A GRAPHIC TOOL FOR ANALYSIS IMAGE USING WAVELETS

M.C. Ana María Rodríguez Domínguez
Benemérita Universidad Autónoma de Puebla
Av. San Claudio y 18 Sur. Col. San Manuel. Puebla
amrodriguez@ece.buap.mx

ABSTRACT

This work provides a graphical environment as a tool to easily manipulate images using wavelets. With this tool, the user can learn and experiment how the wavelets are used in images by performing the Wavelet Transform of the image, then performing the Inverse Wavelet Transform in order to reconstruct the original image. By changing parameters involved in these processes and examining the effects produced on the image, we can process an image to achieve a desired goal. In addition, new wavelets can be tried by just giving one of the filters.

This software introduces image segmentation where each segment can be processed with a different wavelet.

Object oriented techniques were used to provide a flexible and extensible tool. Motif under Linux was used as the basis for the graphical interface and C++ as the application language.

1. INTRODUCTION

Wavelets have recently become a powerful tool in some research and develop areas and each time more people is taking advantage of wavelets in all the world. It was the main motivation of this thesis: to have an accessible computer tool which lets people to know about the wavelets behavior in an easy way, to have the possibility of create a new and own wavelet, to make the basic operations of direct and inverse wavelet transform (which are the basis for all the analysis with wavelets), to modify the parameters included in that operations and experiment the effects of employ different combinations of wavelets on a single image.

This work is centered in handling images due to wavelets can be applied in some interesting problems related with images, but it also contains the structure for signal managing [15].

WaReFrame (Wavelet Research Framework), which is the name for this software, serves as a framework for beginners and more experimented wavelet users. The graphical environment simplifies the use for non programmers and Object Oriented techniques in C++ easy extensibility by adding new functions when a user with some programming and wavelets knowledge requires it.

2. CONCEPTS ABOUT WAVELETS

A wavelet is a function that should be oscillatory (waves) and localized in time (to have amplitudes that quickly decay to zero in both, the positive and negative directions) [25]. This function is adaptable to your application or signal, that is, you are able to select the wavelet that best fits with your test signal or analysis.

The wavelet basis is a set of functions based on a single function ϕ called **scaling function** which is expanded and contracted following the recursive difference equation:

$$\phi(x) = \sum_{k=0}^{M-1} c_k \phi(2x - k) \qquad \text{(Eq. 1)}$$

where c_k are the coefficients determined by constraints of orthogonality and normalization and M is the number of non-zero coefficients (the order of the wavelet).

The process where ϕ function is scaled (expanded or contracted) and shifted is called **dilation and translation,** in this way localization in both, frequency and time can be obtained.

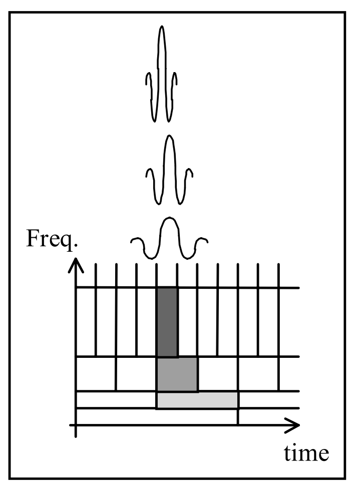


Figure. 1 Basis functions tilings on the Time-Frequency plane.[1]

To make wavelet transform in discrete time, we need to realize the following: what was a function becomes a sequence; the process that was a convolution now is a filtering process. Then, "the wavelet decomposition of

[1] Taken from [8]

data is derived *from 2-channel subband filtering* with two filter sequences: (g_k) the smoothing or **scaling filter** and (h_k) the detail or **wavelet filter**. These filters should have the following properties:

1. $\sum_k g_k = \sqrt{2}$ (Eq. 4)
2. $h_j=(-1)^j h_{1-j}$ (Eq. 5)
3. $\sum_k h_k = 0$ (Eq. 6)
4. $\sum_k g_k g_{k+2m} = \delta_{0,m}$ for all m (Eq. 7)

