Politechnika Wrocławska

Faculty of Pure and Applied Mathematics

Field of study: Applied Mathematics
Specialty: Computational Mathematics

Master Thesis

APPLICATIONS OF WAVELETS IN DATA MINING

Kinga Kurowska

keywords:
wavelets
statistics
learning and adaptive systems

Short summary:
*** To be changed ***
The aim of this thesis is an analysis and comparison of statistical methods employing wavelets. Special attention will be paid to data mining and machine learning methods.

Supervisor	dr inż. Andrzej Giniewicz		
	Title, degree, name and surname	*Grade*	*Signature*

*For the purposes of archival thesis qualified to: **
a) Category A (perpetual files)
b) Category BE 50 (subject to expertise after 50 years)
** Delete as appropriate*

stamp of the faculty

Wrocław, 2018

Contents

Introduction

*** Few words about my thesis... Generally what is a data mining, what are the wavelets, what are the applications in data mining and finally, what is the main purpose - edge detection. ***

Data mining is connected with computing huge amount of data, thus to analyse those data it is require to use algorithms with low computational complexity. Wavelet transform is very efficient.

*** VERY GOOD PARAGRAPH - JUST NEEDS TO BE WRITE WITH DIFFERENT WORDS ***

Real world data sets are usually not directly suitable for performing Data Mining algorithms. They contain noise, missing values and may be inconsistent. In addition, real world data sets tend to be too large and high-dimensional. Wavelets provide a way to estimate the underlying function from the data. With the vanishing moment property of wavelets, we know that only some wavelet coefficients are significant in most cases. By retaining selective wavelet coefficients, wavelet transform could then be applied to denoising and dimensionality reduction. Moreover, since wavelet coefficients are generally decorrelated, we could transform the original data into wavelet domain and then carry out Data Mining tasks.

Pre-processing:

Denoising signals and images. Wavelet denoising - transform data into wavelet domain, remove noise components (with lower frequency) and then back to the original domain.

Data Transformation?

Dimensionality reduction. The idea is to simplify data, by getting rid off the less relevant information. In a wavelet domain we retain only the largest coefficients. Then, after getting back to the original domain we obtain simplified data.

Machine learning processing:

Clustering. Low frequency parts are correlated with regions of objects concentration and the high frequency parts correspond to the areas with sudden changes in the objects distribution. Thus, clustering can be conducted by recognising correlated components in the wavelet domain.

Classification. Regression. Distributed Data Mining. Similarity Search/Indexing. Approximate Query Processing. Traffic Modelling.

Chapter 1

Wavelets theory

A "wavelet" literally means a small wave. This term says a lot about its nature. Wavelets are a family of functions which oscillates like wave and should be compactly supported. Additionally, the wavelet has zero mean.

Definition 1.1. Wavelets are created by scaling and shifting of the, so called, mother wavelet $\psi(t)$. The child wavelets are defined as

$$\psi^{(a,b)}(t) = |a|^{-\frac{1}{2}}\psi\left(\frac{t-b}{a}\right),\ a > 0, \tag{1.1}$$

where a is a scale parameter and b translation parameter.

