COVIS – Feature detection lab

Deliverables		
Format: Printed report + zipped source code by email.	**Due date**: 24 January	**Page limit**: 6 (not incl. code)
Feedback		
Format: Annotated report.	**Due date**: 7 February	
Evaluation		
Graded		

Instructions for reports

The lab and the corresponding report are to be done in pairs.

Reports should be returned in Mr Vargheese's mail box (building S, mail box 408).

Source code should be returned by email (Anoop.Vargheese@irccyn.ec-nantes.fr). Use your ECN email address to send it. The attached zip file should follow the following naming convention: name1_name2.zip, *e.g.* “newton_laplace.zip”. Also, for the email, use the subject “COVIS Lab 3-4 Submission”.

Goal

To track a given object in a video using OpenCV's feature detection and matching tools.

Object to be tracked

The object to be tracked is the white bulb in figure 1, which is the first frame of the video provided for the lab.

Fig 1. The object to be tracked

Requirements

The tracking will be shown by displaying, on top of each video frame, a rectangle which encloses the bulb. The inclination of the rectangle should be the angle of the bulb with respect to its orientation in the first frame of the video.

The SURF detector is to be used.

Two methods will be tested:

1. By matching the object in current frame to the object in the first frame of the video.
2. By matching the object in current frame to the object in the previous frame of the video.

Tips

- The bounding box of the bulb in the first image is given. The interval in x (image columns) is [24,196] and in y (rows) [44,210].
- The `VideoCapture` class must be used to declare the video input.
- The provided video is in color: you will convert each frame to grey scale.
- The `detect` and `compute` methods are used to perform key point detection and descriptor calculations, respectively.
- The matcher will be chosen as a `FlannBasedMatcher`.
- The `match` method calculates the actual matches.
- To visualize the calculated matches, `drawMatches` can be used.
- To calculate the transformation between two images, use `findHomography`.
- To calculate the images of the corners of a rectangle by the transformation, use `perspectiveTransform`.
- To actually see the displayed images, you need to call the `waitKey` function in your main loop. Otherwise, the `highgui` modules do not get the chance to perform the display actions.

The aim is not to blindly use the methods given above. You must reach of reasonable level of understanding as to what they actually do.

Report

The report should:

- Explain your method, for each of the cases, so the code can be read easily. I should not need to figure out your ideas: you must explain them to me.
- Compare the two methods.

Code

The code must follow good programming practices. It should be included in the PDF **and** delivered as a zip file. Make sure the code can be executed and give the necessary guidelines for the teacher to be able to compile and execute it.

Do not include the video in your submission.