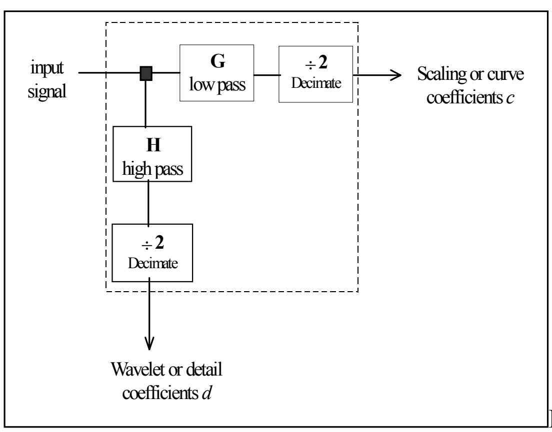


Figure 2.8 Wavelet Transform Processing Block.[2]

Reconstruction (**inverse wavelet transformation**) is performed in the opposite direction:

[2] Taken from [25]

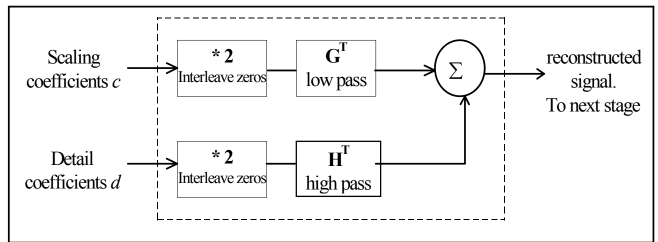


Figure 2.9 Inverse Wavelet Transform Processing Block.[3]

If we assume a signal with 2N length, the full wavelet decomposition is found applying recursively the wavelet decomposition block , in figure 2.10 is shown this process.

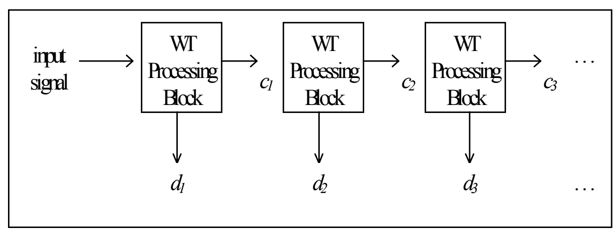
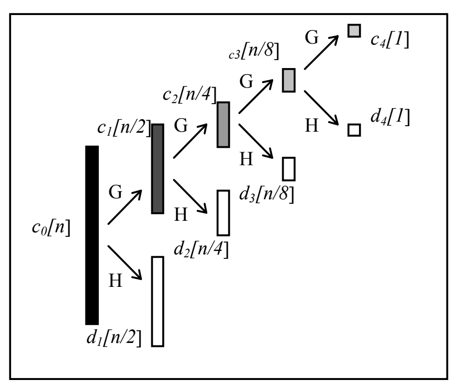


Figure 2.10 Wavelet Transform in Superior Levels.

The process ends when the scaling coefficient vector length is 1, after $\log_2(N)$ blocks. This can be seen as a pyramid or tree algorithm [16] (fig. 2.11). At the base of the pyramid is the original signal, a first level of wavelet transformation is applied and two vectors (c_1 and d_1) half the size of the original are obtained; a second level corresponds to make a new transformation over the scaling coefficients (c_1) to get two vectors with a half size of the previous vectors and successively.



[3] Taken from [25]

Figure 2.11 Discrete Wavelet Transform as a Pyramid.[4]

The multiresolution wavelet property allows us to approach the signal by successively adding details, that is, by successively refining the signal.
In applications such as signal compression, detail coefficients from the wavelet transform are not stored or, at least, not all of them, because the main information is contained in the average coefficients. In this cases the inverse wavelet transform takes the detail input coefficients as zero. For certain applications, eliminated wavelet coefficients may be only at certain levels (usually the higher levels), or eliminate the detail coefficients below a threshold value because their effect on the signal or image is not very important.

The Non-Standard Decomposition alternates between operations on rows and columns, that is, a step of horizontal pairwise averaging and differencing values in each row of the image is performed, next a vertical pairwise averaging and differencing to each column of the result is applied. In this way an image is "divided" in four equal parts: the average (scaling) coefficients, horizontal details, the vertical details and the diagonal details. This process is repeated on the quadrant of the average values and is efficient because it computes, on each step, a quarter of the coefficients of the previous level

It is easy to see that the average coefficients quadrant is in the lower left corner, but the usual distribution is: the upper left part corresponds to average (scaling) coefficients, the upper right has the horizontal details, the lower left part contains the vertical details and the lower right has the diagonal details (fig. 2.15). This depends on how are the blocks ordered into the matrix (since the unidimensional wavelet transform), but this is not critical while the transformations keep cohesion.

scaling coefficients	horizontal detail coefficients
vertical detail coefficients	transversal detail coefficients

Figure 15 Usual Coefficient Bands Distribution in the 2D Wavelet Transform.

3. WaReFrame CHARACTERISTICS

WaReFrame has not just been designed to be a friendly environment to use wavelets, it also tries to be a set of useful classes for other developments.

A first approach is to identify the main divisions that should be made over the entire system:

I. The graphical interface.
II. Managing Images.
III. Managing Wavelets.

Figure 4.3 shows the used classes and its collaborators. *MyWindow* class is the main class in the program because it organizes the execution of all the functions contained in the classes. Then, MyWindow is a class related with the basic classes (*image* and *wav_syst* and *wavelet*).

[4] Taken from [7]

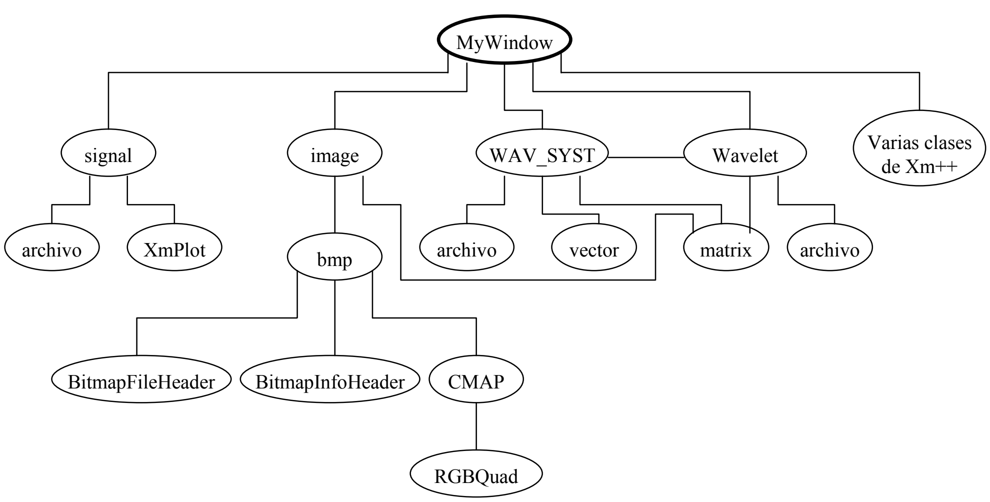


Figure 4.3 Communication between classes in WaReframe.

When planning this work, it was thought that a better way to experiment with wavelets without programming is by a graphical environment.

Working on images spends so much memory and computational time, so, it was selected the Linux platform over DOS system to make this program because its better memory management and because there are some free software useful for this project.

Be unexpensive is one desirable characteristic in this tool, so it was taken advantage of some libraries and packages included in machines using X Window System and other free libraries for Linux such as ImageMagick, Matrix++, XmPlot, actually the only library that has a cost is MOTIF.

Other desirable feature is the reusability, so WaReFrame was developed using Object Oriented Programming, because its properties of reuse and inheritance. This characteristic have allowed to use some libraries of classes such as MOTIF library, MATRIX++ library, XmPlot library in building this graphical environment.

