AD_____

Award Number DAMD17-99-1-9154

TITLE: "Model & Expansion Based Methods of Detection of Small Masses in Radiographs of Dense Breasts"

PRINCIPAL INVESTIGATOR: Andrew F. Laine, D.Sc.

CONTRACTING ORGANIZATION: Columbia University
New York, New York 10027

REPORT DATE: June 2000

TYPE OF REPORT: Annual

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2000	Annual (1 Jun 99 – 31 May 00)

4. TITLE AND SUBTITLE
"Model and Expansion Based Methods of Detection of Small Masses in Radiographs of Dense Breasts"

5. FUNDING NUMBERS
DAMD17-99-1-9154

6. AUTHOR(S)
Andrew F. Laine, D.Sc.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Columbia University
New York, New York 10027

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

This report contains colored photos

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes progress made in during the first year of study. Our goal is to detect masses in dense mammograms having a diameter less than 1 cm. The "idea" of this project is to detect subtle masses by tuning the central frequency and width of a basis function generating overcomplete expansions. By modeling the shape of a mass through this flexibility we hope to detect small and subtle masses in dense breasts and improve the chances of early detection in screening mammography. In the first part of our investigation, we first evaluated existing tools to compute overcomplete expansions of multiscale signals. We compared in one dimension the CWT and the DWT for a proof of concept concerning any advantage of pursuing refinement of scale. We processed phantom masses, and 1D intensity profiles of real masses mammograms to evaluate feasibility. In order to identify the best scale, we evaluated the use of maxima singularities and a correlated model using three masses of different size. Our study answered the question of weather of not dyadic scales were sufficient to detect masses in a dense mammograms. We clearly showed that reasonable approximations of mass shapes could be obtained through overcomplete expansions that computed voices between the traditional dyadic scales.

14. SUBJECT TERMS
Breast Cancer

15. NUMBER OF PAGES
39

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

FOREWORD

Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the U.S. Army.

____ Where copyrighted material is quoted, permission has been obtained to use such material.

____ Where material from documents designated for limited distribution is quoted, permission has been obtained to use the material.

____ Citations of commercial organizations and trade names in this report do not constitute an official Department of Army endorsement or approval of the products or services of these organizations.

____ In conducting research using animals, the investigator(s) adhered to the "Guide for the Care and Use of Laboratory Animals," prepared by the Committee on Care and use of Laboratory Animals of the Institute of Laboratory Resources, national Research Council (NIH Publication No. 86-23, Revised 1985).

____ For the protection of human subjects, the investigator(s) adhered to policies of applicable Federal Law 45 CFR 46.

____ In conducting research utilizing recombinant DNA technology, the investigator(s) adhered to current guidelines promulgated by the National Institutes of Health.

____ In the conduct of research utilizing recombinant DNA, the investigator(s) adhered to the NIH Guidelines for Research Involving Recombinant DNA Molecules.

____ In the conduct of research involving hazardous organisms, the investigator(s) adhered to the CDC-NIH Guide for Biosafety in Microbiological and Biomedical Laboratories.

Andrew Lane 8/11/00

PI - Signature Date

TABLE OF CONTENTS

Introduction

This report describes progress made in during the first year of study on the project entitled "Model & Expansion Based Methods of Detection of Small Masses in Radiographs of Dense Breasts." In general our goal is to detect masses in dense mammograms having a diameter less than 1 cm. The basic "idea" of this IDEA catagory award is to detect subtle masses by tuning the central frequency and width of a basis function. By modeling the shape of a mass through this flexibility (i.e., changing the shape of the "bump") we hope to better detect small and subtle masses in dense breasts and improve the chances of early detection through screening mammography.

This fist annual report addresses progress made towards accomplishing the goals described in Tasks 1a and 1b, "Feasibility assessment and design of model based method." In addition we have begun the initial steps described in Task 2b, "Detector analysis and implementation."

Body

In our proposal we made the conjecture that existing methods of analysis computed (efficiently) on dyadic scales were not sufficient for the detection of masses: a lesion may be too blurry at one scale, and too precise at the next finer dyadic scale. Thus, in the first part of this study, we attempt to answer the question "Is it sufficient to work with dyadic scales, or is there an absolute need to compute coefficients between the scales?"

In order to provide a more portable platform for development and not "reinvent the wheel" when available, we choose to use or modify existing libraries and programs available within the mathematical community. An ancillary benefit to this approach is that when this code is made available to the research community (at the end of the project) it will be far more easy to use having been built upon a popular commercially available and supported programming environments.

Thus, we first evaluated analysis tools in our laboratory that provided the closest functionality we anticipated: (a) Matlab software, with the *Wavelab* extensions, (b) the *LastWave* software written in C++. Major analysis tools described in Mallat's book [1] "A wavelet tour of signal processing algorithms" have been implemented in Matlab. However many are not. For example we discovered that there did not exist a program to carry out a CWT_2D analysis in either Matlab or LastWave software packages. Thus we began by rewriting in Matlab several existing LastWave algorithms, such as the Discrete Wavelet Transform in two dimensions without downsampling, using the "Algorithme à Trous" [1,2].

Methodology

To begin, we compared in one dimension the CWT and the DWT for different data sets: phantom masses, and 1-D intensity profiles of real digital mammograms. Then we evaluated the shape of the "Mexican hat" for suitability in a matched filtering detection paradigm. In the next section, we introduce the notion of "voices" which allow us to compute representations in between octaves.

Voices, octaves and notation

Beyond the (Discrete Wavelet Transform) DWT which computes only at dyadic scales corresponding to octaves, it is possible to decompose at a finer granularity and compute scales between these octaves: these are called *voices* [1]. Simply, a voice constitutes a subdivision of an octave.

If we consider a wavelet mother ψ, the corresponding wavelets are:

$$\psi_{m,n}(x) = a_0^{-\frac{m}{2}} \psi(a_0^{-m}x - nb_0) \text{ where } \begin{cases} a_0 \text{ is the dilatation parameter,} \\ b_0 \text{ is the translation parameter,} \\ (m,n) \in \mathbb{Z}^2. \end{cases}$$

In the dyadic case, $a_0 = 2$ and $b_0 = 1$:

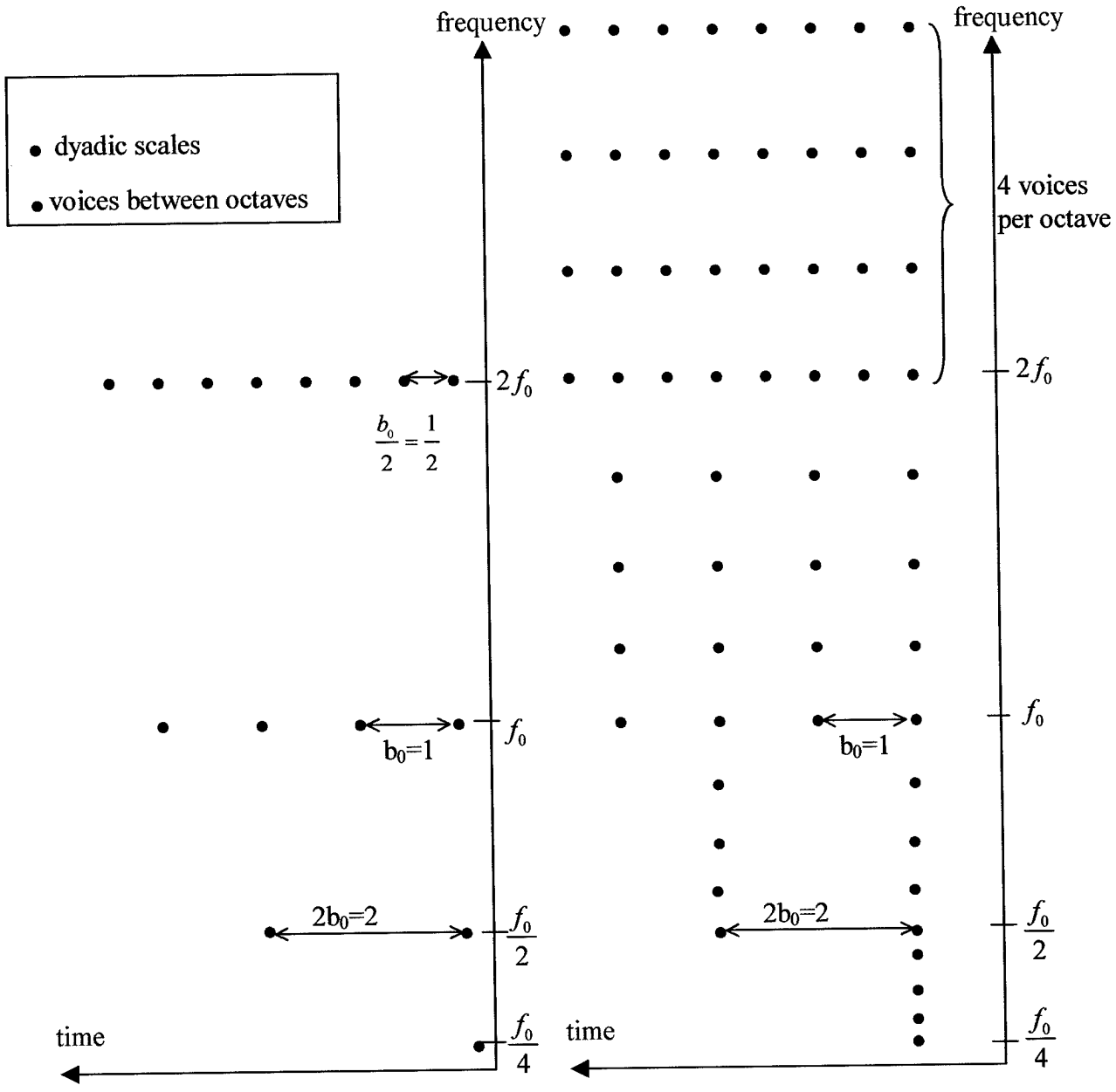
$$\boxed{\psi_{m,n}(x) = 2^{-\frac{m}{2}} \psi(2^{-m}x - n)}$$

Decomposing N voices per octave means creating N functions $\psi_{n,m}^{\nu}$ and looking at the frame $\{ \psi_{m,n}^{\nu}; (m,n) \in \mathbb{Z}^2, \nu = 1,...,N \}$.

