Report of Laboratory 6 (Computer Vision)
Oriolo Andrea

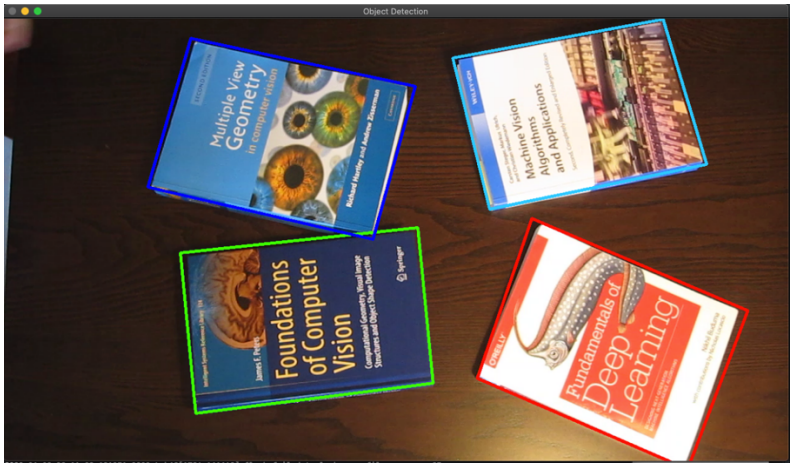
The following report talks about object recognition and tracking to detect and track a set of objects in a video.

The program requires as input the path of the video and of the sample images folder.
First, using the *computeInitialState* method, calculate the initial information by taking the first frame of the video, the sample images and the threshold value to select the best matches to consider. In this method the keypoints and descriptor of the sample objects and the video scene are calculated using the SIFT algorithm; then the matches between the descriptors are calculated through BFMatcher (Brute-force descriptor matcher), which for each descriptor in the first set, finds the closest in the second.
To refine the matches found, only the "best matches" are selected, that is those that have a distance less than the "*ratio * minDistance*" value, which respectively represent the value of the threshold initially given as input and the minimum distance between the calculated matches. In our case, the value that we deemed suitable for the threshold with a better result is 4.
In this way, we calculated the set of inliers through *findHomography* using the RANSAC algorithm, subsequently identified the four angles for each sample image and then used the *perspectiveTransform* method, which transforms a scattered set of 2D or 3D vectors into a floating-point transformation matrix, in order to identify the angles of each object on the video scene.

Calculated the starting points and angles from the video, for each successive pair of frames of the video and for each example object their characteristics are traced by means of pyramid Lukas-Kanade tracker, using the *calcOpticalFlowPyrLK* method which produces an output "status" vector where each element of the vector is set to 1 if the flow for the corresponding features has been found, otherwise, it is set to 0.

Only the features set to 1 are considered so that the transaction is carried out only for these.
The transaction of the features is performed in the same way always through *findHomography* with the RANSAC algorithm and, again, for each frame, through perspectiveTransform the angles of the objects are identified so that the rectangular contour can be calculated using the *drawRect* support method.

The result obtained with the example video is the following:



We also tried with another video and some other objects by not changing any parameters and it works equally well.

The example we tried is the following: