

# Visual Computing

## Exercise 1

All programming has to be done in Matlab. Some Matlab commands you might find useful:

- *imread*: reads an image
- *im2double*: converts image entries to double format
- *ginput*: input from mouse
- *aviread*: reads an avifile
- *avifile*: creates a new video file
- *mean*: mean value computation
- *cov*: covariance computation
- *pinv*: Moore-Penrose pseudoinverse of matrix
- *ones*: creates an array filled with 1s
- *zeros*: creates an array filled with 0s

Many loops and if-conditions can be avoided in Matlab. For example

---

```
1 mask = zeros(height,width);
2 for y = 1:height
3     for x = 1:width
4         if (I(y,x) > t)
5             mask(y,x) = 1;
6         end
7     end
8 end
```

---

can be written as

---

```
1 mask = I > t;
```

---

Similarly useful:

- $A(:,1)$  returns all elements of the first column
- $A .* B$  is an elementwise matrix multiplication
- $A(:)$  vectorizes matrix  $A$
- *reshape*: reshapes an array
- *repmat*: replicates and tiles an array
- *squeeze*: removes singleton dimensions
- *find*:  $A(\text{find}(A==2)) = 3$  replaces all entries equal to 2 with 3

## 1.1 Bluescreen

Your matlab functions should load the file *videos/bluescreen.avi*. Ask the user to select one or more background pixels in the first frame to create the background model. To visualize the results, play an image sequence showing the per frame masks.

- Implement a matlab function to extract foreground objects in bluescreen movie footage. The blue background color should be specified by a single color value  $[r, g, b]$ . You may obtain such a value by using the matlab command *ginput* to select a blue image pixel. Read the pixel value and store it as reference value. To determine whether a given pixel should be foreground or background, threshold the absolute distance of its value to the reference value, i.e.  $|[r, g, b] - [r', g', b']| < t$ . Based on pixelwise decisions, a mask can be created to specify foreground and background.  
Try different values for the threshold and the reference color. To test your results, use the short video provided. Note that not the whole background consists of bluescreen. To overcome this problem, use the precomputed mask *videos/mask.bmp* provided.
- To further improve the results, specify the blue background by an exemplar set of blue background pixels. You may select this set using *ginput*. After calculating mean and covariance of the pixel values, the Mahalanobis distance can be used to decide whether a color value originated from the background or the foreground.

## 1.2 Per-pixel Model

Your matlab functions should load the files *videos/jugglingBG.avi* and *videos/jugglingTest.avi*. To visualize the results, play an image sequence showing the per frame masks for *jugglingTest.avi*.

- Probably you have noticed that the approach in the previous task failed in the upper region of the background, which was not covered with blue sheets. To handle more complex scenarios, extend the background model to a per-pixel model. For each pixel, compute the mean and the covariance of its values over a number of frames which do not contain foreground objects. You may then use the Mahalanobis distance to classify pixels in sequences containing foreground objects. You can use the same threshold for all pixels.  
To create the model, use *jugglingBG.avi*. This scene was captured for several frames without foreground objects. Based on the background model, foreground pixels can then be identified and masks can be created for *jugglingTest.avi*.