Analyzing with N voices means finding N different frequency levels which correspond to the N different central frequency localizations of $\psi^1,...,\psi^N$, all translated by the same translation step (Figure 1(b)). Such a lattice can be viewed as the superposition of N different lattices of the type shown in Figure 1(a), "stretched" by different amounts in the frequency direction. One possible choice for ψ^{ν} is

$$\boxed{\psi^{\nu}(x) = 2^{-\frac{\nu-1}{N}} \psi(2^{-\frac{\nu-1}{N}} x)}.$$

If $|\hat{\psi}(\xi)|$ (which we assume to be even) peaks about $\pm\omega_0$, then $|\hat{\psi}^j|$ will be concentrated around $\pm 2^{-\frac{j-1}{N}} \omega_0$ (in the same way as in the dyadic case; if $|\hat{\psi}|$ has two peaks in frequency at $\pm\xi_0$, $|\hat{\psi}_{m,n}(\xi)|$ then peaks at $\pm 2^m \xi_0$ which become the two localized centers of $\psi_{m,n}$).



(a) Dyadic wavelet transform: $\psi_{m,n}$ is localized around $2^m n b_0$. $a_0=2$ and we assume $b_0=1$.

(b) An analysis with four voices: the different voice wavelets ψ^1, ..., ψ^4 are assumed to be dilatations of a single function ψ,

$$\psi^j(x)=2^{-\frac{j-1}{4}}\psi(2^{-\frac{j-1}{4}}x).$$

Figure 1: The time-frequency lattice comparing traditional dyadic time-frequency partitions (red dots) and tiling of the time-frequency plane via voices (red and black dots).

To designate the j^{th} voice of the i^{th} octave, we will use the following notation:

- Ai,j for the approximation coefficients,
- Di,j for the detail coefficients.

The equation computing the scale when given an 'octave', 'current_voice' and 'number of voices' is:

$$scale = 2^{octave + \frac{current\_voice}{number\_voices}}.$$

Moreover, we adopt the following convention: the first octave (octave number zero) corresponds to the width between scales $1 + 2^{\frac{1}{number\_voices}}$ and 2. The dyadic scale of an octave is the last voice within this octave (scale = $2^{\text{octave}+1}$). On the lattice shown in Figure 2, we consider a signal of 512 points (2^9). This means 9 octaves (octave 0 to octave 8). Thus the coarsest scale is 512 points and the finest consists of $1 + 2^{\frac{1}{nvoices}}$ points. For example, when we display the **second voice** of the **fourth octave** (four voices per octave being computed), we obtain the scale $2^{4+\frac{2}{4}}$, or we effectively compute coefficients at **scale 23** (as shown in Figure 2).

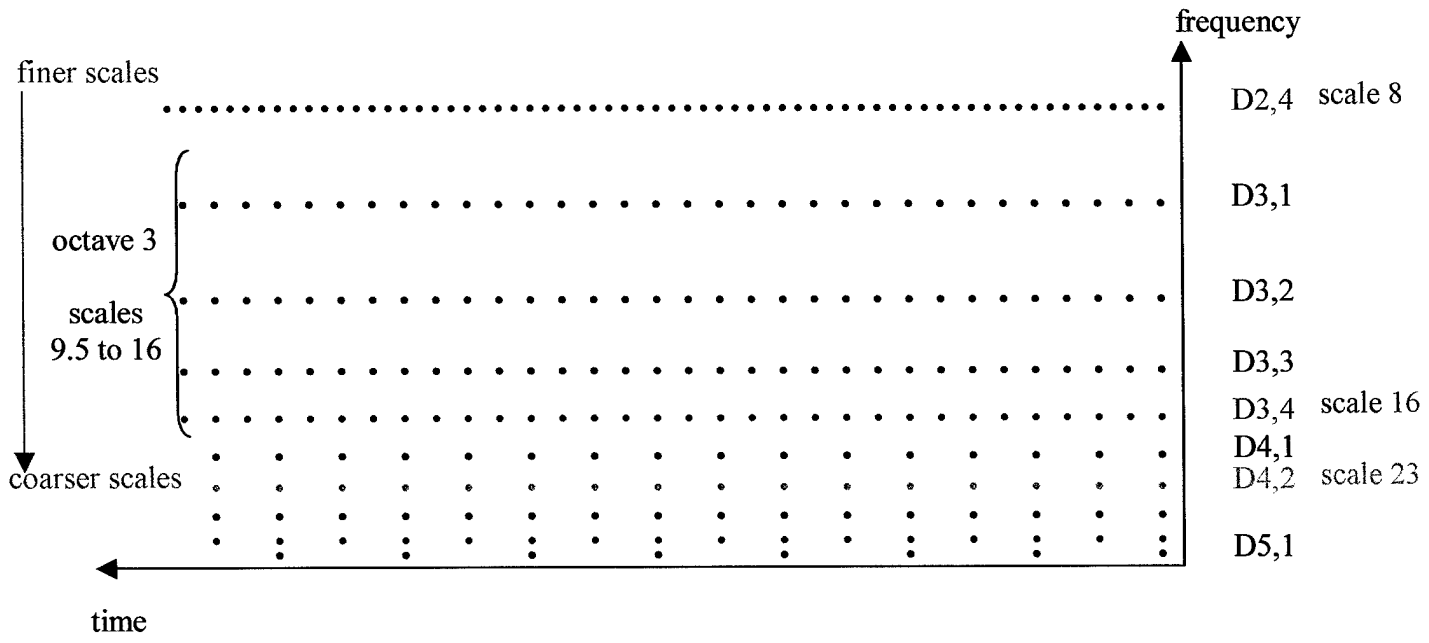


Figure 2: Linking the notions of scale, octave and voice: The time-frequency lattice for a scheme with four voices per octave, including the scale axis.

Experiments in one-dimension

In this initial phase we carried out a systematic study using 1-D algorithms of the continuous and the discrete wavelet transform. We modified *LastWave* utilities (`package_wtrans1d`) to compute a DWT or CWT in 1D, and display the plot of the coefficients.

For the CWT, different types of expansions are possible including the first, second, third and fourth derivative of a gaussian function and the Morlet complex wavelet. We concentrated on the first and the second derivative of a gaussian function, applied on signals simulating **three masses of distinct sizes** using the same kind of noise (Figure 3) and on line profiles extracted from mammograms with real masses.

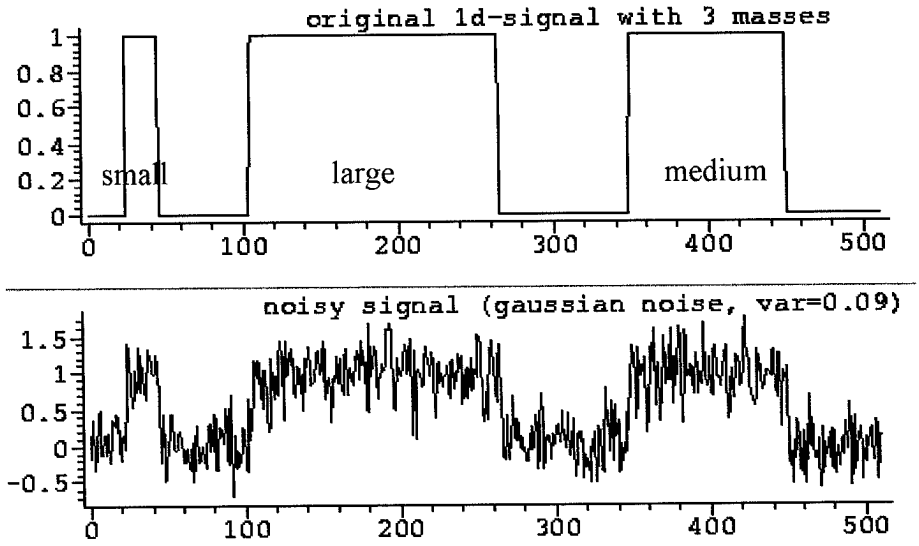


Figure 3: Phantom signal with an addition of a white gaussian noise of variance 0.1.

The program computing the continuous expansions was designed with the input parameters describing below:

`cwtd` <*aMin*> <*nOct*> <*nVoice*> <*wavelet*>

Parameters : **aMin** is the first scale applied to the transform,
nOct the number of octaves,
nVoice the number of voices per octave,
wavelet part of the gaussian wavelets family ('g1, g2, g3, g4').

The results displayed are with and without thresholding of the coefficients. The program corresponding to the thresholding operator was

`wthresh` <*threshold*> [-e <*alpha*>],

where the option '–e' is used to change the threshold value along scales. For example the threshold at octave 'o' and voice 'v' is specified by $threshold * 2^{alpha*(o-1+\frac{v}{nvoice})}$.

In addition, we modified an existing program in *Wavelab* for computing DWT in one dimension which does not downsample along scales. The program is called `FWT_ATROU` and is based on the "atrou " algorithm, using a cubic spline basis. The results for both simulated profiles and real masses are presented in the next Section.

The CWT in two dimensions

Expansions computed from the CWT are easy to understand :

(1) Compute the fast Fourier transform of the image.