The programming language used in building this graphical environment is C++, an Object Oriented Programming Language widely used which have, in addition, the inheritance property and whose standard compiler is included in Linux system. In this way a programmer can take the classes created for this environment and use it in other application, modify these or add some other functions. A class to manage the bmp graphic file format has been constructed. It uses classes such as *COLORMAP*, *BitmapFileHeader*, *BitmapInfoHeader*, *RGBQuad*; other class is **image** which is based in an BMP object, but that is changeable if the programmer decides to make a class to manage other image file format; **wavelet** and **wav_syst** are other important classes because they create, manage and perform wavelet operations.

While analyzing images using wavelets with some packages, the idea to use more than one wavelet in a single image, became as a new requirement for our tool: segment a single image and apply a different wavelet at each part of the image or some of them, in this way, the user is able to determine where an specific wavelet is suitable or not.

The main characteristics that the **WaReFrame** (Wavelet Research FrameWork) focused on images has are the following:

◊ Load and Save an image in any of the most common image file formats (the formats permitted by ImageMagick package).

◊ Load a pixel matrix from an ASCII file and display it as an image.

◊ Save the pixel matrix of an image to an ASCII file.

◊ Display the loaded image as an independent window.

◊ Choose a wavelet from a given set by means a file selector.

◊ Introduce a new wavelet just giving the coefficients of the low pass synthesis filter, the program will generate the high pass synthesis filter and the low pass and high pass analysis filters and it will check conditions such as orthonormality, etc. If this set of filters don't have this properties, the new wavelet will be rejected.

◊ Choose parameters to use in wavelet processes, i.e. the parameters of resolution level in analysis and reconstruction, threshold and levels in which details will be considered for reconstruction, number of vertical and horizontal segments of the image.

◊ Save the parameters set (level, number of segments, wavelets for each segment, threshold) on a configuration file to be utilized in other session of WaReFrame loading that file.

◊ Make the Wavelet Transform to the image in use and display results.

◊ Make the Inverse Wavelet Transform to reconstruct the analyzed image.

◊ Change wavelets to use on each segment of the image.

◊ Modify the image editing pixel by pixel.

4. CONCLUSION

A graphical environment to manage wavelets over images is given: **WaReFrame**. In an attempt to give the user a simple way to learn and graphically experiment with wavelets; a new tool has been created in the form of a graphical interface. The beginner user does not need to know (in principle) the complex mathematics related with wavelets, he/she just has to load and try different images, wavelets and parameters and compare the obtained results; a more experienced user can create, in addition, a new orthonormal wavelet, save results, to add some algorithms, etc. It is a tool for any kind of users.

Because WaReFrame was designed using the OOP techniques, the classes designed during WaReFrame development can be used outside the graphical environment as a library, that is, classes such as *image*, *wavelet*, and *WAV_SYST* can be used by other programs written in C++. In addition, WaReFrame and the classes created in its development, can be easily extensible to add some other function or algorithm and enhance their behavior.

WaReFrame provides a simple way to perform image analysis *by segments* (of the same dimension) in an easy way. The user just has to indicate the number of segments and to analyze each one with a selected wavelet. In this way the user can run a multi-analysis on a single image.

WaReFrame can create a new orthonormal wavelet just specifying the low pass synthesis filter, then WaReFrame produces the other three filters and checks the fulfillment of the requirements to qualify as a wavelet.

WaReFrame is simple to use because of its graphical environment. Other tools consist of libraries (XWPL exception) written in C, Matlab or Mathematica language. So, the user has to know the laguage. WaReFrame has a friendly interface ready to use. In the case of toolboxes for Matlab or Mathematica, it is similar in its use.

Handling of images and signals in a single tool. The existing libraries and GUIs are specially made for one-dimensional or two-dimensional signals, not for both of them.

Easy way to segment a signal and analyze it using different parameters for each segment. This characteristic is not available in any other tool.

WaReFrame contains a simple wavelet creation method. With this option, the user can create orthonormal wavelets right away, store it in a file and load it later. Not all the existing tools have this function.

This work could be as large as the applications of wavelets in some knowledge areas and as extensible as the advances in wavelet theory. This was just the first approximation to build a new tool for anyone who wants to investigate the wavelet behavior on images.

