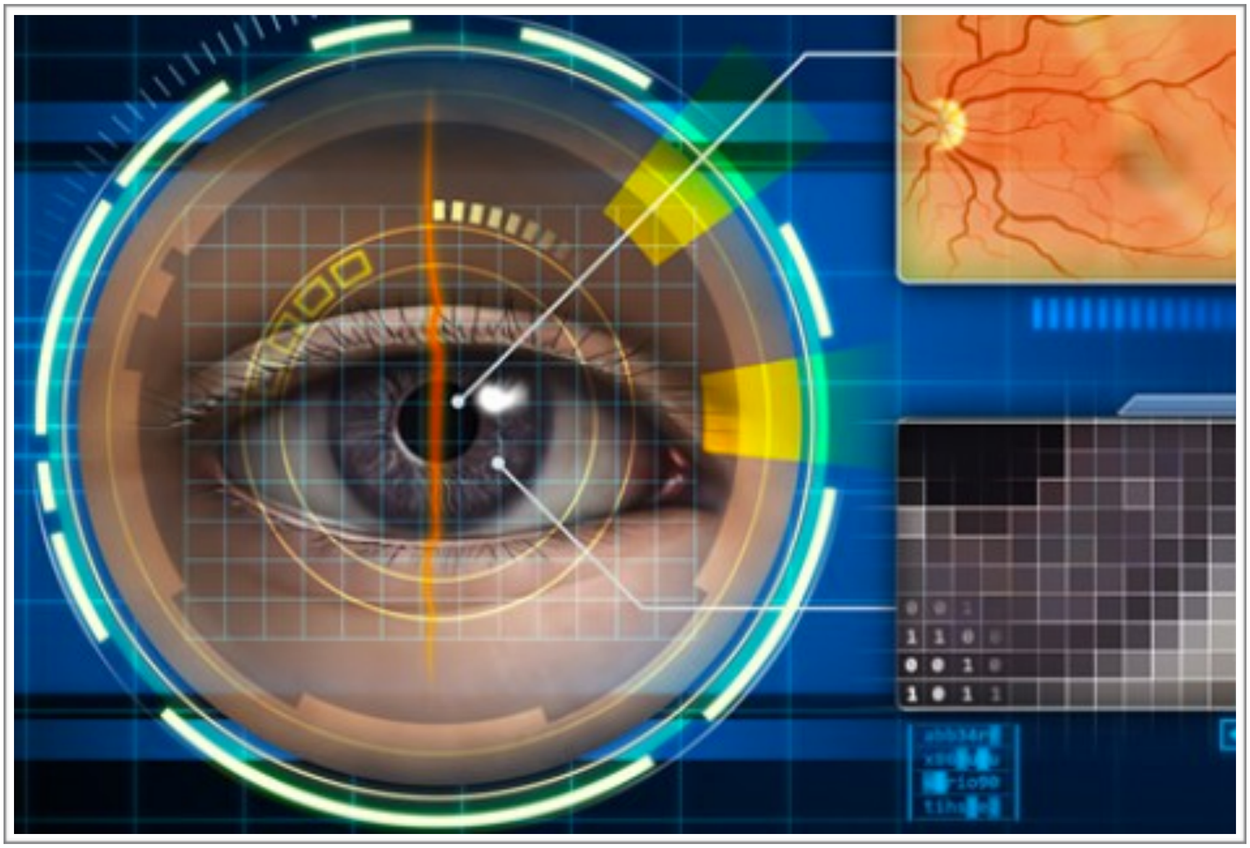
Wavelet based Edge Detection

CSE 573- Computer Vision and Image Processing



Aravind Kumar Ramesh [UBID: aravindk]

Fall- 2015

Submitted on : 14th December 2015

Wavelet Based Edge Detection

using Scale Multiplication in Wavelet Domain

Literature Review

Edge detection is an essential process in image analysis and many techniques have been proposed. Edge Detection is an important process as it can be used to identify boundaries separating one region from the other. In this particular paper by Zhang, Bao [1], a new edge detection by scale multiplication in wavelet domain is proposed. The idea is to multiply two adjacent scales of dyadic wavelet transforms as a product function to magnify the edge structures and suppress noise. **Wavelet Transforms** are very useful. They overcome the shortcomings of Fourier transform as they provide localization in both the frequency and spatial domains.