There are plenty of different mother wavelets, for example

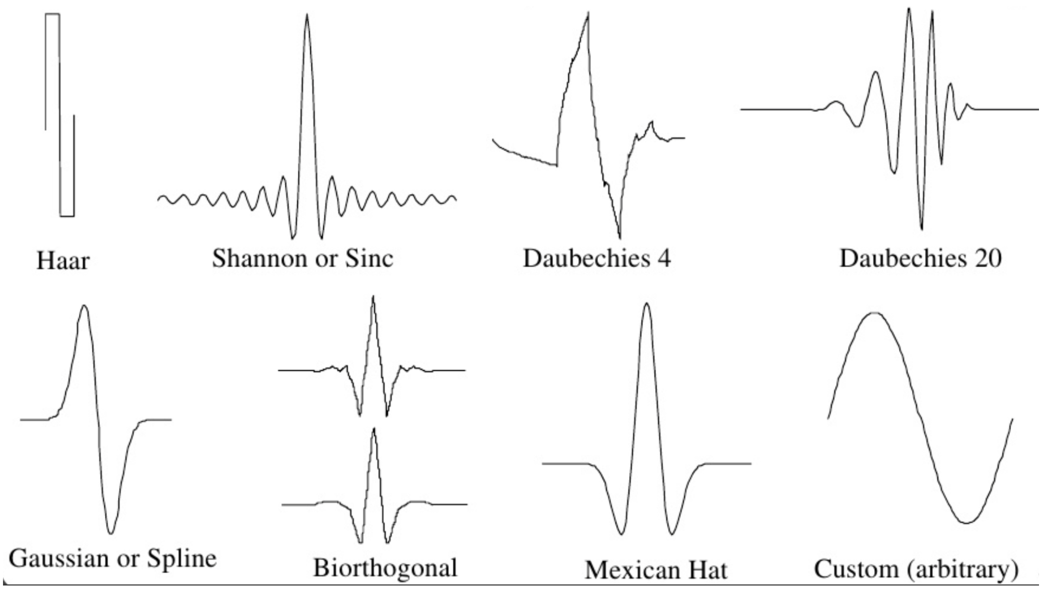


Figure 1.1: Different types of wavelets.

*** Add more about wavelets properties! ***

1.1. Daubechies wavelets

Each type of wavelet function is more suitable for different applications. The best for image analysis are the Daubechies wavelets.

Definition 1.2. Daubechies wavelets are collection of orthogonal and compactly supported functions. A denotation for those wavelets is dbN, where N means a maximal number of vanishing moments.

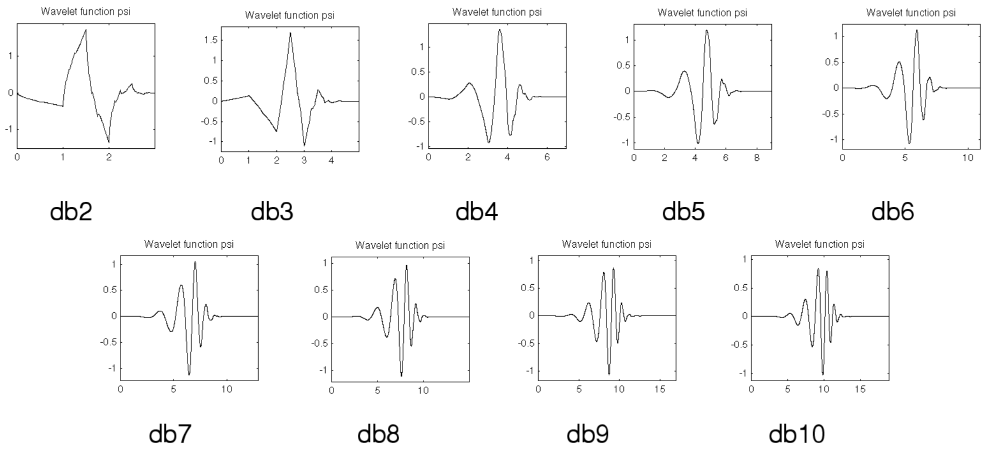


Figure 1.2: Daubechies wavelets.

1.2. Wavelet Transform

*** Add some short introduction! -¿ po co to, co daje, do czego i dlaczego o tym piszemy ***

Definition 1.3. Wavelet transform

$$W(a,b) = \int_{-\infty}^{\infty} y(t) a^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) dt, \tag{1.2}$$

where a is scale parameter, b translation parameter and $y(t)$ original signal.

1.2.1. Wavelet transform vs Fourier transform

*** General information about integral transforms ***

Definition 1.4. Fourier transform

$$Y(f) = \int_{-\infty}^{\infty} y(t) e^{-i\omega t} dt, \tag{1.3}$$

where $y(t)$ is time domain signal and $Y(f)$ is frequency domain signal.

