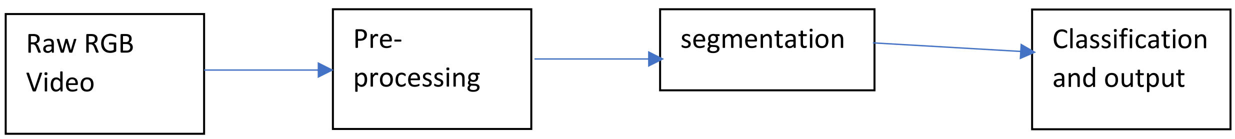
Cade Parkison – U0939163
CS 6640
9/2/2018

Assignment A1: Basic Image Processing Functions

Problem 1

(Report) Sketch the software architecture for an image processing system which analyzes video sequences for moving vehicles. Give the major components, describe their function, describe the interface (i.e., data structures passed and meanings) between components, any files used and the major data structures needed.



In the simple architecture above, a raw RGB video source is the input. The frames of video are sent to a pre-processing step that would do things like transformations, convert to grayscale, blurring operations, etc. The next step is the segmentation part which could be achieved with many different types of algorithms, even machine learning. Once the image has been segmented into semantic regions, this data is passed to a classifier that classifies each pixel as either the car or not the car. This is an oversimplification but it illustrates the main components of these systems.

Problem 2

Acquire a 5-second video using the provided Matlab function, and handin the video as well as the set of individual frames of the video.

Problem 3

(Report/Code) Develop a function to extract the background of a video sequence and describe its performance on the video from problem.

Below is the grayscale background image extracted from my video sequence where a car drives across the field of view. As can be seen, my algorithm did a great job of removing the car from the video and returning just the static background. I ran this on another video where a bus drives by and the performance was not as good. Due to the large size of the bus and the slow speed it was moving, parts of the bus are sill visible in the center of the image.

Figure 1: Background of car passing

Figure 2: Background of bus passing

Problem 4

(Report) Given an image (i.e., one frame) from a video, explore the use of correlation of small sub-images of the image over the whole image to see if you can segment regions with common appearance. Describe the experiments in enough detail so they could be repeated (i.e., state which Matlab functions you used and how; show images of the results).

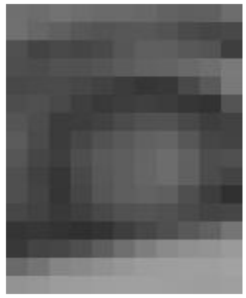
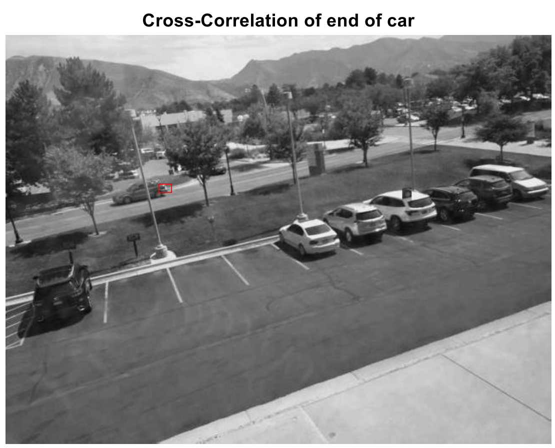


Figure 3: Original Image (left), and Sample Region on moving red car (right)

Starting with the image above (left), I converted it to greyscale for simplification. Then, I selected a small pixel region on the back of the moving car (right). Using this selection, I used the Matlab function xcorr2 to find where the small image fits into the larger image. After finding the maximum cross-correlation, I drew a red box around this location on the original image. As can be seen below, the algorithm was very close to finding the exact region, although the exact pixels have been shifted a bit.

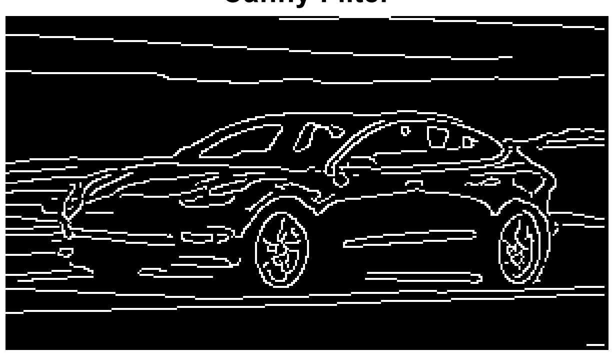


Problem 5

(Report) Given an image with a vehicle (i.e., one frame) from a video, explore the use of differential operators on the image to better understand what structure may be used to extract vehicles. Describe the experiments in enough detail so they could be repeated (i.e., state which Matlab functions you used and how; show images of the results).

I used the above image of a car for testing of differential operators to detect edges. The first thing I tried was Canny edge detection using Matlab's edge() function. When ran on the image above, the following is produced as output.

Canny Filter



While this does not exclusively extract the vehicle edges, it does at least recognize the edges of the vehicle. To further extract only the vehicle, some pre-processing steps could make it perform better. Steps like blurring or applying a Laplacian operator could help here.

Problem 6

(Report) Given an image with a vehicle (i.e., one frame) from a video, explore the use of the Fourier transform on sub-window of the image to better understand what structure may be used to extract vehicles. Describe the experiments in enough detail so they could be repeated (i.e., state which Matlab functions you used and how; show images of the results).

The Fourier transform can also be used to perform correlation, or template matching. First, I used the 2D fast Fourier transform, function fft2(), on both the image and the template to transform them to the frequency domain. Next, I convolved the two matrices and took the real component. The peaks in intensity in the frequency domain of this convolution correspond to the occurrences of the template. To find this location in the original image (spatial not frequency) I found the maximum pixel value at the peaks in the frequency domain and used this value to set a threshold in the image. Any pixel above this threshold should be a pixel in a region that resembles the template.

Problem 7

(Report) Given an image with a vehicle (i.e., one frame) from a video, explore the use of converting from RGB to HSV to better understand what structure may be used to extract vehicles. Describe the experiments in enough detail so they could be repeated (i.e., state which Matlab functions you used and how; show images of the results).

By converting the image of the vehicle from RGB to HSV, it is much easier for an algorithm to detect colors accurately. Using HSV adds robustness to changes in lighting conditions. As an example, two pixels on a red car might have slightly different shades due to lighting. These pixels might have widely different RGB values but their hue might be similar because they are both red.

By thresholding an image by specific HSV values, we can extract all the regions in an image with similar colors, thus allowing us to extract a solid colored vehicle.