CSCI B 657 Computer Vision

Assignment - 1

Part 2: Detecting Objects

**What we have done?**

Initially, the convolution functions convolve\_general and convolve\_seperable has been written to convolve the image matrix.

**Functions**:

**Convolve\_general**: It takes the input image and a 3X3 mean filter.

**Convolve\_seperable:** It takes the input image and a row Gaussian filter and a column Gaussian filter.

**filter\_creation:** It takes the sigma value and creates the filter based on the sigma value. (Ref: Digital Image Processing Fundamentals Page: 115)

**sobel\_gradient\_filter:** [1 2 1] [-1 0 1]T gradient filter is applied to the image the get the X derivative, and [-1 0 1][1 2 1] ]T is applied to the image to get the Y derivative.

Sobel X gradient Sobel Y gradient

|  |
| --- |
|  |

**gradientMagnitude**: This function takes the X gradient and Y gradient, computes the gradient and return the matrix.



**find\_egdes:** find\_edges take the input image, and generates the gradients in X and Y direction.

Gradient Magnitude is calculated by taking the sqrt of sum of squares of the X gradient and Y gradient.

**Approach 1: Hough transform**

For this first the image is smoothened with the Gaussian filter and the sobel operator in X direction and Y direction is calculated.

Gradient magnitude is changed to binary image with 255 if the value is greater than 150 and 0 otherwise.

Now the edge points are converted to polar coordinates Below is the plot of the accumulator for the points in binary image.

|  |  |
| --- | --- |
| Hough Accumulator | Accumulator Peaks |

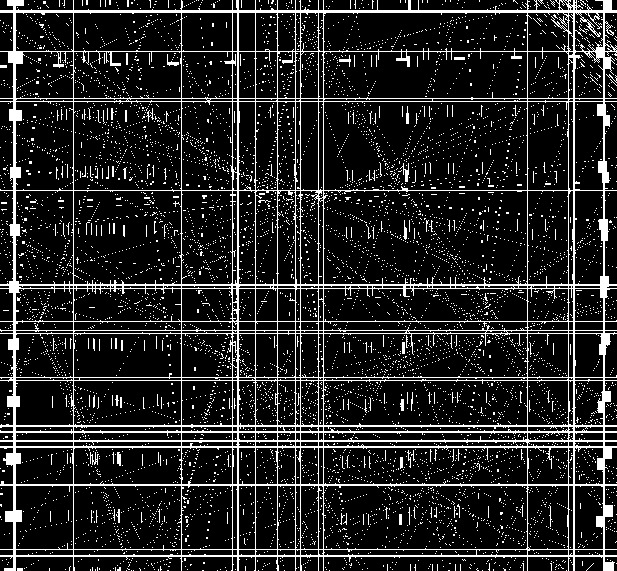
Here the X axis ranges from -90 to 90 and Y axis ranges from –rMax to rMax(max distance from center to image end point)

Non-maximal Suppression:

Here to find the peaks we have implemented the non-maximal suppression with the bin size of 2X2 and picking the local maxima in it.

Here we had a problem to detect the relevant edges from the lines

This is the output of the peaks of the accumulator.



**Approach 2: Sliding Window:**

Since we weren’t successful to determine interest points from Hough transform, we have implemented sliding window approach.

Here, we have used car templates to compare with the original image. A cross correlation is applied to the template and the image part under consideration to find the similarity, here a high value is cross correlation gives us the peak in similarity and return the position of the ICs.

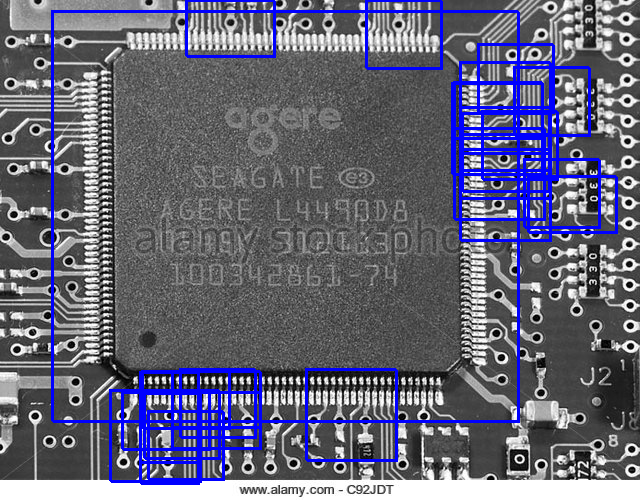
Here the image, the sobel operator was applied in X and Y direction, and the difference in gradient is calculated.

When the template resolution is similar to the image, we don’t compare the image since there will be a less probability of having a large IC.

The problem with the sliding window is that the resolution of the image and the templates, we could see noise in the output image.

Here are results of this approach:

|  |  |
| --- | --- |
|  |  |
|  |  |



**Reference:**

<http://ieeexplore.ieee.org.proxyiub.uits.iu.edu/stamp/stamp.jsp?tp=&arnumber=5375779>

<http://www.cs.utoronto.ca/~fidler/slides/CSC420/lecture17.pdf>

<http://www.ics.uci.edu/~majumder/DIP/classes/EdgeDetect.pdf>

<http://me.umn.edu/courses/me5286/vision/Notes/2015/ME5286-Lecture9.pdf>

<https://pdfs.semanticscholar.org/37c2/eba4ab5959891c45912c8cd8b6b8f0734026.pdf>