What differs both transformations is the type of function. In Fourier case there are sine and cosine functions, wherein wavelet transform uses wavelets. Why use the Wavelet transform? Sine function oscillates on the whole real axis, thus it cannot represent abrupt changes. However, the Wavelet transform is localized in space and time, so it can be used to detect trends or sudden changes in signals and images.
Moreover, wide range of wavelet functions is a main advantage of wavelet analysis.

1.3. Discrete Wavelet transform

There are two types of the wavelet transform:

Wavelet transform	**Fourier transform**
Suitable for stationary and non- -stationary signals	Suitable for stationary signals
High time and frequency resolution	Zero time resolution and very high frequency resolution
Very suitable for studying the local behaviours of the signal	No suitable
Scaled and translated mother wavelets	Sine and cosine waves

Table 1.1: Differences between Wavelet Transform and Fourier Transform.

- Continuous Wavelet Transform (CWT),
- Discrete Wavelet Transform (DWT).

DWT is used to denoising and compression of signals and images. Also, DWT allows to detect smooth regions interrupted by edges or abrupt changes in contrast of images.
Scale and translation parameters are defined as

$$a = 2^j \text{ and } b = 2^j k, \ \ j, k = 1, 2, \ldots . \tag{1.4}$$

to avoid redundancy in coefficients.
The figure 1.3 on a page 9 shows how DWT works. Discrete Wavelet Transform splits signal with two filters: $g(n)$ - low pass filter (LPF) and $h(n)$ - high pass filter (HPF). The LPF captures a part with lower frequencies which refers to the main signal. Whereas, the HPF captures higher frequencies - a noise of the signal. Subsequently, both parts are downsampled by a factor of 2. This decomposition can be repeated on the LPF part of the signal. Hence, the next levels of DWT coefficients.

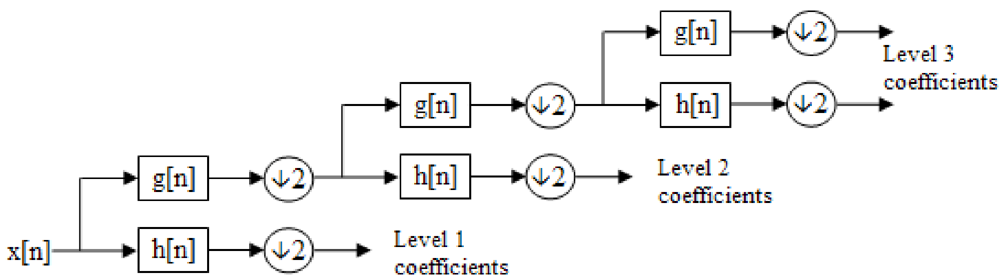


Figure 1.3: Discrete Wavelet transform on a signal $x(n)$.

1.4. 2-D Discrete Wavelet transform

2-Dimensional Discrete Wavelet Transform works similar way as 1-D with High Pass Filter, Low Pass Filter and downsampling, except that one level of the decomposition includes double filtering, on columns and rows. The figure 1.4 shows an image decomposition. Firstly, the DWT is applied on columns of the input image and then on the rows of the both outputs. Ultimately, there are four results:

- LL - result of LPF applied on both, columns and rows,
- LH - result of LPF applied on columns and HPF on rows,
- HL - result of HPF applied on columns and LPF on rows,
- HH - result of HPF applied on both, columns and rows.