Scale Multiplication is very effective when it comes to edge detection. This is carried out by taking two wavelets with different scales and multiplying them together. The wavelet transformed image then has two components, which are the horizontal and vertical components. These components are then multiplied with the corresponding vertical and horizontal components in the next scale.
Thresholding is then used to remove small values from edge images. . More often than not, most edge detectors will generate non-zero values for all the image pixels in the edge image. Hence with thresholding, we achieve a better edge map.

Introduction

The project attempts to carry out edge detection using wavelets. It requires the image to be decomposed using wavelet transform for wavelets with 2 different scales and multiplying the corresponding result to find the edge map of the image. Wavelet Transform (WT) is naturally a multi-scale. The dyadic wavelet constructed by Mallet and Zhong is a quadratic spline, which approximates the first derivative of Gaussian. The corresponding dyadic wavelet transform (DWT) is equivalent to

Canny edge detection. Mallet calculated the local maxima of DWT at each scale and formed a multi scale edge representation of the image.
Signals and noise have different singularities mathematically. In DWT domain, it is represented that the edge structures is present observably at each sub band while noise decreases rapidly along the scales. It has been observed that multiplying the adjacent scales could sharpen edges while diluting noise. In the paper by Zhang and Boa they present a scale multiplication based edge detection scheme where the two sub bands are multiplied as a product function. Unlike many multi scale edge detectors, where the edge maps were formed at several scales and then synthesized together, our scheme determines edges as the local maxim in the product function after thresholding . The scale multiplication enhances image structures and surpasses noise. An integrated edge map will be formed efficiently while avoiding the ill-posed edge synthesis process.

Approach to the Project:

NOTE: The following text has been adopted from the original paper [1]

The process of edge detection involves selecting a suitable mother wavelet and scaling of it to produce the father wavelets. This will give us two wavelets with different scales. Then the given image is wavelet transformed using wavelets in two scales. The transformed image of one scale is multiplied with the other transformed image. We do some post processing to get the desired edge map.
The first and the foremost thing in the project is the proper selection of the wavelets. There are many wavelets available and only a few of them can be used for the project execution. This is because few wavelets are continuous and cannot be used in discrete domains, such as for images. Even while choosing a discrete wavelet, we have to consider the fact that the project involves scaling of the mother wavelet which is possible only if the wavelet chosen has a proper scaling function.

The two wavelet transforms used are Haar and DB2, SYM2.

In the paper[1], scale multiplication involves using Gaussian filter differentiated in X and Y direction to find the respective gradients. The Gaussian filter is then scaled for a bigger scale and their respective gradients are found. This gives the edges in X and Y direction found using the Gaussian derivative filter. By using the formula given below, the two wavelet functions for the wavelet is found.

$$\psi^1(x,y) = \frac{\partial\theta(x,y)}{\partial x}, \qquad \psi^2(x,y) = \frac{\partial\theta(x,y)}{\partial y}$$

The given image is filtered using the 2 wavelets calculated above. This gives the $W^1{}_1$ and $W^2{}_1$ for the scale J. Now the other wavelet is constructed using the same way but with a larger Gaussian filter scale. This will yield 2 wavelets which is applied to the image to get the wavelet transformed image $W_2{}^1$ and $W_2{}^2$. Thus the edge in the X and Y direction of the image is found using the wavelets.

$$P_j^{f,1}(x,y) = W_j^1 f(x,y) \cdot W_{j+1}^1 f(x,y)$$

Now after finding the edges in X and Y direction using the 2 scales of the wavelet (i.e. Gaussian filter here) we multiply the edge image in X direction with the corresponding edge image in X direction found using the higher scale. The same is done for Y direction edge image.

Now using the below formula the edge map of the image is found using the scale multiplication process.

$$M_j f(x,y) = \sqrt{P_j^{f,1}(x,y) + P_j^{f,2}(x,y)}$$

Now the obtained edge map M f(x, y) is thresholded and some post processing is performed. One of them is finding the edge point only at the point where edge has the local maximum in the direction of the gradient given by A f(x, y):

$$A_j f(x,y) = \arctan\left(\frac{\operatorname{sgn}(W_j^2 f(x,y)) \cdot \sqrt{P_j^{f,2}(x,y)}}{\operatorname{sgn}(W_j^1 f(x,y)) \cdot \sqrt{P_j^{f,1}(x,y)}}\right)$$

This is process is called the non-maximal suppression. By deleting the points in the edge image whose magnitude is less than the neighboring pixels which are in the direction of the pixel under consideration.