(2) For each voice and for each octave, (a) the user enters an expression of the window (Fourier transform of the wavelet). The Fourier transform of the wavelet is calculated on a support depending on scale. (b) Multiply both Fourier transforms, (c) Computes the inverse Fourier transform.

Due to it's symmetric isotropic shape, we decided to investigate the Mexican Hat as a model for an expansion basis, which is the second derivative of a gaussian function. Thus we calculated the expression, in two dimensions, of the Fourier transform of the Mexican Hat.

$$\begin{cases} \text{Gaussian function : } f(t) = e^{-\frac{t^2}{2}}. \\ \text{Fourier transform of f(t) : } \hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t) * e^{-i\omega t} dt = \sqrt{2\pi} e^{-\frac{\omega^2}{2}} \end{cases}$$

$$\begin{cases} \text{Second derivative of a Gaussian function (Mexican Hat) : } f_2(t) = (t^2 - 1) * e^{-\frac{t^2}{2}} \\ \text{Fourier transform of the Mexican hat (in 1d) : } \hat{f}_2(\omega) = \int_{-\infty}^{+\infty} f_2(t) * e^{-i\omega t} dt = \omega^2 \sqrt{2\pi} e^{-\frac{\omega^2}{2}} \end{cases}$$

In two dimensions, we have :

$$\begin{cases} \text{Mexican Hat : } f(x,y) = (1 - \frac{x^2+y^2}{2}) * e^{-\frac{x^2+y^2}{2}} \\ \text{FT : } \boxed{\begin{array}{l} \hat{f}(\omega_1,\omega_2) = \iint (1 - \frac{x^2+y^2}{2}) * e^{-\frac{x^2+y^2}{2}} * e^{-i(\omega_1 x + \omega_2 y)} dxdy \\ = -\pi e^{-\frac{\omega_1^2+\omega_2^2}{2}} (\omega_1^2 + \omega_2^2) \end{array}} \end{cases}$$

Therefore, the algorithm can be described in the following steps:

- Compute the 2 dimensional Fourier transform of the image,
- Knowing the expression of the Fourier transform of the Mexican Hat in 2d and we calculate this function at discrete points (ω_1, ω_2) depending on the scale,
- Multiply these two matrices,
- Display the real part of the inverse Fourier transform of the result.

The Matlab program file which accomplishes this is provided in Appendix 1.

Results

Model based methods using 1D signals

We extracted from the profile of a 2D mass, samples of 1D signals. Plots of two intensity profiles from the real mammogram shown in Figure 4 are displayed in Figure 5 below.

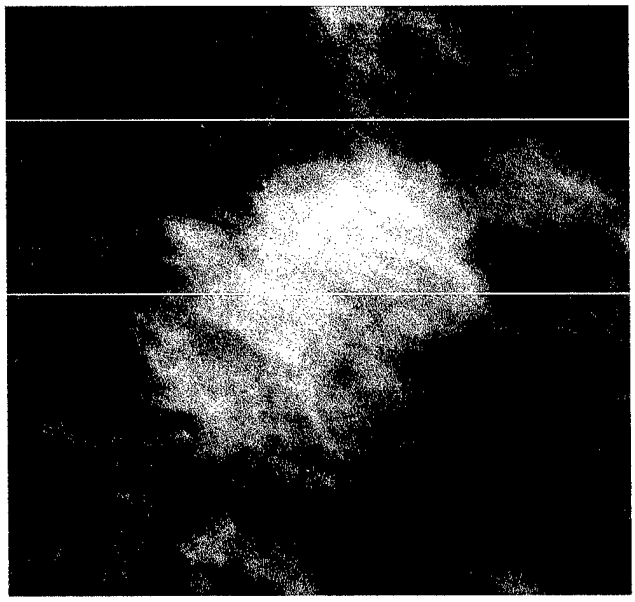


Figure 4: Real mass from mammography (South Florida University database) The white lines show locations of intensity profiles displayed in Figure 5.

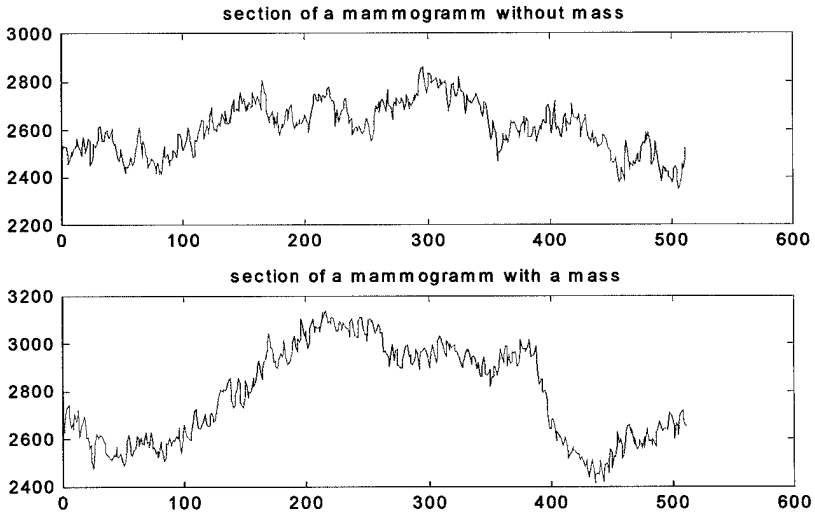
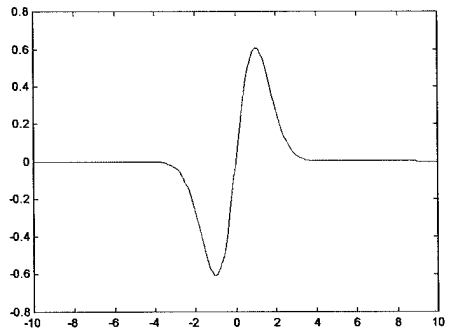


Figure 5: Intensity profiles through the mammogram, (y=100 and y=250).

As shown in Figure 8, we added gaussian noise on phantom masses so that the 1D signal had the same shape as a real mass. We worked with two different variance values: v=0.09 (shown in Figure 3) and v=0.36.

We present below some interesting plots obtained using: (a) noises with different variances, (b) a continuous and a discrete wavelet transform (with several wavelets for each), (c) with and without thresholding.



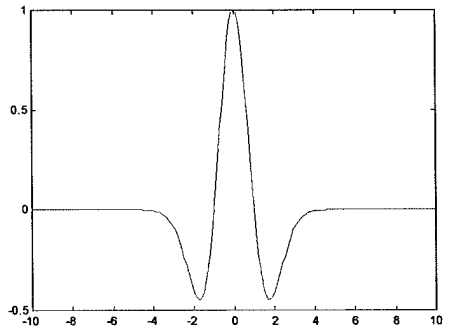


Figure 6. (Left) 'g1' wavelet basis, first derivative of a guassian function.
Figure 7. (Right) 'g2 ' wavelet basis, second derivative of a guassian function.

We computed the expansion with two types of bases: 'g1' and 'g2', meaning first and second derivative of a gaussian function shown in Figures 6 and 7 respectively.

Results for the continuous case are shown first; we display only the detail coefficients.

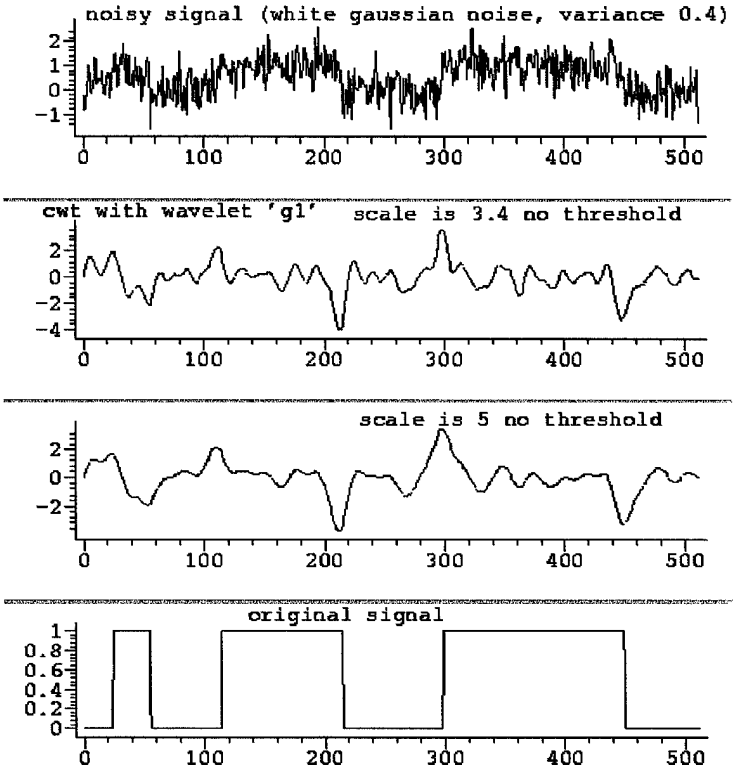


Figure 6: CWT using basis 'g1' without thresholding.

