Advanced Vision Assignment 2 Report

Mindaugas Dabulskis, Marat Subkhankulov

26th February 2013

1 Introduction

This report describes the work carried out for the second assignment of the AV course. The aim of the assignment was to track three coloured balls through a set of video frames. The algorithm for ball detection and segmentation is discussed. The performance of the approach is evaluated using a gold standard.

2 Algorithm and implementation

In order to segment out the balls from the subject image (the current frame being considered), we removed background regions, and thresholded using values obtained from training data.

The following regions had to be removed from the image:

1. Static background
2. Juggler's clothing
3. Juggler's hands
4. Juggler's face
5. Shadow caused by the juggler on the door

To remove the above regions the following techniques were applied:

1. Mask 1: Average background subtraction
2. Mask 2: Clothing mask based on intensity
3. Mask 3: Hand skin mask based on saturation
4. Classification based on a training sample

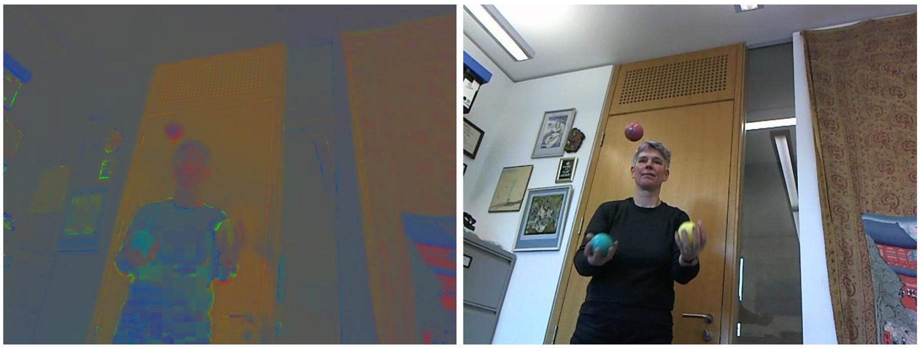

Figure 1: Training sample for each ball

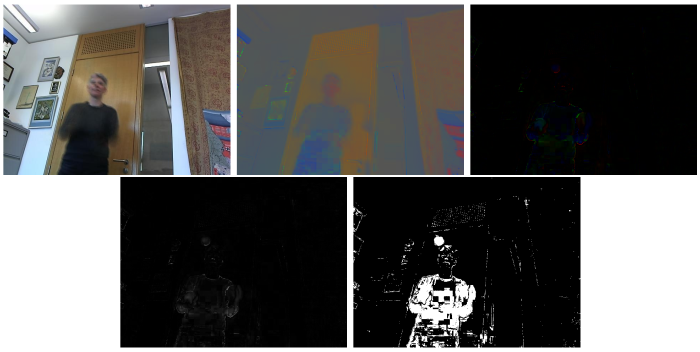

Figure 2: Training sample for each ball

2.1 Mask 1: Average background

A picture of the empty room had been given, however when subtraction of the background from the subject image did not eliminate the shadow of the door, which had similar values to the yellow ball even in normalized RGB. Thus we obtained a mean of all the frames and used the resulting image as the background; By converting the subject and the new background images to N-RGB, subtracting and thresholding out the low brightness regions we obtained a mask which eliminated:

1. Static background
2. Shadow on the door
3. Juggler's face

2.2 Mask 2: Clothing mask

Mask 1 did not remove the clothing of the juggler due to the smoothing effect of averaging; This posed a problem as the clothes had similar chromaticity to the green ball. We took advantage of the clothes' dark colour to easily theshold out the clothing using a manual threshold.

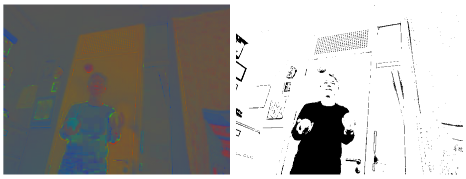

Figure 3: Training sample for each ball

Figure 4: Training sample for each ball

2.3 Mask 3: Hands

It seemed reasonable to use N-RGB for segmenting out the balls as the lighting effects were lare largely removed, however the hands of the juggler were not removed by the previous masks and had similar chromaticity to the red ball. We removed the hands by taking advantage of the skin's unique saturation value.

2.4 Segmentation

Having applied all of the masks the background was largely removed. Thresholding could now be performed. Conversion to N-RGB eliminated diffuse and specular lighting effects on the balls, but necessitated thresholding on all three channels; Since the intensity of all the pixels was largely the same after normalization, it was clear that the majority of variation lies in the hue of the spheres - using this single value simplified thresholding as only one channel could be used to successfully classify the ball pixels.

Threshold values for hue were obtained from a training sample. The training sample obtained by sampling the pixel as the centroid of each ball given by the ground truth. The lower and upper thresholds were 2 standard deviations from the median of the sample - this was done to ignore outliers.