Then proper thresholding is done using the threshold value selected using the following formulae.

$$t_{sc}^{i}(j) = c \cdot \|\psi_j^i\| \cdot \|\psi_{j+1}^i\| \cdot \sigma^2 \cdot (\sigma_{j,+}^i)^2 \tag{25}$$

where c is a constant and

$$\|\psi_j^i\| = \sqrt{\iint (\psi_j^i(x,y))^2 \, \mathrm{d}x \, \mathrm{d}y} \tag{26}$$

$$\sigma_{j,+}^i = \frac{1}{2}\sqrt{\iint (\psi_j^i(x,y)/\|\psi_j^i\| + \psi_j^i(x,y)/\|\psi_j^i\|)^2 \, \mathrm{d}x \, \mathrm{d}y} \tag{27}$$

The equation (28) gives the threshold value to be used to remove the non-significant grey levels in the edge image to get the proper edge map of the input image.

The above process was strictly followed in the project execution in which the wavelet selected is HAAR in one case and DB2 in the other. The wavelet that was constructed is properly scaled to get the scaled wavelets which is needed to perform the wavelet transform based edge detection by scale multiplication.

$$t_{sc}(j) = 0.8 * \sqrt{t_{sc}^1(j) + t_{sc}^2(j)} \tag{28}$$

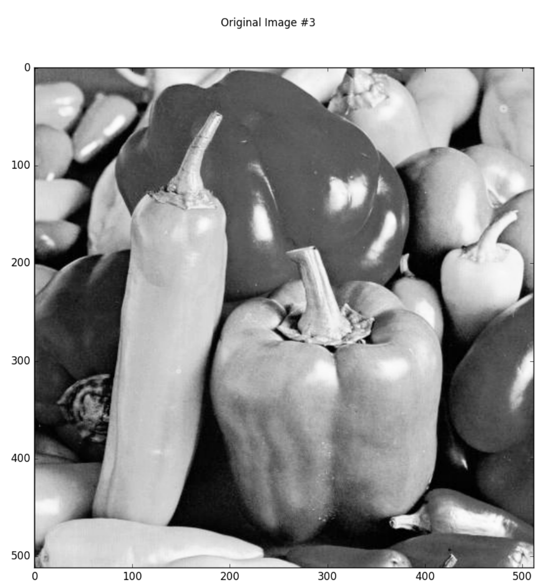
Implementation:

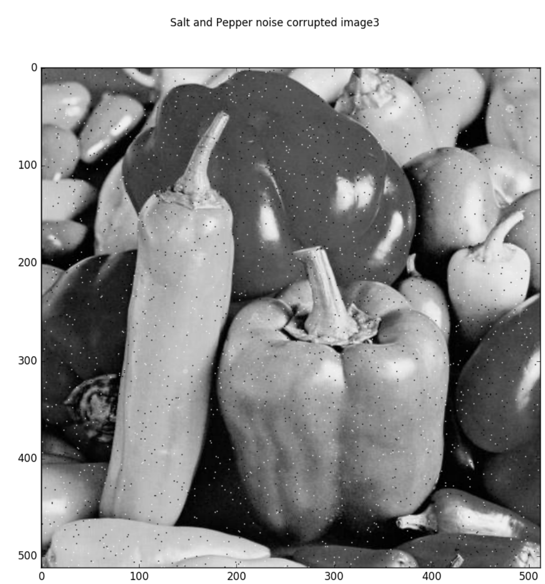
1. Read the test image using OpenCV
2. Apply Gaussian , Salt & Pepper noise to the test image.
3. Wavelet is chosen, using the PyWt module and the filter parameters are computed. This is done to compute the Low pass and high pass decomposition filter coefficients. Implementation was adopted from the MATLAB code base

for the orthfilt function. (Reference : https://github.com/gnattar/main/blob/master/universal/helper_funcs/wavelettool/orthfilt.m)

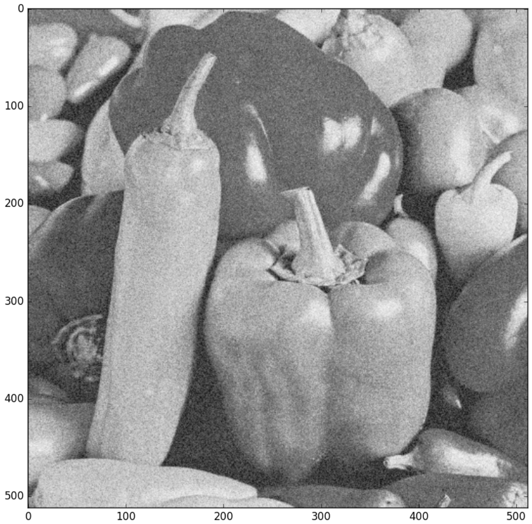
4. With the filter coefficients computed, we apply the decomposition to the image and it's noise corrupted version for 1 level.
5. Now, we do the same for the next level and their corresponding vertical and horizontal components are computed and multiplied together. This will suppress noise and retain edges in the image.
6. The M (x,y) is found which gives the edge map of the image. Thus the edge map of an image is found using the scale multiplication technique of the wavelet transform. Thresholding is finally done.