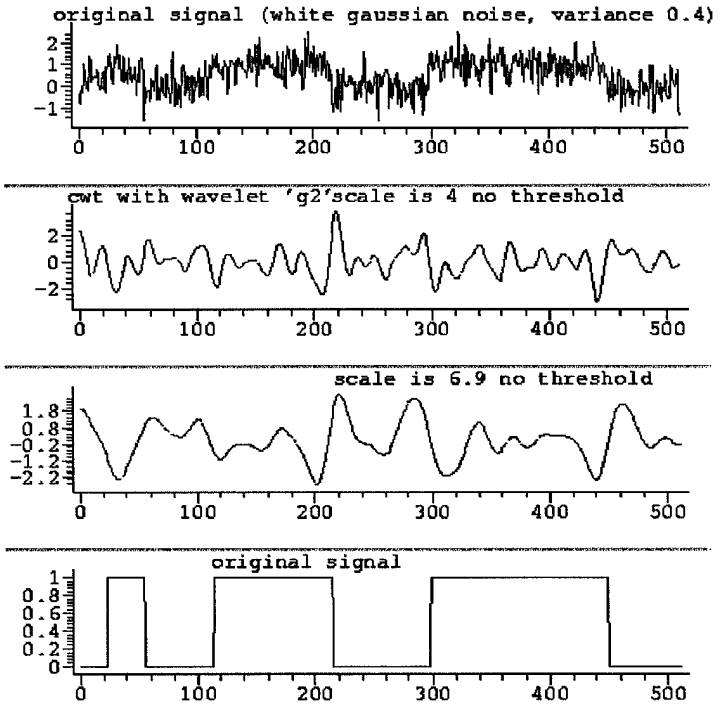


Figure 7: CWT using wavelet 'g2' without thresholding.

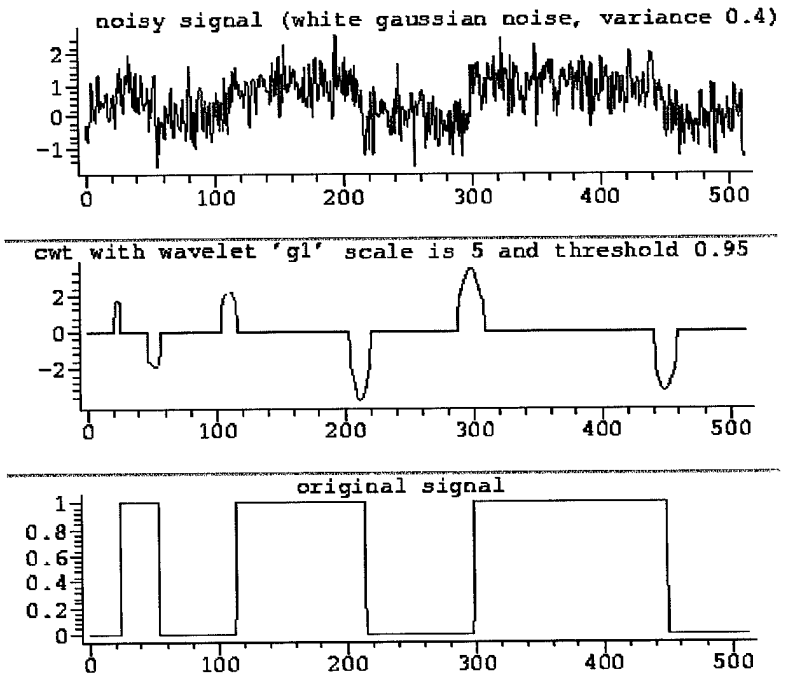


Figure 8: CWT using wavelet 'g1' with thresholding.

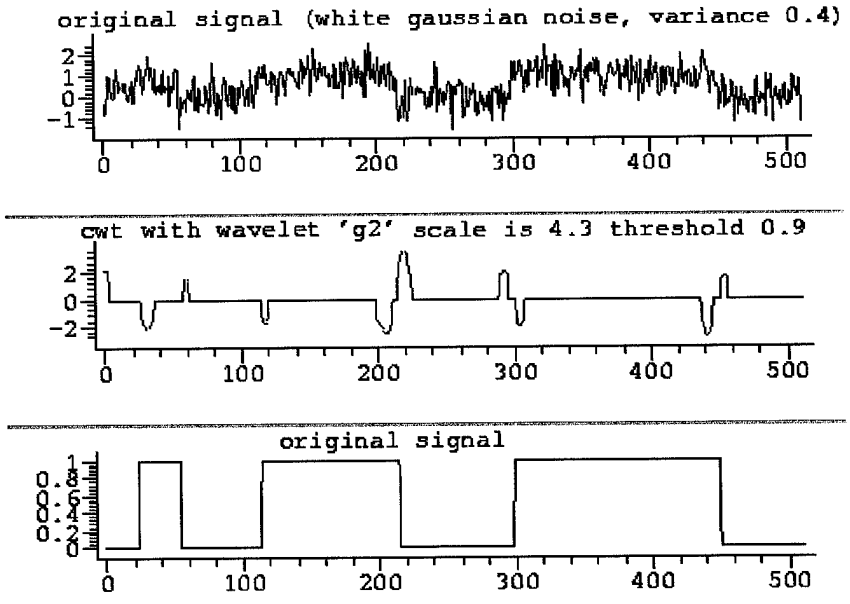


Figure 9: CWT using wavelet 'g2' with thresholding.

We then processed a phantom signal that mimicked the profile of a real mass in a mammogram. What is interesting here is the fact that for a mass of a narrow length (the first one on our case), the 'g2' wavelet does not exactly detect the end of the wavelet (there is a delay), which does not appear when we compute the transform using the 'g1' wavelet. Nevertheless, we decided to perform a continuous transform in 2 dimensions with a Mexican hat wavelet, which behaves as a second derivative of a gaussian function.

It is important to remark that there does not exist a 1D DWT program within the community of existing software including public, web, and commercial sources which does not down-sample along scales. As a result, the following plots exhibited aliasing artifacts and underestimate the performance of this expansion.

We computed the transform with three different multiscale expansions: 'D1' (Daubechies 1, one vanishing moment), 'spl1' (a linear spline) and 'spl3' (a cubic spline). Below we display the thresholded results from a linear spline expansion:

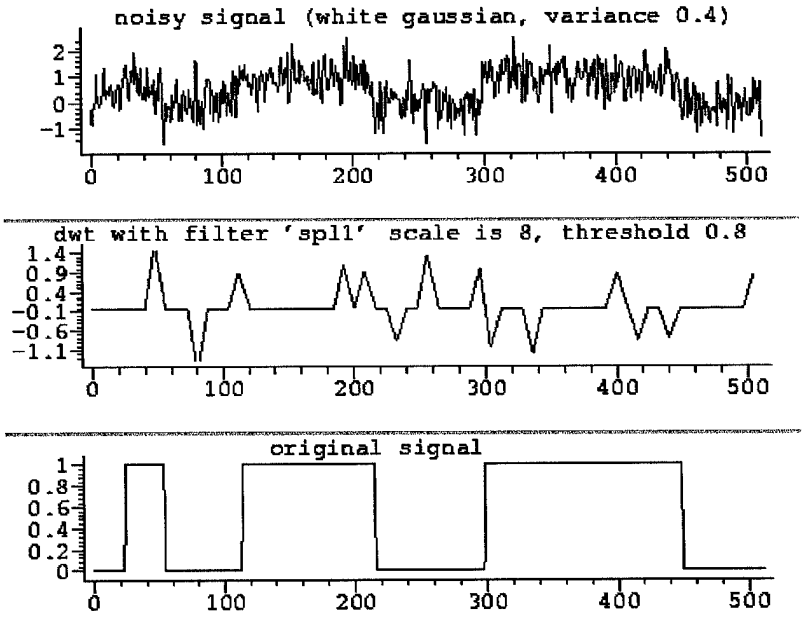


Figure 10: DWT using wavelet 'spl1' with thresholding and downsampling.

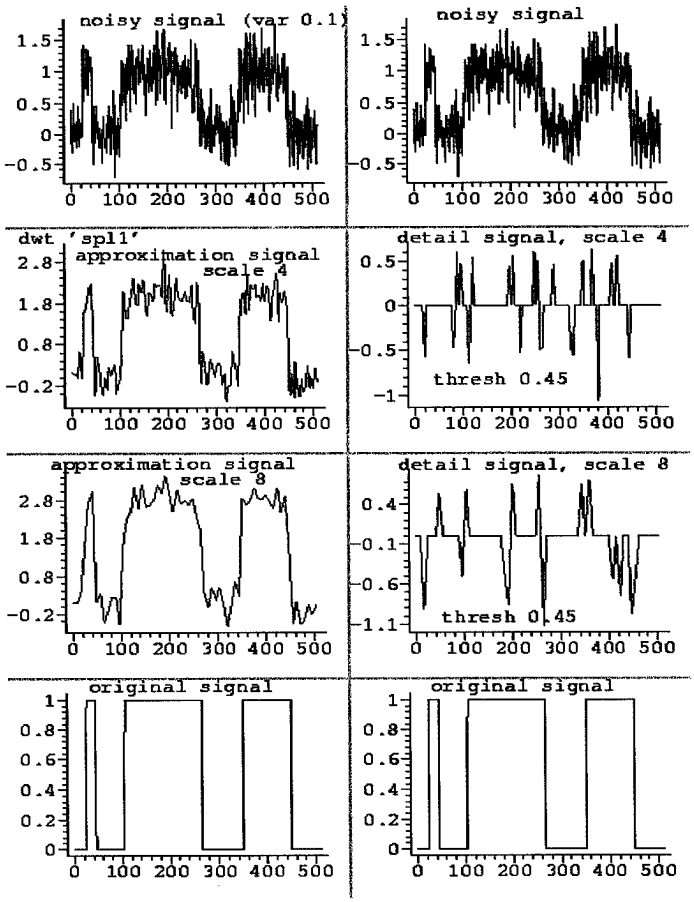


Figure 11: DWT using wavelet 'spl1,' both the approximation and detail signal shown, with downsampling.

We observed that the results were superior compared to Figure 12 with respect to the detail components of the signal. In fact, the variance of the white gaussian noise was less significant (0.1 compared to 0.4). Thus Figures 12 and 13 show why downsampling along scales is not desirable for our detection application in mammography. Processing the 1D section of a real mass with downsampled representations was a failure, since it is was almost impossible to identify the edges of the mass at any detailed scale. Below, Figures 15 and 16 show results of the DWT expansion computed without downsampling in 1D working (under the *Wavelab* environment). These results motivated us to write a *Wavelab* program **DWT_2D.m** which computed the fast discrete dyadic wavelet transform in two dimensions without downsampling, using the 'algorithme à trous' [1].

