

Lab 6 Report – Object recognition and tracking

Structure of the program

The code is organized in two sources files ("HW6.cpp" and "trackingImages.cpp") and one header file ("trackingImages.h"). The first source file contains only the main of the program, while the second the implementation of the class used to manage the aim of the experience.

Command line arguments

The program should be launched using two command line arguments: the first is the path for the input video while the second is the path of the folder containing the objects images. No further inputs are requested to the user.

Process workflow

The process starts by saving each frame of the input video into a vector of images, then it extracts the ORB feature for each object and for the first frame. With this information it computes the matches between each book image and their position in the frame. To better draw the rectangle around the books it's necessary to select the best matches with a minimum distance criterion (with ratio = 2) and with the help of the Ransac algorithm (with threshold = minimum distance).

After performing the feature matching we get into the last part. For each frame and for each input object we compute the new positions of the feature points with `calcOpticalFlowPyrLK()`, we compute the movement with `findHomography()` and with `perspectiveTransform()` we get the new positions of the corners of the surrounding box.

Instead of saving a video with the result, we decided that was easier to show each frame with a small time out in the `waitKey()` function call.

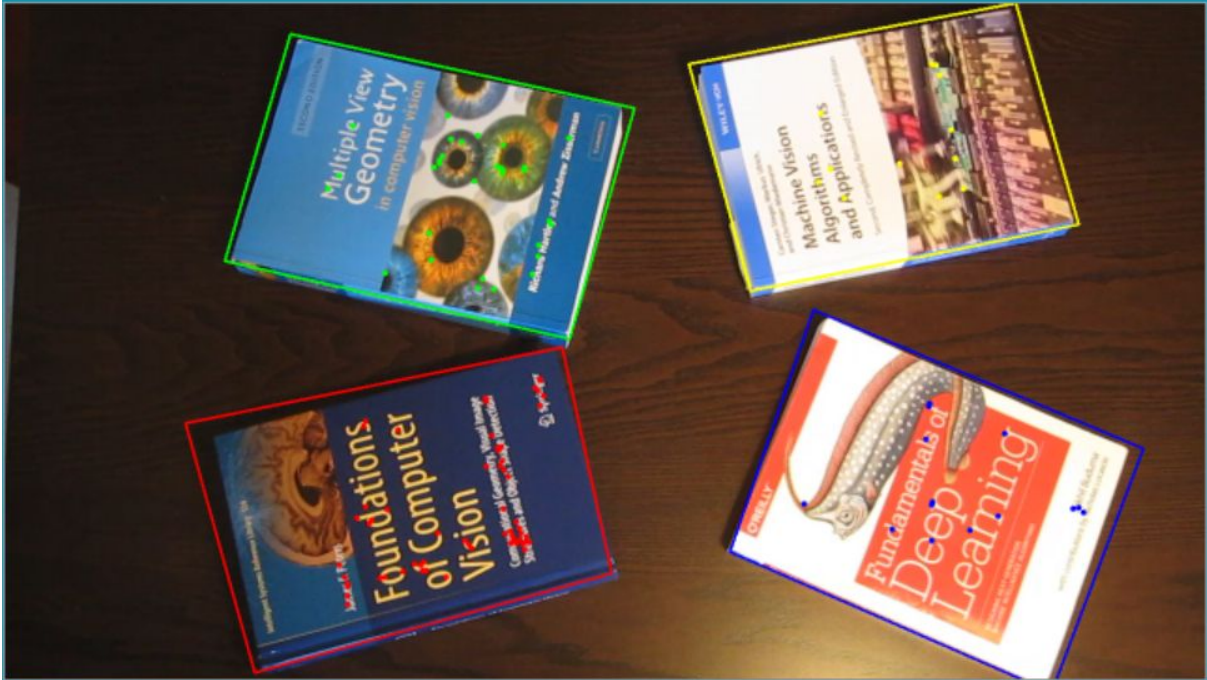
Difficulties and lessons learned

The first difficult found it's about drawing the box around the books. The more problematic is the one named "obj1", because it has too many outliers. To avoid this problem, we follow the suggestion to setup the maximum number of features detected in the object at 5000 and in the first frame of the video at 20000. This operation helps so much to draw the starting boxes in a better way. Then we change the parameter of the ratio for the threshold to adjust the result final.

While tracking the features was pretty straightforward, the boxes started to lose the rectangular form after some iterations, we prevented this as much as we could by modifying the threshold for ransac and the window size in the function that implements Lukas - Kanade algorithm.

In the following images are reported two frames of the computed video with the tracking of the objects.

Video



Video

