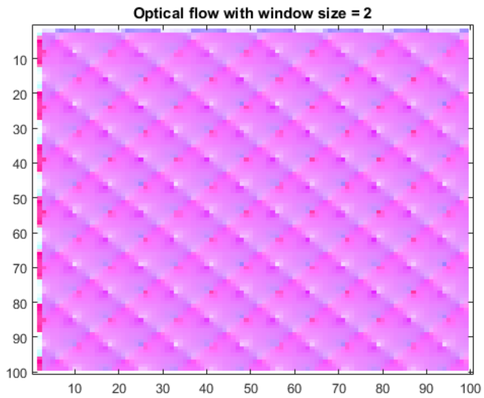


(b) Estimated optical flow with $w = 20$

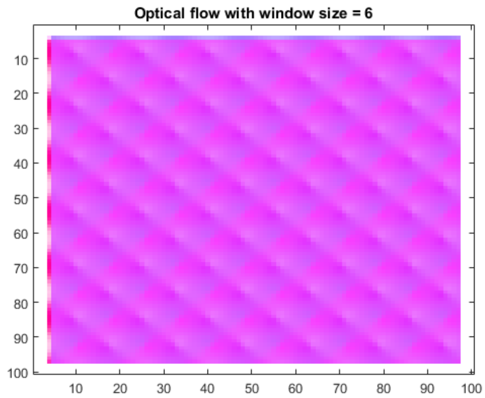


(c) Mean angular error in relation with w

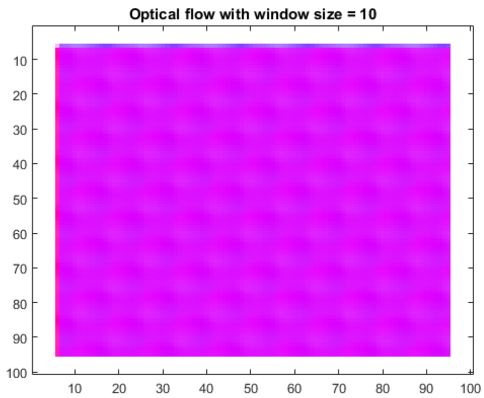
Figure 5: Study of w minimizing the mean angular error on *mysine* data



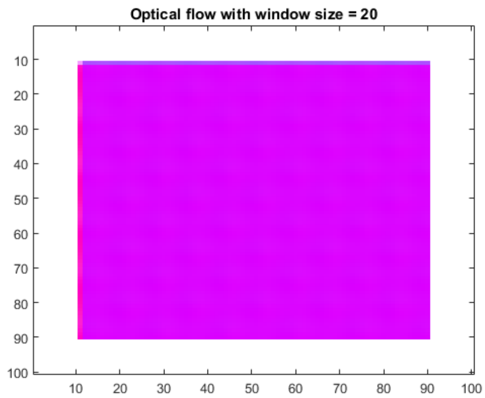
(a) window size = 2



(b) window size = 6

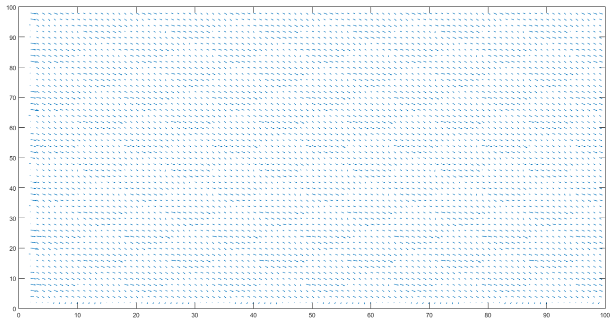


(c) window size = 10

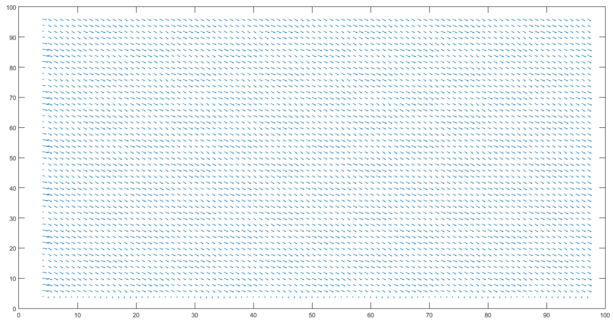


(d) window size = 20

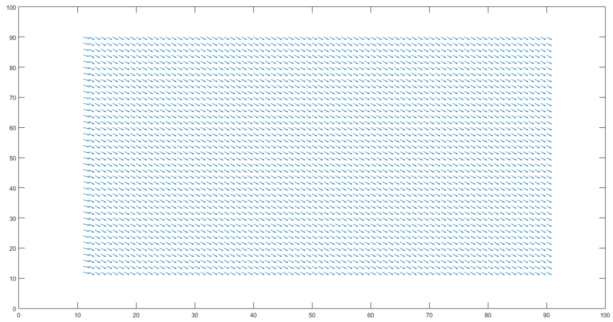
Figure 6: Effect of the size of the window on *mysine* data