Expansion using a "Mexican Hat" basis

Next, still dealing with 1D signals we processed signals still composed of masses with white gaussian noise, but of variance 0.1. The basis for the expansion was a "Mexican Hat" wavelet comparable to the 'g2' wavelet.

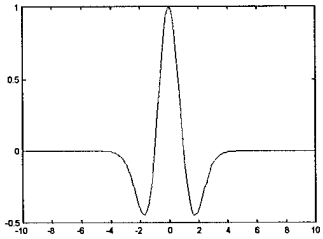


Figure 12: "Mexican Hat," a second derivative of a gaussian function.

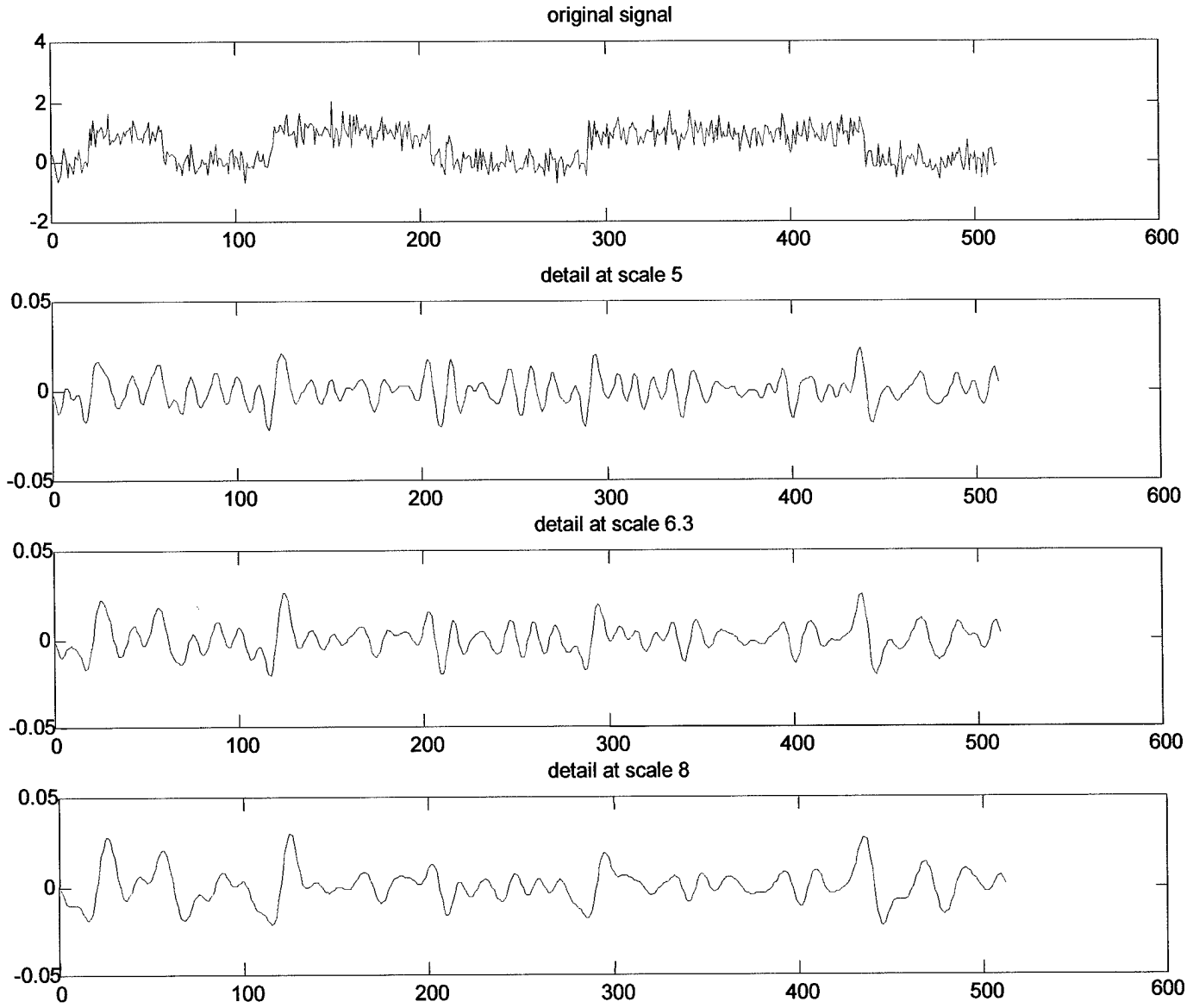


Figure 13: CWT using a 'Mexican Hat' wavelet at a given octave. Displayed are the first three voices of this octave (the noise added is of variance 0.1).

We complete this section on 1D models with the DWT as implemented in the *Wavelab* toolbox which does not carry out downsampling along scales. Thus, these results are more relevant than previous results shown earlier. Below in Figure 16, we show both the approximation and detail signals at two different scales (8 and 16).

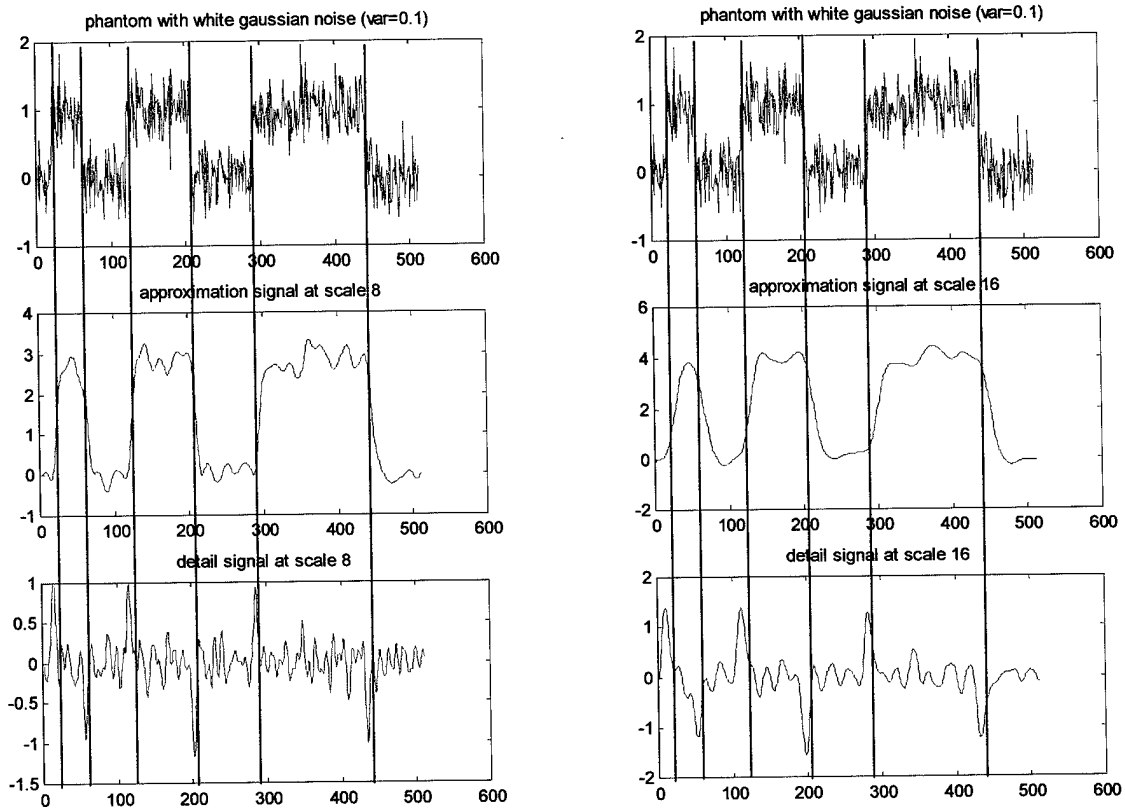


Figure 14: Expansion of a phantom model containing three masses of different sizes using a DWT without dowmsampling with gaussian noise (variance = 0.1).

Modeling of lobular masses with the CWT_2D

We designed a simple phantom of a lobular mass with the addition of white gaussian noise (variance of 4) as shown in Figure 17. Our *Wavelab* program ***cwt_2d.m*** computed the values of the transform coefficients. When using such an expansion, we obtain positive values in the center of the mass, then negative values around it. This raised an issue when we later calculate the correlation between the coefficients and the original region of interest containing the model of the mass (without noise). We investigated two ways to deal with this:

- keep the positive values only,
- add to each coefficient the value of the smallest: thus biasing the coefficients so they are positive valued.

For completeness, we display both possibilities in Figures 18 and 19. Thereafter, we will show only the biased representations. The signal-to-noise ratio (**SNR**) depending on the variance of the noise and the size of the mass is also displayed for comparison.

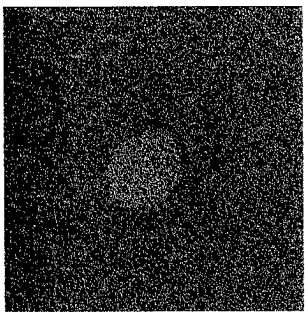


Figure 15: Noisy mass, white gaussian noise, variance = 4.