Result:

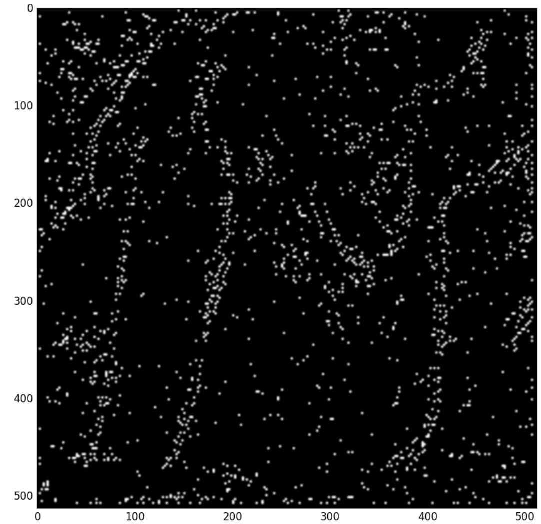




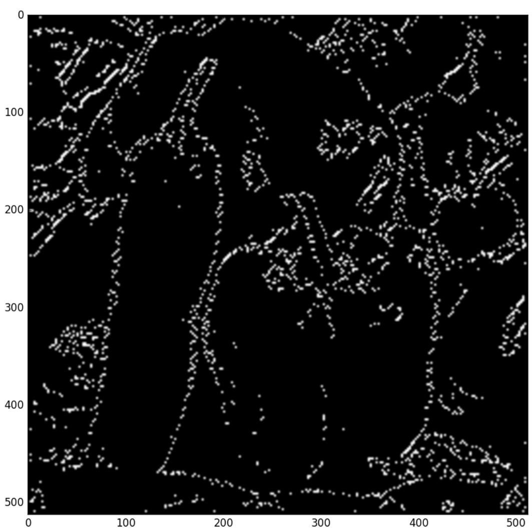
White Noise corrupted image3



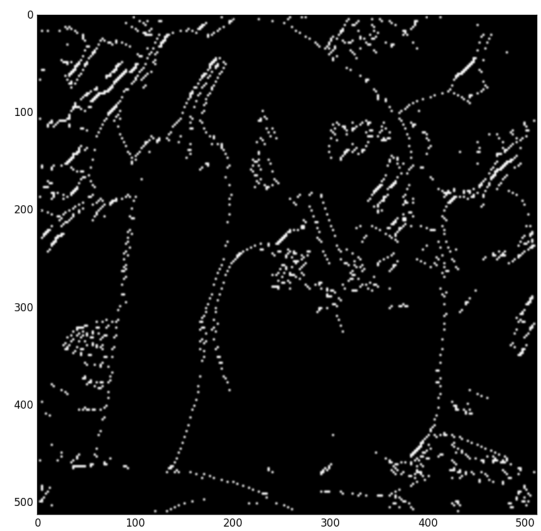
Edge of Gaussian Noise Corrupted Image 3 using db2 wavelet



Edge of Gaussian Noise Corrupted Image 3 using haar wavelet



Edge of Original Image 3 using haar wavelet



Deviation

1. The algorithm developed is efficient. But the output is non-continuous due to thresholding. This can be resolved by applying edge relaxation techniques.
2. The edge of the image even in the presence of the Gaussian or the impulse noise was easily detected using the Scale multiplication technique. The amount of noise in the edge map is drastically reduced due to the scale multiplication performed which removes the noises in the image.

Reference

[1] L. Zhang and P. Bao, "Edge detection by scale multiplication in wavelet domain," Pattern Recognition Letters, Vol. 23, No. 14, pp. 1771-1784, December 2002

[2] http://www.codeproject.com/Articles/21293/Edge-Detection-in-Images-with-Wavelet-Transform

[3] https://github.com/gnattar/main/blob/master/universal/helper_funcs/wavelettool/orthfilt.m

[4] http://www.mathworks.com/discovery/wavelet-transforms.html