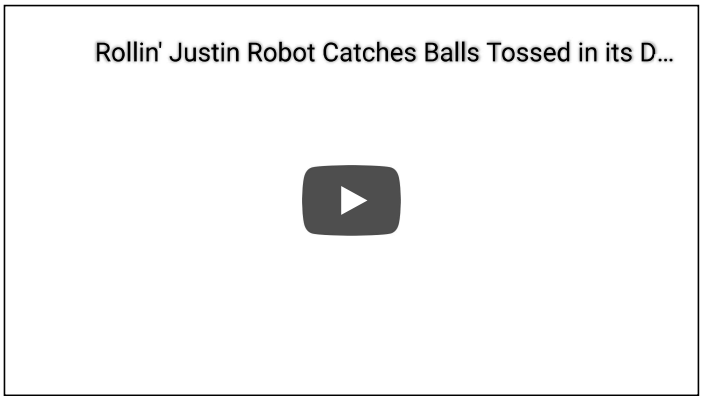
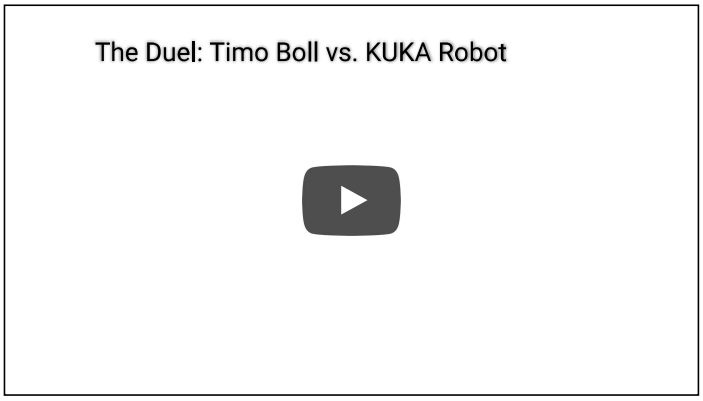
1130-EMARO-MSA-1006 # Computer Vision

T2: Image processing and segmentation.

The goal

Your goal in this exercise is to detect the ball being thrown towards the robot. Based on the knowledge gained in previous tasks, you have to prepare the script that will detect the flying ball and, if possible, calculate its trajectory in 2D (in pixels) or 3D (in meters). Sample robotic systems using similar algorithms can be seen in the following movies.





Input data

As an input, you will get some static images of the ball you are going to detect along with the calibration images for the camera that was used to record them. This data should be used to create the “model” of the ball.

Based on the ball model you have to create an algorithm able to recognize this ball in the image sequence. On the sequence, the ball will be moving with ballistic trajectory (you can treat the throw as a perfect parabolic motion).

Task advancement

Segmenting the ball [2p]

The most important thing is the detection of the ball itself. Your algorithm must be able to detect the ball on both learning images and in testing sequences. The output of this step is the mask of the ball. You can use Color Thresholder App to get a good starting point and then Image Segmenter to finally filter the image.

Calculating the ball parameters (image) [1p]

For the detected ball you have to calculate its parameters in the picture. You have to calculate at least:

- ball position *(x, y)*
- ball radius *r*

You can also calculate the current velocity *(vx, vy)* in the image.

Calculating the ball parameters (world) [1p]

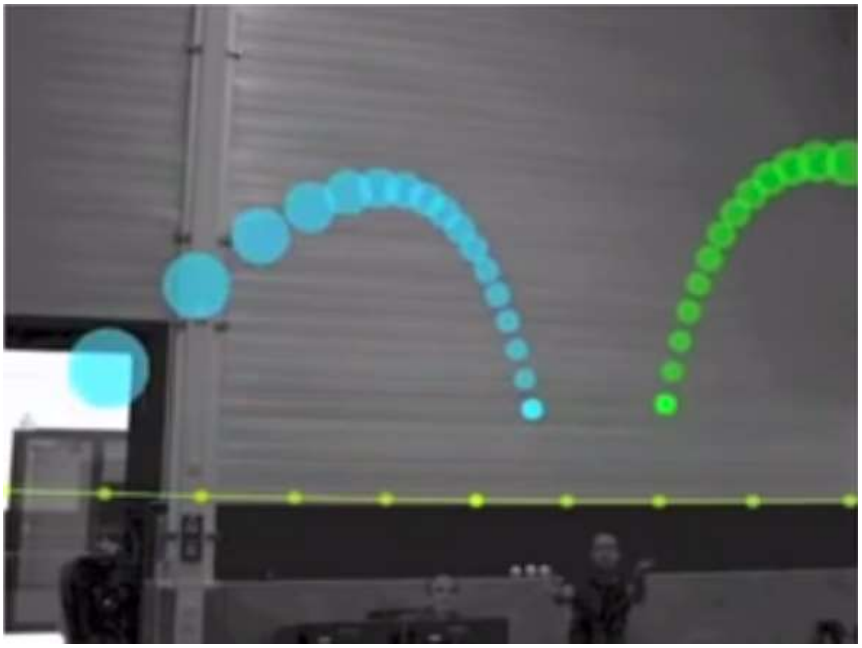
Based on the calculated pixel parameters and known camera model you have to calculate the real-world position *(X, Y, Z)* of the ball. Assume the ball size (diameter) equal to 20 cm.

Trajectory estimation and prediction [2p]

The last step is to calculate the estimated trajectory of the ball (parabolic). For each frame, you have to calculate the trajectory based on the ball detections up so far. For the first few images, the trajectory will be far from the real one, but after a short time predicted trajectory should be consistent with the real ball path. For this task, it is enough to calculate the 2D trajectory based on the pixel positions of the ball (but it will not be a perfect parabola because of perspective). You can also try to calculate the real 3D trajectory (it should be parabolic).

Results visualization [1p]

Try to make the visualization of the results live as the pictures are processed. The final result should look similar to the one shown in the first movie, around 0:40:



Useful functions

There are a lot of methods in computer vision and image processing toolboxes. Some of them, that can be helpful in solving the task, are given below.

- Input image, after loading, will be encoded in RGB color space. For the color-based segmentation, you will probably need to convert your image to HSV or YCbCr colorspace.