

# Lab 6

Mattia Giacomello  
I.D. 1210988

Margherita Lazzaretto  
I.D. 1239115

## 1 Goal

The aim of this laboratory is to detect and track a set of objects in a video through feature matching and optical flow methods.

## 2 Procedure

To achieve the desired goal we designed the class `VideoObjectTracking`. This class loads the frames of the input video and the set of objects to track. Two methods are implemented, one for each main task: matching and tracking.

The first one is `matchFeatures`, that detects and computes the SIFT features for each object and for the first video frame. Then it performs matching of the features of each object with the ones in the frame to retrieve the reprojection matrix. This allows to project the objects corners in the frame image and obtain the borders to track. The choice of using SIFT was dictated by the necessity of having a more robust feature description, since some of the objects to track have weak traits that might be mismatched with, for example, the ORB detector. The method takes as optional input the value of the ratio between the maximum accepted distance between descriptors and the minimum computed one. The default value is set to 5, which resulted to be the best choice after some trial and error steps.

The second method is `computeTrackedFrames`, which implements the main task of the laboratory, that is the tracking. Starting from the first frame, from which the features have been extracted using the `matchFeatures` method, it computes the optical flow between two consecutive frames through the Lukas-Kanade method. In this way we obtain the features of the new frame analysed. Comparing them with previous ones it is possible to find the perspective transformation in order to project the borders of the objects onto the new frame. The optical flow is represented by yellow arrows connecting the old position of the keypoints to the updated one. In the Lukas-Kanade method is used a window of size  $11 \times 11$ , which gives the best tradeoff between precision of the results and computational complexity. In fact, this is the smallest size that allows us to obtain, in a relatively small time, a satisfying tracking of the features. Indeed we do not lose any feature during the tracking process.

We noticed that lower sized windows do not guarantee a good tracking. This could be due to the fact that smaller windows make the algorithm less robust to brightness changes and large motions. During the video, in fact, some illumination changes occur, regarding in particular the bottom right book. This influences the behaviour of Lukas-Kanade algorithm, requiring the use of a sufficiently big window.

In Figure 1 are reported some illustrative frames of the tracked video, on which are highlighted the keypoints in yellow and the boundaries of the objects in red.

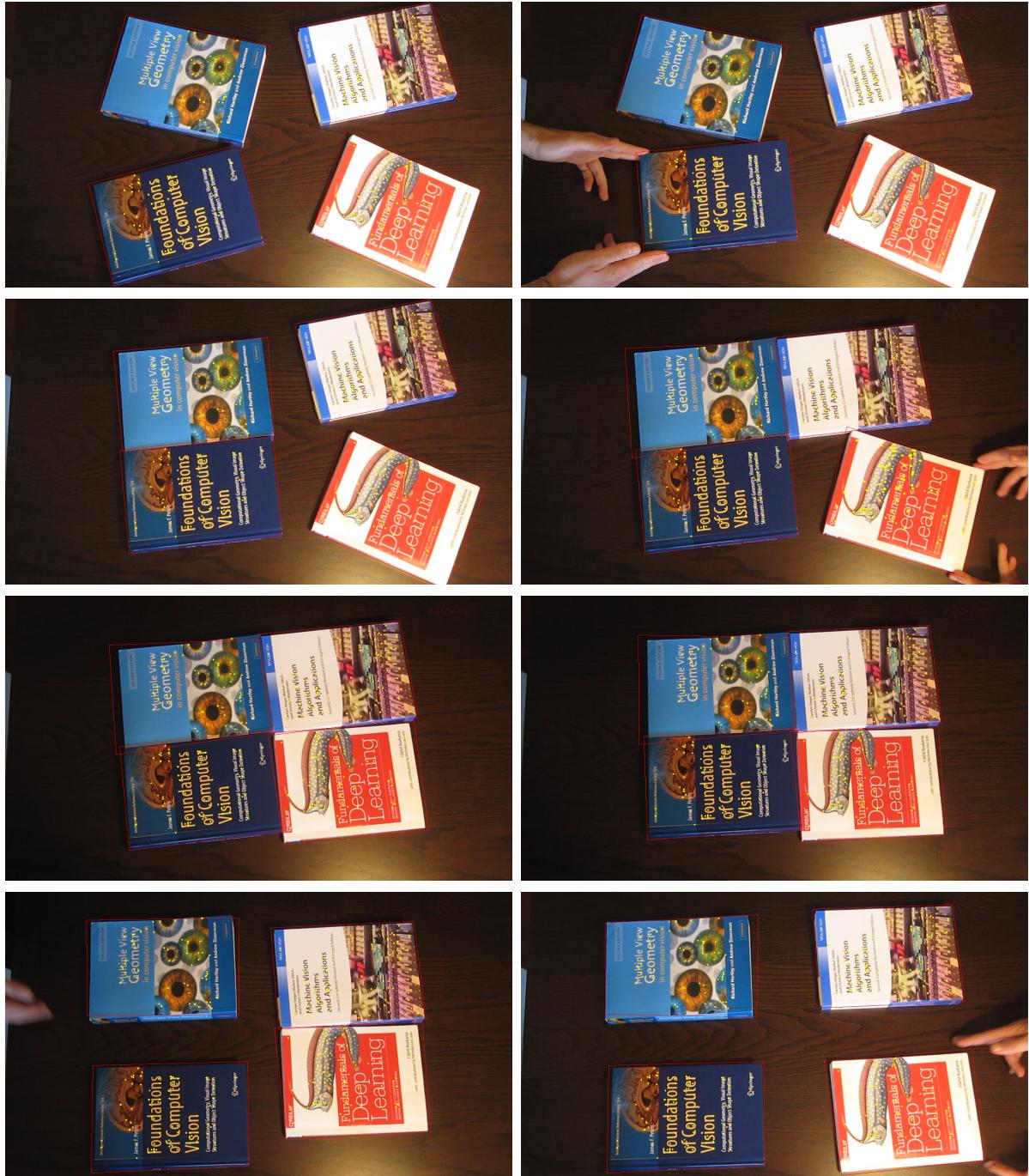


Figure 1: Examples of subsequent frames of the video with tracked keypoints (in yellow) and borders (in red).