Computer Vision

Assignment 1

(shukumar@iu.edu, riteagar@iu.edu, rtawde@iu.edu)

Part 1 - Report

Assumptions

1. The input image is a grayscale image.
2. Images of resolution of and above 512x512 pixels.

Decisions

1. For watermarking the image, we have taken the radius of the circle(r) to be of 100 pixels which is applied on Fourier transform of the image.
2. The alpha (α) value is set to 0.05 because we found it to have least distortion effect on original image after watermark is added.
3. No. of points (l) in circle is set as 220. Same is the no. of bins.
4. Threshold (t) is set to 0.003 while detecting whether watermark is present or not.

Results

- Noise removal works best for the specified image(noise1.png)
- Image watermarking is done as specified with the above selected parameters. It can be seen with the spectrogram of the watermarked images.
- Watermark detection worked for the images tested with 70% true positives while rest being false positives. Here is the table with the correlation coefficient found in test images with two different values of N.

n = 44	0.001434	n = 53	0.001211
	0.001053		0.000971
	0.001016		0.000864
	0.000816		0.000717
	0.000754		0.000548
	0.000738		0.000508
	0.000611		0.000445
	0.000498		0.000413
	0.000481		0.00036
	0.000301		0.000222
	0.000181		0.000186
	1.19E-05		0.000159
	-0.00038		0.000137
	-0.0004		9.38E-05

<u>How to run the code</u>

Part 1.1 Find spectrogram (magnitude of FFT) of the image

Command ./watermark 1.1 input_image.png spectogram_image.png

Part 1.2 Remove noise from the image

Command ./watermark 1.2 noise1.png output_image.png

Output_image.png will show a picture with noise removed.

Part 1.3 Add watermark to the image

Command ./watermark 1.3 input_image.png watermarked_image.png add N

N is the seed value to srand() function. N is an integer. We have tried the code for value N =44,50,53.

Check if watermark is present in the image

Command ./watermark 1.3 input_image.png output_image.png check N

Here we provide the same N value which was used to insert watermark.

Part 2 – report

- Convolution of an image with kernel is implemented and the one with separable kernel takes less time.
- Sobel operator is used to detect edges in an image and the threshold is kept after testing for different values, the one with the highest accuracy is taken as the threshold.

Car detection:

- We have used manually cropped template to detect cars in a parking lot. There are some tricks that are used after analyzing the images carefully.
- We tried by matching templates of a car inside an image and took the most matching ones (based on confidence) but that did not give us a good output since the pixels are randomly distributed for each car based on each shape and size.
- We then calculated the number of pixels inside a template and tried to match that for each image, for the algorithm to recognize a rectangle to be a car.
- It kind of works because there are very few edges except for trees and parking lines that could be detected as cars and even for them it is very less. The algorithm gets the car positions though not accurately and suffers especially for Plaza.png.
- The algorithm works only for the given images as we do not have the template for unknown images and hence the threshold, lower limit and upper limit must be redefined for the new image.