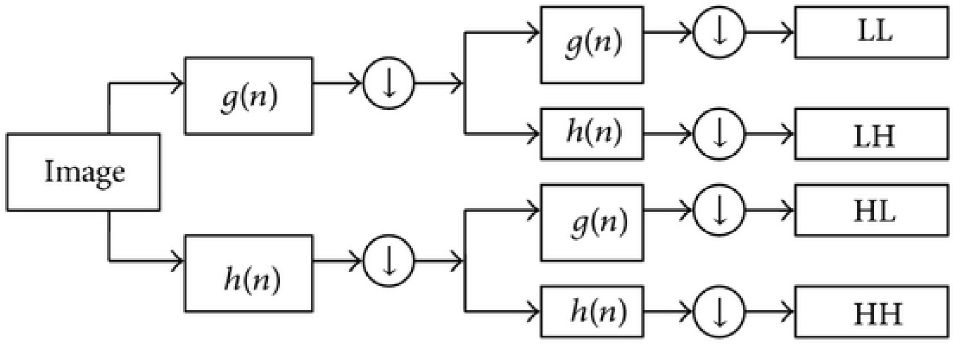


Figure 1.4: 2-D Discrete Wavelet transform on an image.

Recall that the outcome of Low Pass Filter in the previous case was the main signal (without a noise). Thus, a 2-Dimensional equivalent is an approximation of an analysed image. The High Pass Filter captures high frequencies, then for an image the outcome are sudden changes in the image contrast. Now, lets focus on what exactly each result represent. First one, the LL is just an approximation of the initial image. Next, the LH shows abrupt changes in a horizontal direction, whereas the HL part presents similar issues but in a vertical direction. The HH shows sudden changes in a diagonal direction. In conclusion, the output of 2-D DWT gives us an approximation of the image and three parts with abrupt changes in different directions.
The Discrete Wavelet Transform has a wide range of applications in image processing, for example denoising, compressing
*** What are the possible applications of 2-D DWT - inne zastosowania w machine learning i data mining ***
What information gives us these sudden variations? Thanks to those we are able to find a places where two smooth regions meets. This kind of image anomaly could be interpreted as edges.

1.5. Inverse Wavelet Transform

Chapter 2

Edge detection

In this chapter let us focus on the main goal of the thesis, which is edge detection. What is an edge? It is a place where image brightness changes rapidly. There are various methods to identify such discontinuities. The most popular are gradient based (e.g. Canny, Prewitt, Sobel) and Laplacian based. However, there is also another method, which provides similar results and can be more efficient in terms of computation. This method is based on the 2-Dimensional Discrete Wavelet Transform described in section 1.4.

What should be done? *** Link to article Edge detection ***
1. Convert image to grey scale.
2. Apply 2-D DWT on an image.
3. Remove the LL part.
4. Denoise the LH, HL and HH coefficients.
5. Reconstruct the initial image.
6. Post-processing - modify contrast to emphasize obtained edges.

2.1. Implementation of an algorithm

The algorithm is implemented in `Python` using library/package (?) `PyWavelets`. There are used also auxiliary libraries like: `numpy`, `matplotlib`, `PIL` and `scipy`. The `PyWavelets` package contains all features required to edge detection algorithm, i.e. 2D Forward and Inverse Discrete Wavelet Transform, build-in many wavelet functions and thresholding functionality (used to denoise coefficients).
Lets go deeper into algorithm, using a simple example - white square on a black background.

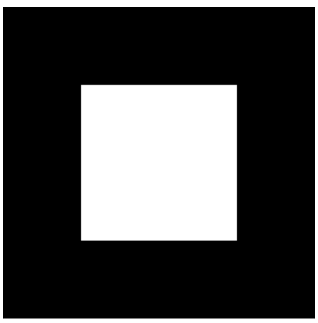


Figure 2.1: An initial image - a white square.

Initially, a colour image must be simplified by conversion to grey scale. Edges are recognised as changes in brightness, so a single pixel should contains only information about a colour (black) intensity. Our initial image is already black and white so we do not need to convert it. Now, we can apply the 2D DWT function. As a result, according to the description in section 1.4, we obtain four components of wavelet coefficients: LL, LH, HL and HH show on a graph 2.2. It is clearly visible that the LL part reflects the initial image and the rest components contains information about rapid brightness changes in particular directions.