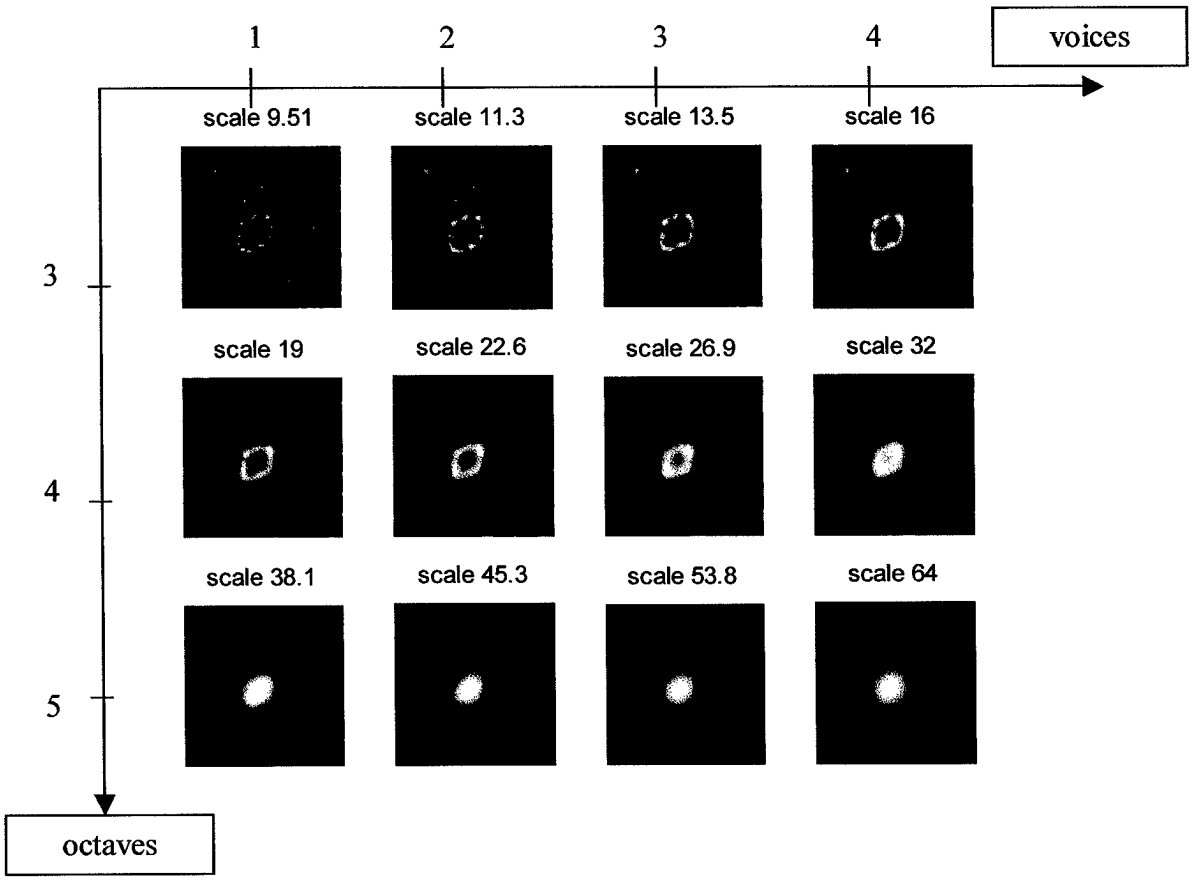


Figure 16: CWT_2D at octaves 3 to 5, four voices per octave.
No threshold applied, positive values displayed.

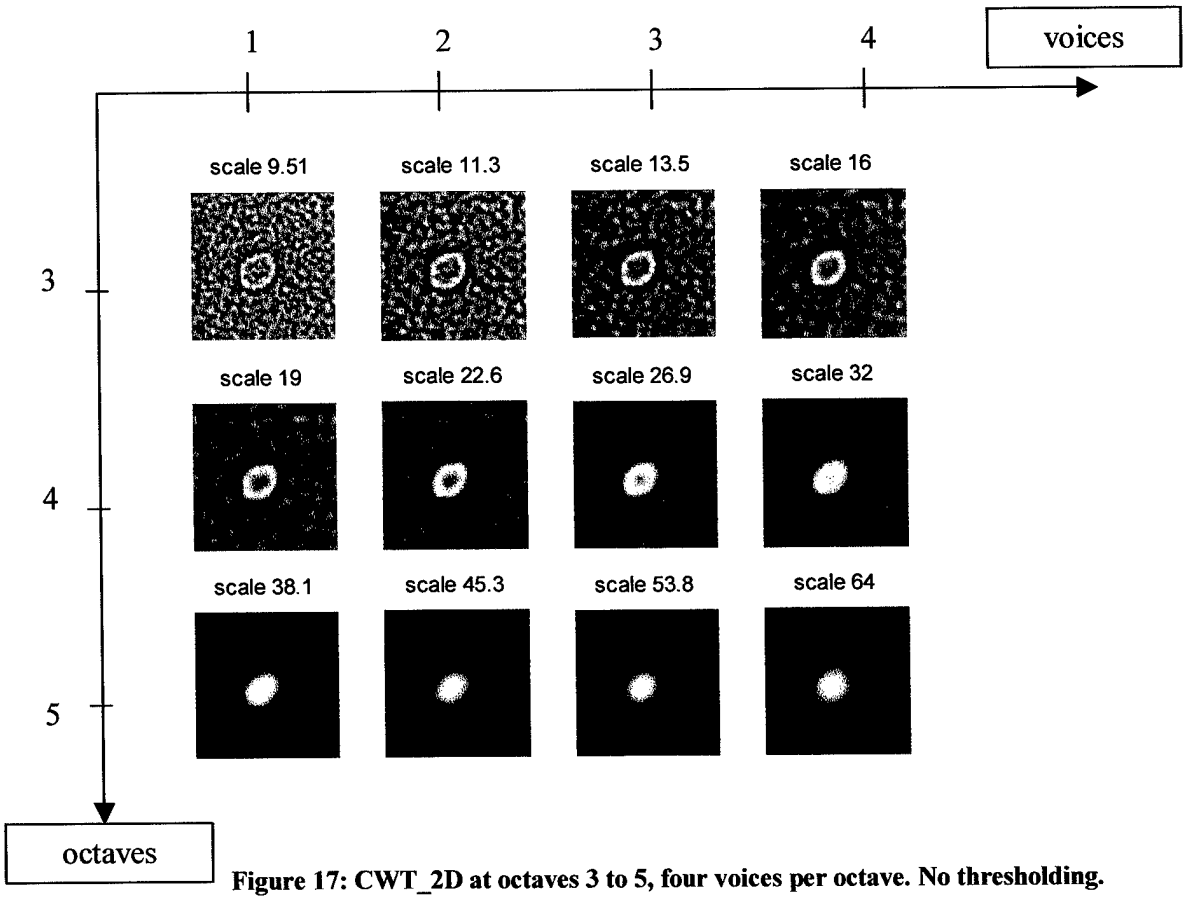


Figure 17: CWT_2D at octaves 3 to 5, four voices per octave. No thresholding. Biased representation of coefficients.

Below, we display the coefficients of the expansion with a threshold applied.

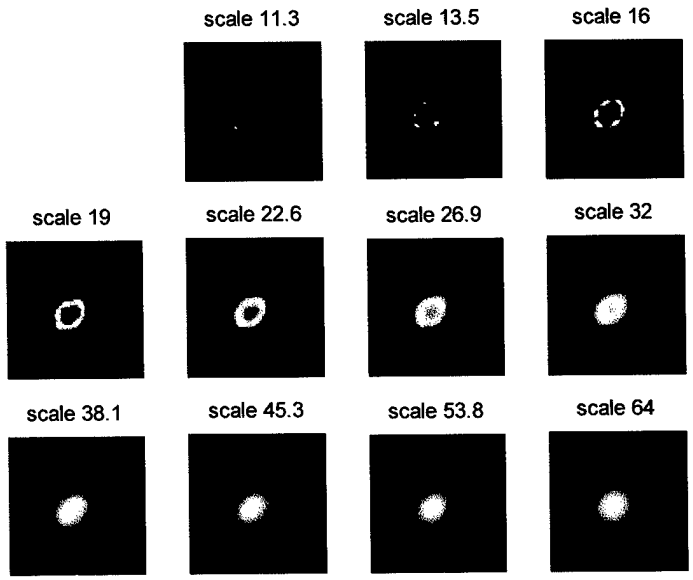
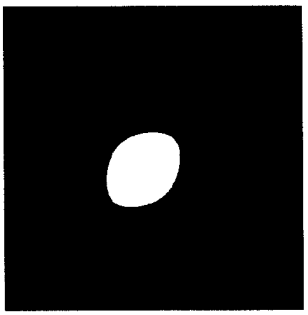


Figure 18: CWT_2D at octaves 3 to 5, four voices per octave. Coefficients were first biased and threshold (T=0.00165) applied.

For purposes of validation of detection (ground truth), the original image is shown in Figure 19, without noise.

Figure 19: Original simulated phantom of lobular mass.



Next we performed the algorithm on an image containing a real malignant mass, as shown in Figure 4. Below we show the 'raw' results in Figure 20, then corresponding alternate representations are displayed in Figure 21.

