

# Computer Vision Lab: Segmentation and Background Subtraction

## CSCI 380 Computer Vision

1. For this lab you are going to implement a background subtraction solution for the included video clip.
2. Since we've been mainly working with images up to this point, we'll keep things simple and break down our video into several images. Recall that a video is really a sequence of images that are rapidly displayed one after another. The following matlab code will break your video into multiple image files:

```
mov = VideoReader('YourVideoFileName.wmv');

for i=1:mov.NumberOfFrames
    img = read(mov,i);
    bwImage = rgb2gray(img);
    outputFileName = sprintf('bwImage%d.jpg',i);
    imwrite(bwImage, outputFileName, 'jpg');
end
```

3. You will want to run the above code only once on your video file. This step is typically called preprocessing.
4. There are several different methods available for background subtraction, and we're going to focus on a basic approach. A basic approach is to track the last 10 pixel values of each video frame. If the intensity of the pixel changes either brighter or darker by a set threshold, we will flag this pixel as a foreground pixel. In order to do this, you will need to know the dimensions of the video you will be using and create a matrix of those dimensions with a third dimension of value 10. For example, if the video you are using is 1920x1080, you will want to create a matrix of size 1920x1080x10. A matrix of this size will allow us to keep track of the last 10 values for each pixel.
5. If you're not sure on the size of your video, you can read in one of the image files and check it's dimensions:

```
myTest = imread('bwImage1.jpg');
size(myTest)
ans =
```

```
1080    1920
```

6. The image above is 1920 by 1080, but Matlab flips the horizontal and vertical so it displays it as shown above.

7. Read in the first 10 frames to create a matrix that you will track the last 10 frames viewed. This matrix will be used to calculate the average value at each pixel location.
8. What you want to do next is create a loop that will run through each of the image files that you created. In each of these iterations you will want to do the following:
  - a. Compare each of the pixel values (height by width) to the average value stored in the average matrix. If the absolute value of the difference is greater than some threshold (start with threshold = 20) set the value of that correspond pixel in the newImage to be white (intensity value 255). If it's less than the threshold you will set the value of that corresponding pixel in the newImage to be black (intensity value 0). If the pixel value is white, it is a foreground pixel, if it's set to black, it's a background pixel.
  - b. Update the average matrix by replacing the oldest image in the matrix with the current image.
9. When you've created a new set of images, you'll want to combine them back together again (Please note you must first create a directory called 'modifiedImages').

```
mov = VideoReader('baseball.mp4');
totalImages = 360;
videoFrameRate = mov.FrameRate;
videoHeight = mov.Height;
videoWidth = mov.Width;

% http://www.mathworks.com/help/matlab/ref/videowriterclass.html
% http://www.mathworks.com/help/matlab/examples/convert-between-image-sequences-and-video.html

outputVideo = VideoWriter('myfile.avi');
outputVideo.FrameRate = videoFrameRate;
open(outputVideo);

for i=11:totalImages
    outputFileName = sprintf('modifiedImages\\bwImage%d.jpg', i);
    myImage = imread(outputFileName);
    writeVideo(outputVideo, myImage);
end

close(outputVideo)
```