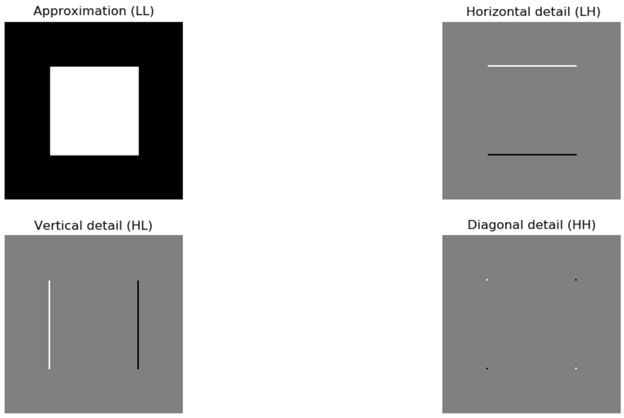


Figure 2.2: 2-D DWT coefficients.

Thus, we are interested only in components which gives information about edges, so we remove the approximation part - figure 2.3.

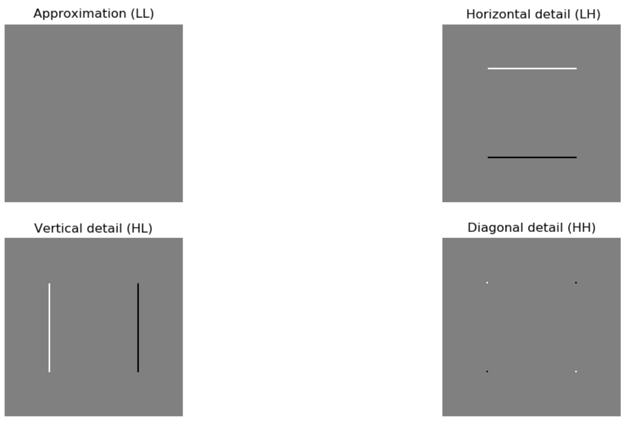


Figure 2.3: 2-D DWT coefficients with removed the LL part.

Subsequently, remaining components can be denoised. It means, we can set rid off small, insignificant coefficients by thresholding. More about setting

the threshold is described in section 2.2. This simple example do not require any denoising, so lets go further.
The last main step is reconstruction of the initial image, i.e. application an inverse 2-D DWT on the denoised coefficients. In the result, we obtain the edges of the initial image presented on a graph 2.4.

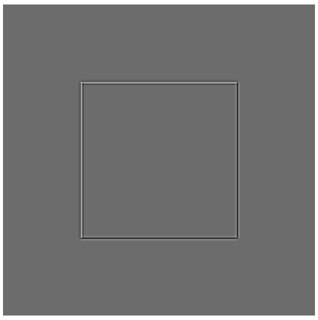


Figure 2.4: A reconstructed image showing the edges.

At the end, we can do some post-processing to emphasize obtained lines. Currently, a background has grey colour and the edges are white or black. Therefore, using simple mathematical calculations we can modify image to have black background and white edges. It is enough to get an absolute value, subtract 128 and then scale by multiplying 2 times. *** Add some more about values in an image array, here or in the previous paragraph *** Finally, we obtain the edges of the square shown on a figure 2.5.

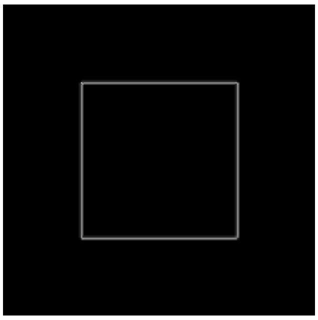


Figure 2.5: The edges of the square image after post-processing.