REFERENCIAS

[1] Bradley J, N., Brislawn C. M., Hopper T. The FBI wavelet/scalar Quantization Standard for Digital Fingerprint Images. *Visual Info. Process. II, vol. 1961 of Proc. SPIE*, pp293 - 304, April 1993.

[2] Budd T. A. *Classic Data Structures in C++* Addison-Wesley Publishing Company 1994

[3] Cohen A., Kovacevic J. Wavelets: The Mathematical Background. *Proceedings of the IEEE,* vol. 84, No. 4, pp 514 - 522, April 1996

[4] Coplien J. O. *Advanced C++. Programming Styles and Idioms.* Addison-Wesley Publishing Company 1992

[5] Deng B., Jawert B., Peters G., Sweldens W. Wavelet Probing for Compression Based Segmentation. *Preceedings of SPIE-2034*, pp. 266 276, 1993.

[6] Edwards T. Discrete Wavelet Transforms: Theory and Implementation
http://bach.ece.jhu.edu:80/~tim/papers/wavelet.ps
Stanford University 1991.

[7] Fournier A. Wavelets and Their Applications in Computer Graphics. Introduction. *SIGGRAPH'95 Course Notes*

[8] Graps A. An Introduction to Wavelets. *IEEE Computational Science and Engineering,* Summer 1995, vol. 2, num. 2, pp 50 - 61

[9] Jawert B., Sweldens W. *An Overview of Wavelets Based Multiresolution Analyses.* Department of Mathematics, University of South Carolina, Columbia SC 29208.

[10] Kernighan B. W., Ritchie D. M. *El Lenguje de Programación C.* Prentice-Hall Hispanoamericana S. A. Segunda Edición 1991.

[11] Mendez A. *Object Oriented Scheduling Simulation Tool: Visualization Component.* Thesis M. Sc. in Electronics. INAOE 1996

[12] Meyer Yves. *Wavelets. Algorithms & Applications* Society for Industrial and Applied Mathematics. 1993

[13] Microsoft Windows Version 3.1 *Programmers Reference.Volume 3: Messages, Structures and Macros for the Microsoft Windows. Operating System..* Microsoft Corporation. 1987 - 1992

[14] Microsoft Windows Version 3.1. Programmers Reference. *Volume 4: Resources for the Microsoft Windows Operating System.* Microsoft Corporation. 1987 - 1992

[15] Ortiz A. *WaReFrame for Signals.* Thesis M. Sc. in Electronics, INAOE 1998.

[16] Reissell L. Wavelets and Their Applications in Computer Graphics. Multirresolution and Wavelets. *SIGGRAPH'95 Course Notes*

[17] Signal Theory Group at University of Vigo. "Uvi_Wave" Wavelets Toolbox for use with Matlab.
http://www.tsc.uvigo.es/~wavelets/

[18] Stollnitz E. J., DeRose T.D., Salesin D. H. Wavelets for Computer Graphics: A Primer (Part 1) *IEEE Computer Graphics and Applications.* May 1995, vol. 15, num. 3, pp 76 - 84.

[19] Stollnitz E. J., DeRose T.D., Salesin D. H. Wavelets for Computer Graphics: A Primer (Part 2). *IEEE Computer Graphics and Applications.*July 1995, vol. 15, num. 4, pp 75 - 85.

[20] Strassi B., Binder S. *Xm++ Class Reference Manual.* University of Vienna, Austria. 1994

[21] Strassi B., Binder S. *The Xm++ User's Guide.* University of Vienna, Austria. 1994

[22] Sweldens W. Wavelets and Their Applications in Computer Graphics. Wavelets, Signal Compression and Image Processing. *SIGGRAPH'95 Course Notes*.

[23] Ueda M., Lodha S. *Wavelets: An Elementary Introduction and Examples.* University of California, Santa Cruz 1995

[24] Vidakovic B., MÜller P. *Wavelets for Kids. A Tutorial Introduction.* Duke University 1991.

[25] Young Randy K. *Wavelet Theory and Its Applications.* Kluwer Academic Publishers.1994