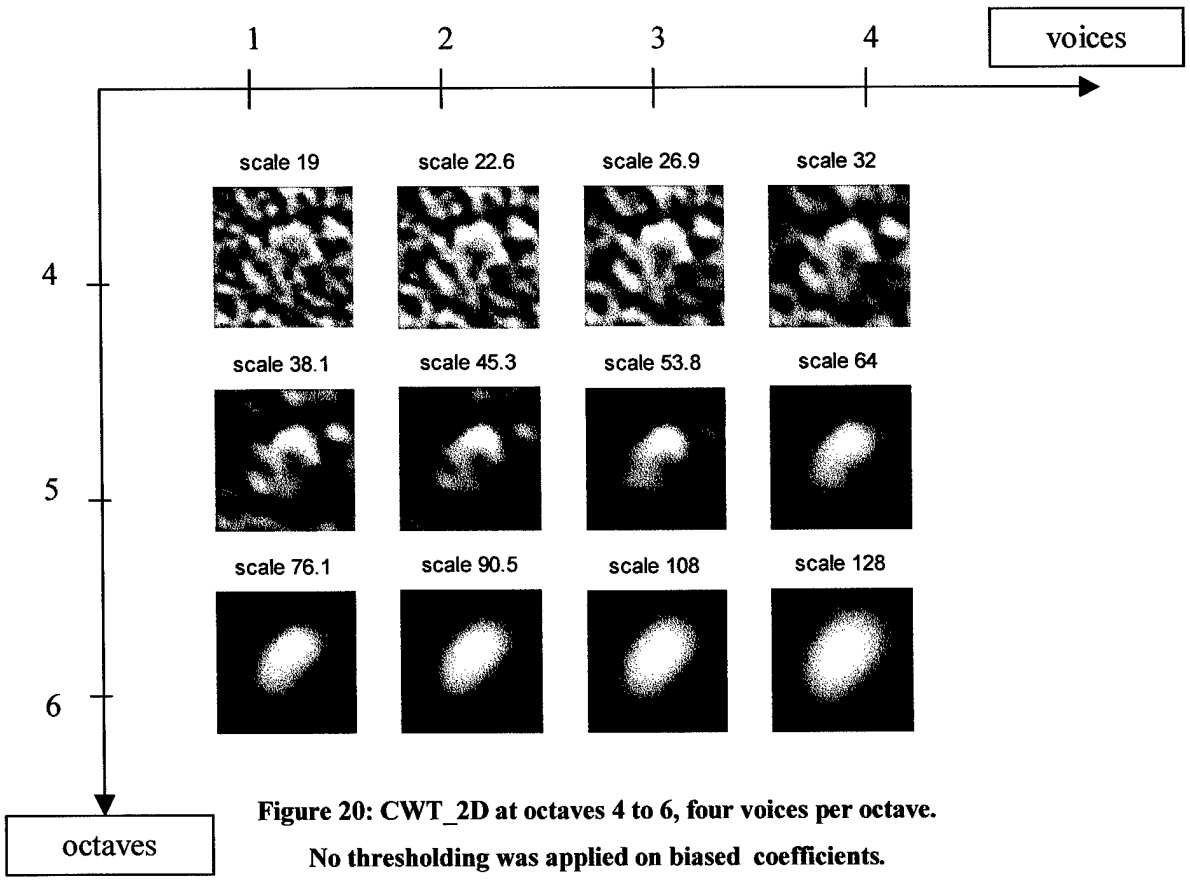


Figure 20: CWT_2D at octaves 4 to 6, four voices per octave. No thresholding was applied on biased coefficients.

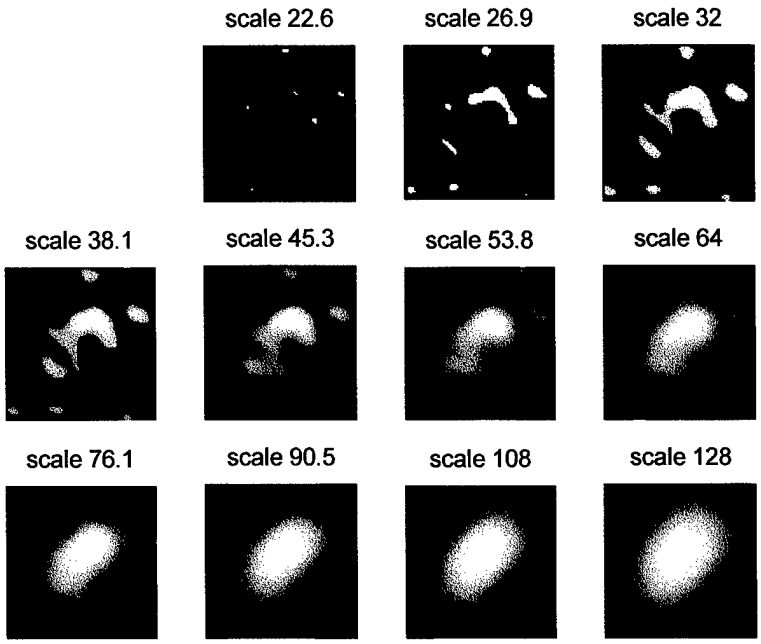


Figure 21: CWT_2D at octaves 4, 5and 6, four voices per octave. Coefficients are biased and thresholded independently at each scale (10 for scale 38 to 20 for scale 128).

The significance of the result shown in Figure 21 is that at least one of these representations exhibit a reasonable approximation of the actual shape of the mass. Thus, in terms of the parameter estimation problem (Task 1b in the Statement of Work) we are satisfied in that it appears at least feasible that there is a hope of detecting a shape that best matches the mass among the set of expanded representations. As described in the next Section, we then proceeded to develop a strategy for detection.

Discussion

We next focused on Task 2b, which includes finding the most suitable scale to detect masses. Motivated by the results obtained for one-dimension expansions, we then implemented a 2D continuous wavelet transform. This algorithm is a new program which we developed for this project and runs under the *Wavelab* environment.

Our CWT_2D program processed phantom masses, as well as real masses. The first way to identify the best scale is to simply display the maxima of the coefficients along scales, providing the magnitude of the maxima at each scale. We also investigated an alternative approach, where we plot the correlation between the original mass and the coefficients of the CWT at each scale.

We expected to find different 'optimal scales' according to the size of the mass. Thus we performed our algorithm on three different size masses, as shown in Figure 22 below.

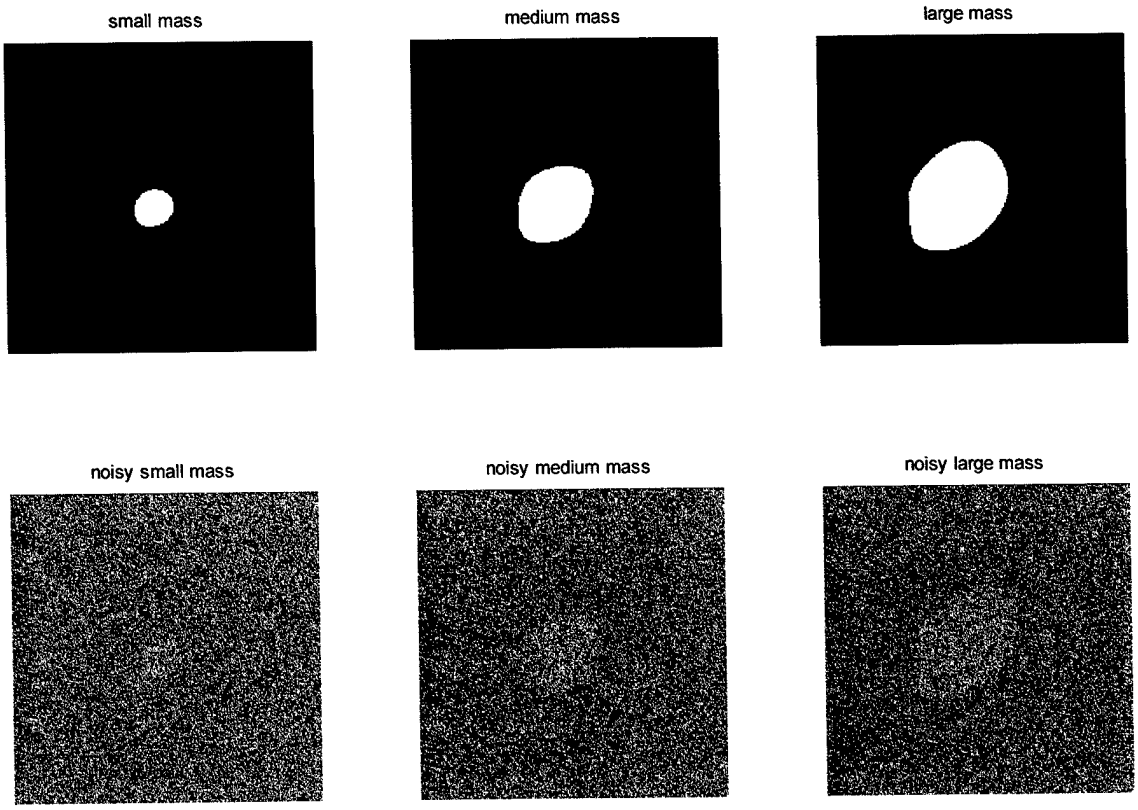


Figure 22: Simulated masses of different size for detection by identification of best scale.

Below a table giving the SNR depending on the size of the mass, and the variance of the white gaussian noise added is summarized.

	Small mass	Medium mass	Large mass
Variance 4	-4.44	1.53	3.92
Variance 1	1.72	7.52	9.99

In the first place, we perform for each mass the cwt2d on the 9 possible octaves (3 voices per octave). Then for each octave and each scale we plot the maxima of the coefficients of the wavelet decomposition (Fig 25).