2.2. Thresholding

There are two types of thresholding hard and soft. Lets denote λ as threshold value and d as wavelet coefficient. The hard one is defined as follow

$$D^H(d|\lambda) = \begin{cases} 0, & \text{for } |d| \leq \lambda, \\ d, & \text{for } |d| > \lambda. \end{cases} \tag{2.1}$$

This thresholding assigns zero value for coefficients below the set threshold. The soft thresholding works in the same way on the coefficients smaller than

λ, but additionally the coefficient bigger than the set threshold are "shrinked" towards zero, as it is defined below.

$$D^S(d|\lambda) = \begin{cases} 0, & \text{for } |d| \leq \lambda, \\ d - \lambda, & \text{for } d > \lambda, \\ d + \lambda, & \text{for } d < -\lambda. \end{cases} \tag{2.2}$$

2.2.1. Hard or soft thresholding

To see the differences between both types of thresholding lets consider a noisy image (fig. 2.6).

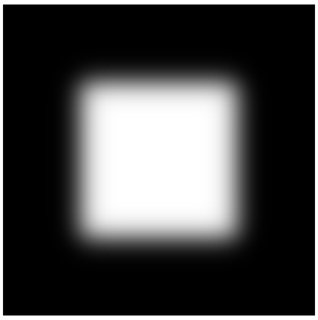


Figure 2.6: A noisy square image.

The coefficients of 2-D DWT are shown on the graph 2.7.

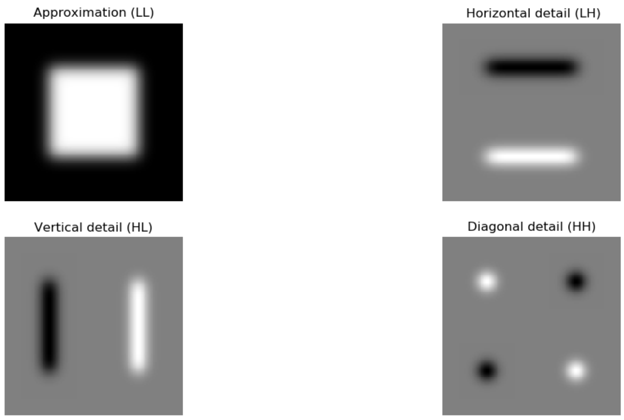


Figure 2.7: 2-D DWT coefficients.

Now, according to the edge detection algorithm, we remove the LL component and we can denoise the others. Results of hard and soft thresholding are presented respectively on a figures 2.8 and 2.9.
Especially on the LH and HL components we can see the distinction between both thresholding. After soft one the lines are smoother and slightly narrower. Therefore, the soft thresholding is better for edge detection (smoother

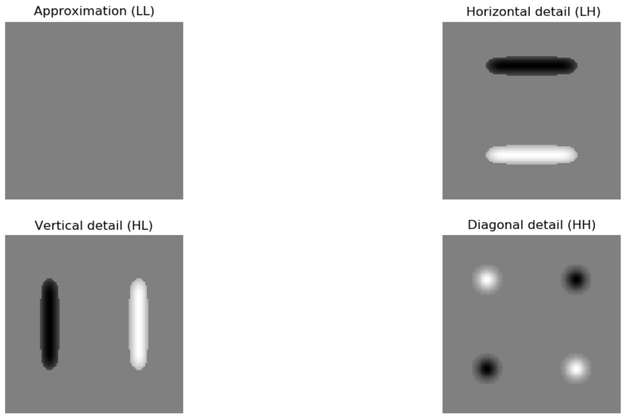


Figure 2.8: 2-D DWT coefficients after hard thresholding.