2.5 Clean up

Final clean up and ball identification was made using matlab **bwlabel** and **regionprops** functions. The function **bwlabel** takes our intermediate file, which we got after applying all the masks, and labels all the connected objects in the file. Later, this file is supplied to **regionprops** function, which measures and returns set of the properties for each connected object in the file. We only needed three properties:

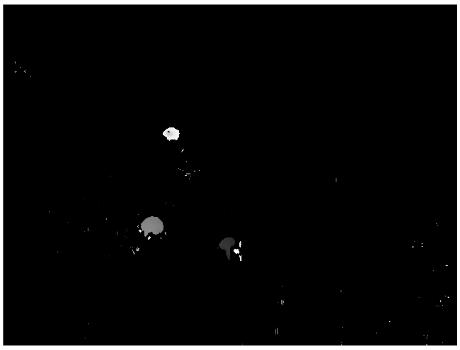

Figure 5: Training sample for each ball

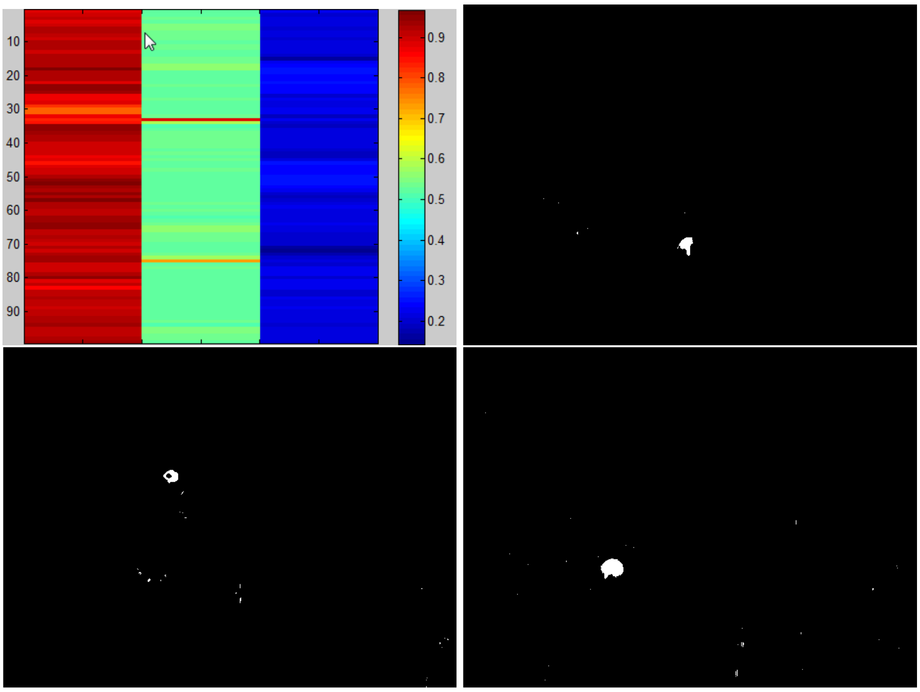

Figure 6: Training sample for each ball

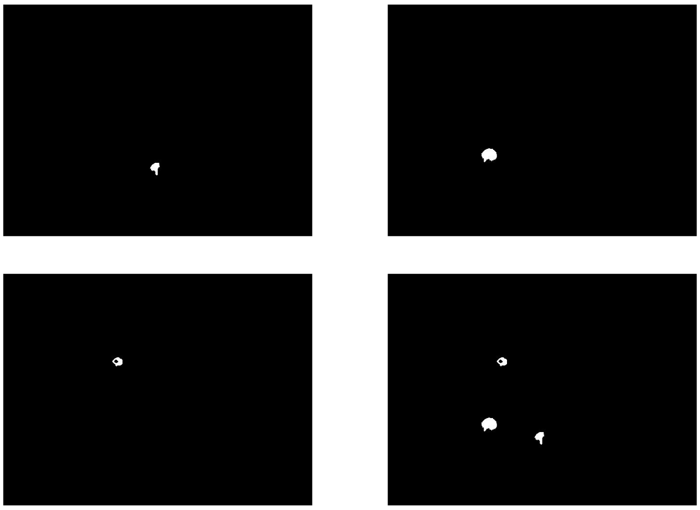

Figure 7: Cleaned images

1. *Area* - number of actual pixels in the region,
2. *PixelIdxList* - vector containing the linear indices of the pixels in region,
3. *PixelList* - matrix specifying the locations of pixels in the region.

The idea was:

1. to find maximum connected object in the file by the pixels area,
2. set all other connected object areas pixel to 0 (delete these areas),
3. find the middle point of the largest connected objects area.

Second point cleaned the image of all the noise, because the object with biggest area was the ball itself. Third point was needed for evaluation, to evaluate how much our detected ball differs from ground truth. We had four different approaches how to calculate the mass of the ball:

1. Use the mean of the area pixels,
2. use the median of the area pixels,
3. use the maximum and minimum pixels in **y** and **x** coordinates and find the mean between them. This creates a bounding box of the area and finds it centre.

The evaluation of these approaches are reported in the section below. We decided to use just normal mean, but left an option to change if needed and found that other approaches with different data sets works better. We stored all found centres in the matrix. The format of that matrix is exactly the same as the format for the ground truth. In next stage we just plot the found mass centres on each image with the centre from ground truth and draw the trajectory at the end by connecting all the points.