COVIS Lab 2: Feature Detection

1 Introduction

This lab is to implement a simple single-object tracking (SOT) algorithm based on homography. The main idea of this lab is as following

- Starting with an image, called the source frame, the bounding box of an object (the white bulb as in Figure.1) can be manually marked.
- For every subsequent frame, the homography between it and the source frame is identified
- This homography is used to map the bounding box in the source frame to the subsequent frame.

Figure 1: The object to be tracked

This SOT algorithm is presented below

Algorithm 1: Single-object tracking based on homography

```
Manually identify four corners of the bounding box of object of interest in the source
  frame ;
Detect key points and compute their descriptors in the source frame ;
for each subsequent frame do
    Detect key points and compute their descriptors in this frame ;
    Match these newly found key points with those in the source frame ;
    Using the matched key points estimate the homography between this frame the
      source frame ;
    Map the bounding box in the source frame to this frame using the homography
      found in the previous step ;
end
```

2 Implementation

This lab is implemented in a single file `tracking.py`. The **expected work** is to fill into the lines marked **TODO** in this file.

Useful OpenCV functions are listed below. Note that here the arguments of these functions are omitted. This information can be found in `docs.opencv.org/4.4.0/d6/d00/tutorial_py_root.html`

- `cv2.cvtColor()`: to convert an image from a color space to another
- `cv2.ORB_create()`: create an object that can detect key points and compute their descriptors in images. To find key points and descriptors, invoke the method `detectAndCompute` of this object.
- `cv2.findHomography()`: find the homography that maps one set of points on one image to another set of points on another image.
- `cv2.perspectiveTransform()`: apply a homography on a set of points