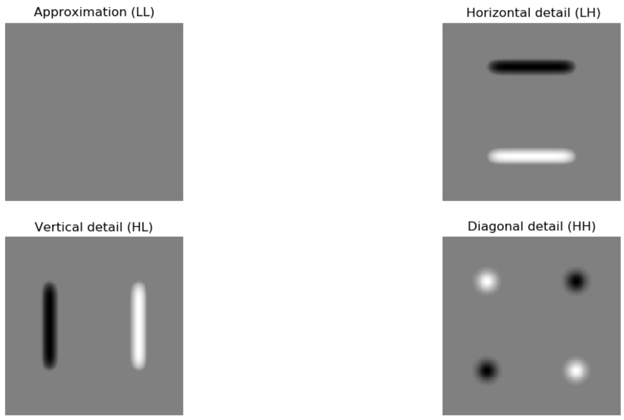
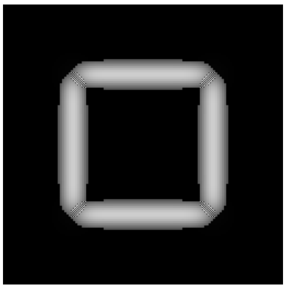


Figure 2.9: 2-D DWT coefficients after soft thresholding.

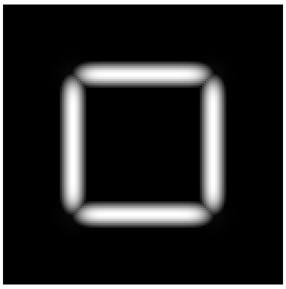
edges are better, /wybacza bledy/zaszumienie bardziej niz hard/). To confirm this conclusion lets compare the final results, after inverse DWT (fig. 2.10)

2.2.2. Setting the threshold

The corollary from the previous subsection is that the soft thresholding provides better results for edge detection. However, there is also a question how to choose the value of a threshold λ to denoise coefficients and do not loose any significant information. A simple idea is to get a quantile of the coefficients in the specific components.
*** Histograms of the coefficients? ***
We decided to start with a quantile on level 0.95. It occurs to be accurate for the majority of standard images. All the earlier results are computed for



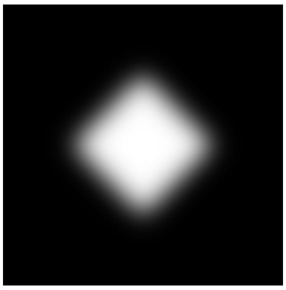
(a) hard



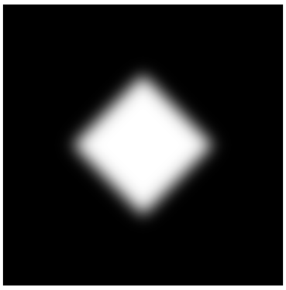
(b) soft

Figure 2.10: Results of edge detection with hard and soft thresholding.

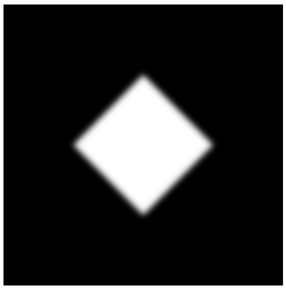
threshold set as 0.95 quantile. Although, images which are more noisy could require changing the quantile level.
Lets consider three images of basic diamond shape with different noisiness, shown on the figure 2.11.



(a)



(b)



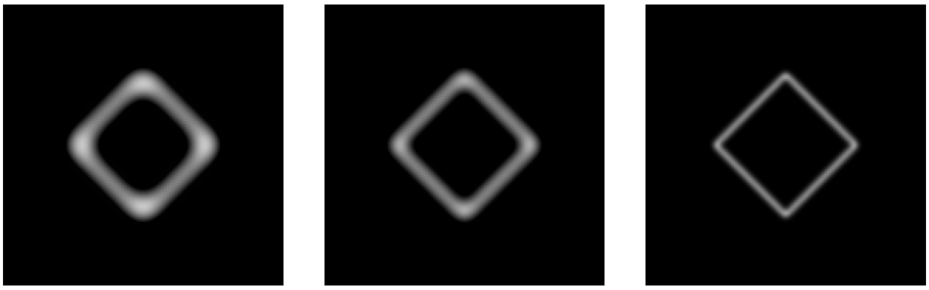
(c)

Figure 2.11: The initial images of diamond with different noise level.

