



## Simple Segmentation and Morphological Operations

**Exercise 20** You will find the image *buoy.jpg* on the exercise website. You need a user name and a password for downloading it. User name: *lme* Password: *inf5*

Display the result after each step using `cv2.imshow()`. The parameters for each function are described in the OpenCV documentation, which you can find online at <http://docs.opencv.org/2.4.11/>.

- (a) Find a way to isolate the red buoy from the background as good as possible. Look at the individual color channels. Hint: you can look at the ratio of colors. Another hint: you should not divide by zero. What can you do instead?
- (b) Create a binary mask, which corresponds to the detected area of the buoy, by thresholding. A binary mask is an image with only one channel. The value 0 corresponds to `false`, all other values to `true`. The result does not need to be perfect. It should contain the general area of the buoy though. You can use `cv2.threshold()`.
- (c) Use a closing operation to close holes in the mask if necessary. Find a good size for the structuring element. Are there potential difficulties in using a closing operation on the buoy? If yes, why and where? You can generate the structuring element by calling `cv2.getStructuringElement()` and apply it by using `cv2.morphologyEx()`.
- (d) Determine the contour of the buoy from the mask. Use `cv2.findContours()` for this. The function returns three arrays `x`, `contours`, `z = cv2.findContours()`. Only `contours` is relevant here. It is an array containing sets of points which lie on the respective outline.
- (e) You may have found several contours due to imperfect segmentation. Assuming that the largest contour belongs to the buoy, find it and draw it on top of the buoy using `cv2.drawContours()`.
- (f) Determine the buoy's bounding box and display only the bounding box's area. The segmented buoy should be displayed on a white background. Use the mask to add the buoy to that background. The bounding box can be determined from the outline result using `cv2.boundingRect()`.
- (g) Compute the buoy's moments using `cv2.moments()`. Compute the moments from the mask.
- (h) Change your python script in such a way that the path to the image file is not hard-coded but given as a parameter to the script. You should be able to the script like this `"python script.py buoy.jpg"` or `"./script.py buoy.jpg"`. Take a look at `sys.argv` for this (<https://docs.python.org/2/library/sys.html#sys.argv>).

**Bonus questions:** What is the meaning of red in nautics and aviation?