(a) $w = 2$



(b) $w = 6$

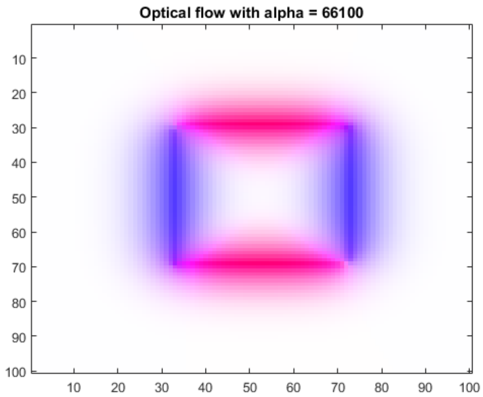


(c) $w = 10$

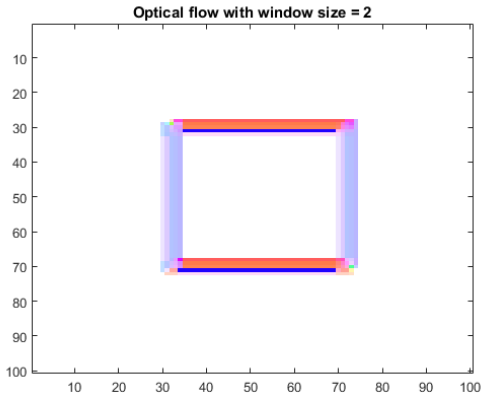
Figure 7: Study of the aperture problem in relation with w on *mysine* data

4 Horn Schunk VS Lucas Kanade

From the results of both algorithms on the *square* data it can be clearly observed the influence of the smoothing parameter of the Horn Schunk method. We can see how the optical flow is propagated, compared to the Lucas Kanade method which gives a sharper result. The vector representation of the optical flow comes to enforce this statement.

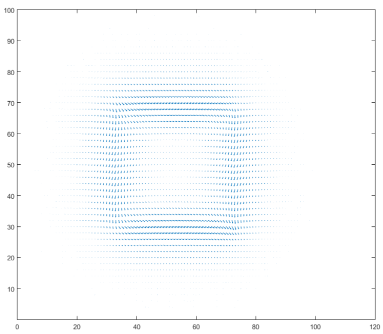


(a) Horn Schunk

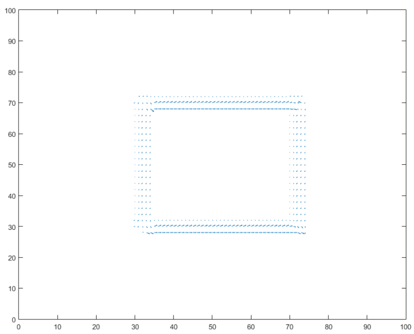


(b) Lucas Kanade

Figure 8: Horn Schunk VS Lucas Kanade on *square* data



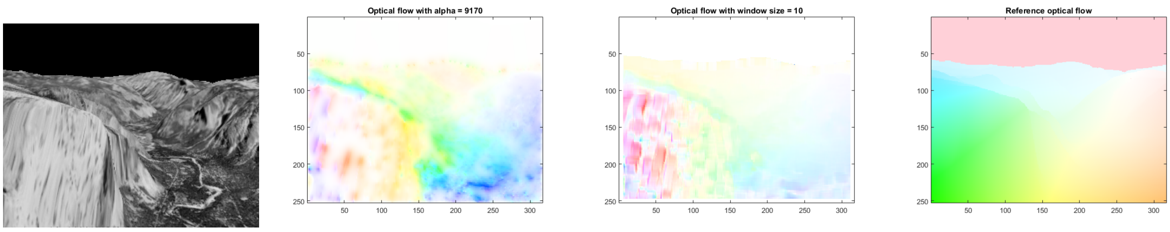
(a) Horn Schunk



(b) Lucas Kanade

Figure 9: Horn Schunk VS Lucas Kanade on *square* data displaying optical flow as vectors

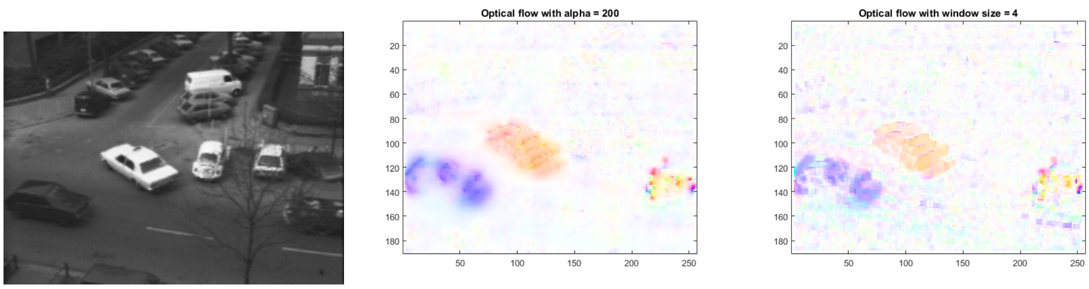
We observe on the *yosemite* data that both algorithms perform similarly on highly textured scenes, assessing that the details of the surface of the mountain in the foreground are in motion, rather than the whole mountain. Horn Schunk takes a very large regularization parameter and Lucas Kanade a large window for approaching the reference optical flow.



(a) Image (b) Horn Schunk (c) Lucas Kanade (d) Reference

Figure 10: Horn Schunk VS Lucas Kanade on *yosemite* data

From the *taxi* frames and the other natural scenes from our test data we can conclude that both algorithms perform similarly and satisfactory. Also, less constraints need to be imposed on the regularization (Horn Schunk) or on the size of the window (Lucas Kanade).



(a) Image (b) Horn Schunk (c) Lucas Kanade

Figure 11: Horn Schunk VS Lucas Kanade on *taxi* data

5 Conclusion

To conclude, both algorithms perform well, but depending on the specificities of the sequence studied, one method can be more suitable than another, especially when there is present the aperture problem. Consequently, the user has to choose the algorithm and its parameters according to the problem formulation.