At first, see how looks the result with mentioned earlier threshold set as 0.95 quantile (fig. 2.12). Algorithm found the edges properly but they are quite wide, especially for the noisier image. Lets try to increase the threshold to quantile on the 0.99 level. The results are presented on the graph 2.13. It can be seen that for (b) and (c) images the edges are narrower and visible. Nevertheless, result for the (a) image is incorrect, threshold was too high and too much information was removed. Hence, lets consider one more threshold on a 0.98 quantile level. The recognised edges for (a) and (b) images are presented on the figure 2.14. For this threshold edges are correctly detected. In conclusion, for noisier images the increased threshold provides more accurate result, but increasing the threshold must be done carefully to not ignore crucial informations.

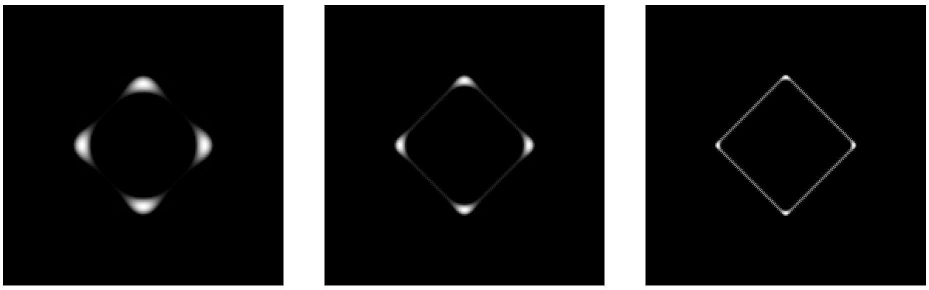
2.3. Selection of a wavelet type

*** Show results for different wavelet types ***



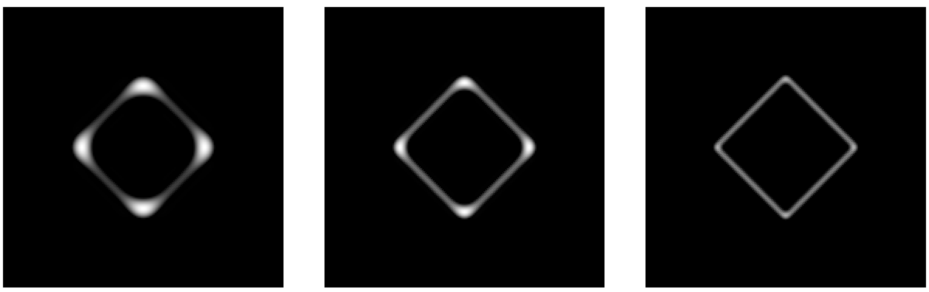
(a) (b) (c)

Figure 2.12: The recognized edges for the diamonds with λ equals 0.95 quantile.



(a) (b) (c)

Figure 2.13: The recognized edges for the diamonds with λ equals 0.99 quantile.



(a) (b) (c)

Figure 2.14: The recognized edges for the diamonds with λ equals 0.98 quantile.

Chapter 3

*** Results. Maybe something about medical applications ***

List of Figures

List of Tables

Bibliography

[1] V. R. Chaganti. Edge detection of noisy images using 2-d discrete wavelet transform. Master's thesis, The Florida State University, 2005.

[2] B. Kessler, G. Payne, W. Polyzou. Wavelet notes. The University of Iowa, Feb. 2008.

[3] S. Lahmiri, M. Boukadoum. Hybrid discrete wavelet transform and gabor filter banks processing for features extraction from biomedical images. *Journal of Medical Engineering*, 2013.

[4] T. Li, S. Ma, M. Ogihara. Wavelet methods in data mining. *Data Mining and Knowledge Discovery Handbook*, pages 603–626, 2005.

[5] L. Zhang, W. Zhou, L. Jiao. Wavelet support vector machine. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B (Cybernetics)(34.1):34–39, 2004.