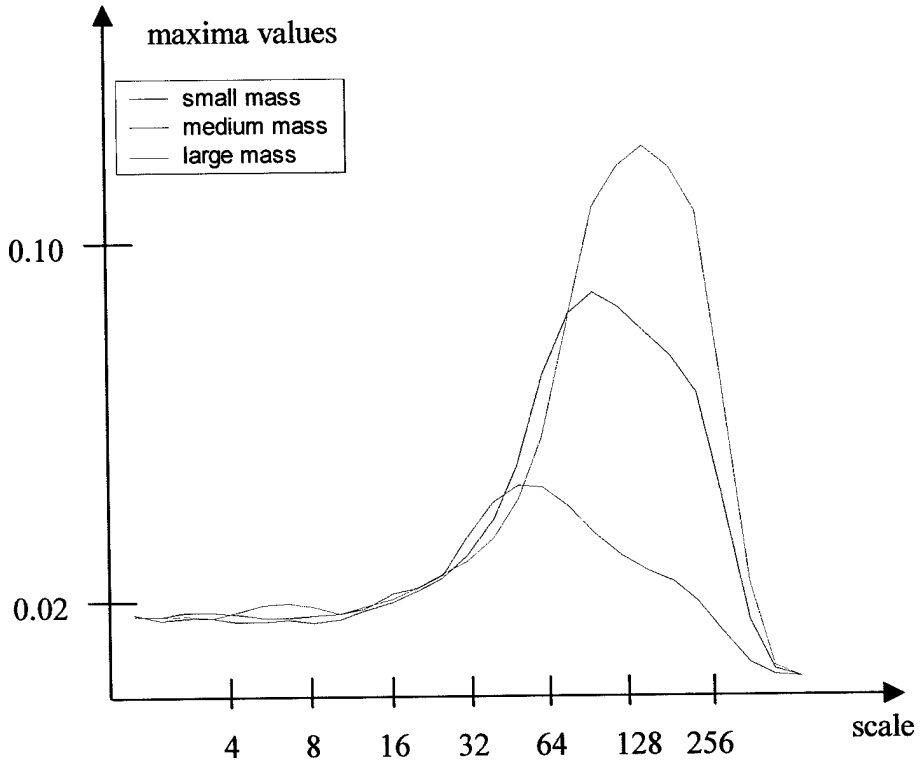


Figure 23: Evolution of the maxima coefficients for the cwt2d expansion across scales.

Position of the maxima of the decomposition: (1) Small mass: scale 40, (2) Medium mass: scale 81, (3) Large mass: scale 128.

Nexrt we performed the cwt2d on the same number of octaves and voices. For each scale we calculated the correlation between the original image (without noise) and the **cwtd2** decomposition with the formula

$$f \circ g = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m,n).g(m,n),$$

where f is the original function (M points) and g contains the coefficients of the CWT (N points).

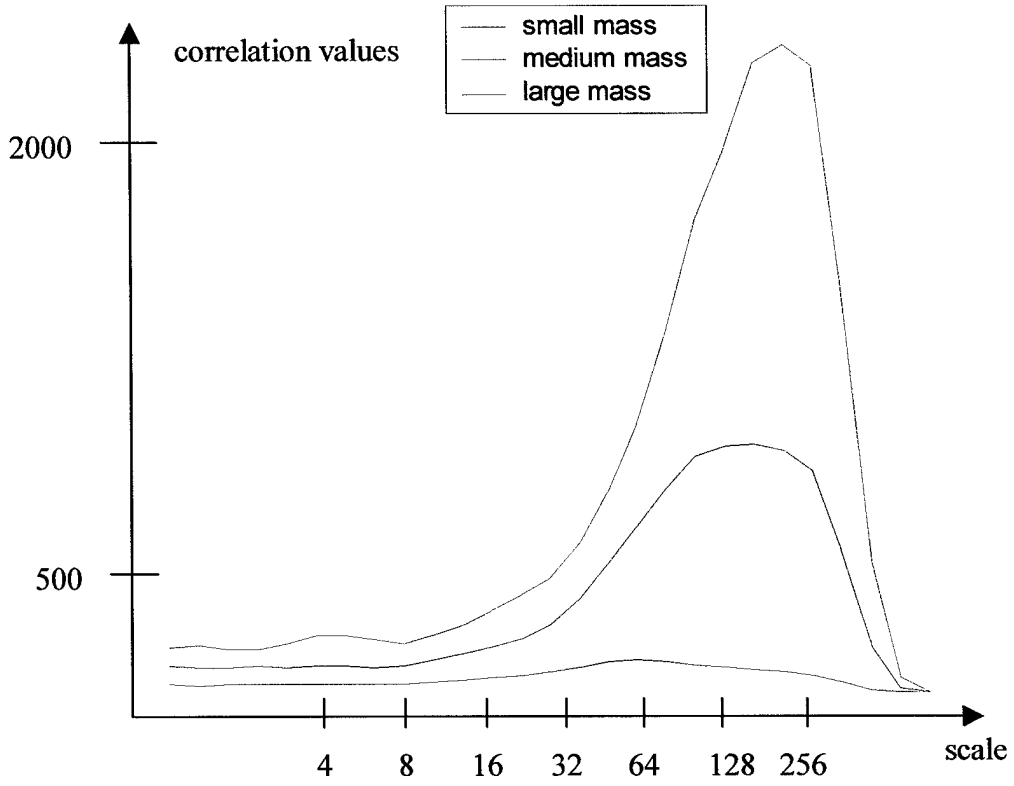


Figure 24: Correlation between the original image and the values of biased coefficients of the expansion.

Maxima of the correlation: (1) Small mass: scale 64, (2) Medium mass: scale 102, (3) Large mass: scale 161.

The most suitable scale using the method of the maxima is roughly the same as the correlation (the difference is only of one 'voice'). When we computed the algorithm with a greater number of voices per octave, we still observed a slight difference between each method. Nevertheless, **this experiment showed that the best scale depends on the size of the mass.**

Next, we show a visualization of the coefficients which suggested how well the 'voices' described a given size mass over a range of scales. These results are shown graphically for each of the three sizes of masses in Figures 25, 26 and 27 below.

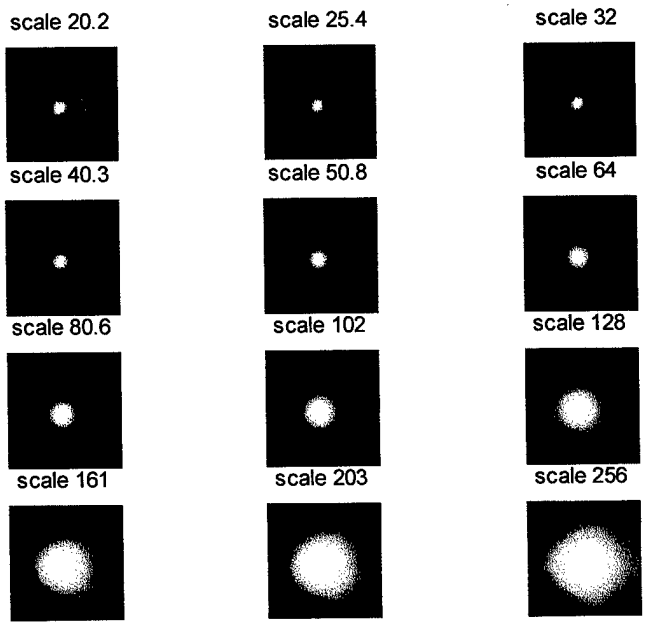


Figure 25: CWT_2D for the small mass at octaves 4 to7, three voices per octave, biased values of coefficients.

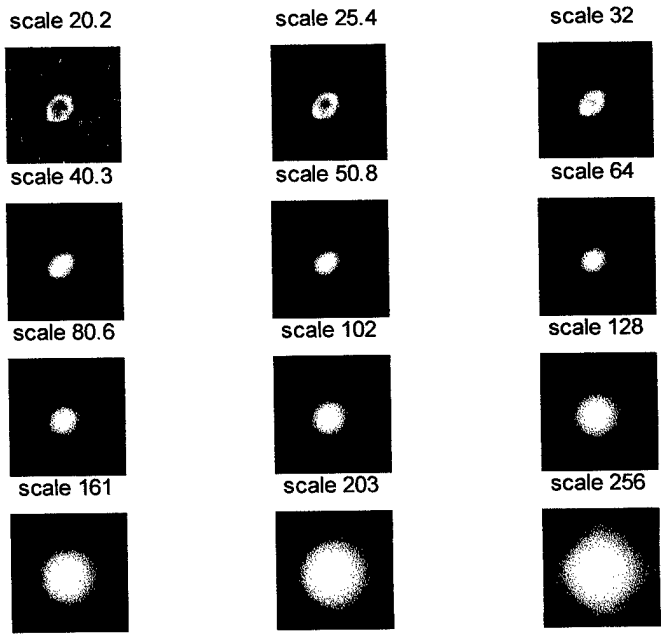


Figure 26: CWT_2D for the medium mass at octaves 4 to 7, three voices per octave.

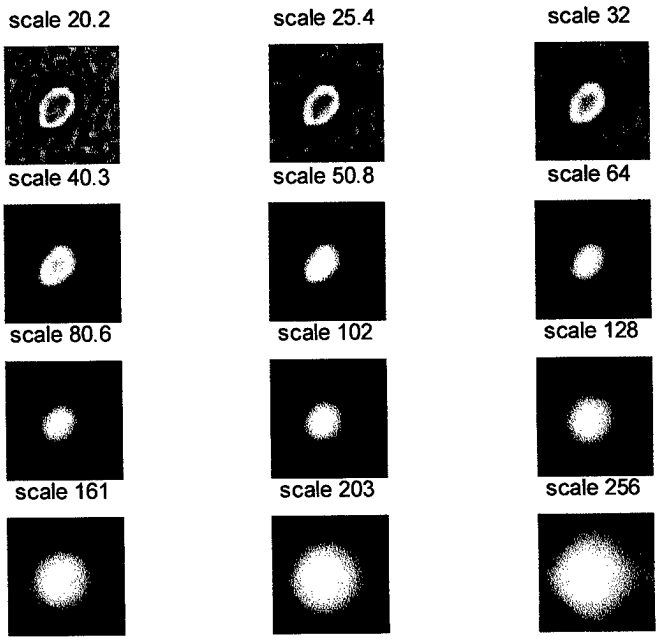


Figure 27: CWT_2D for the large mass at octaves 4 to 7, three voices per octave.

We next carried out the same processing on the real mass (Figure 4) to also find the best scale. Figure 28 shows the maxima, and Figure 29 displays the correlation result.

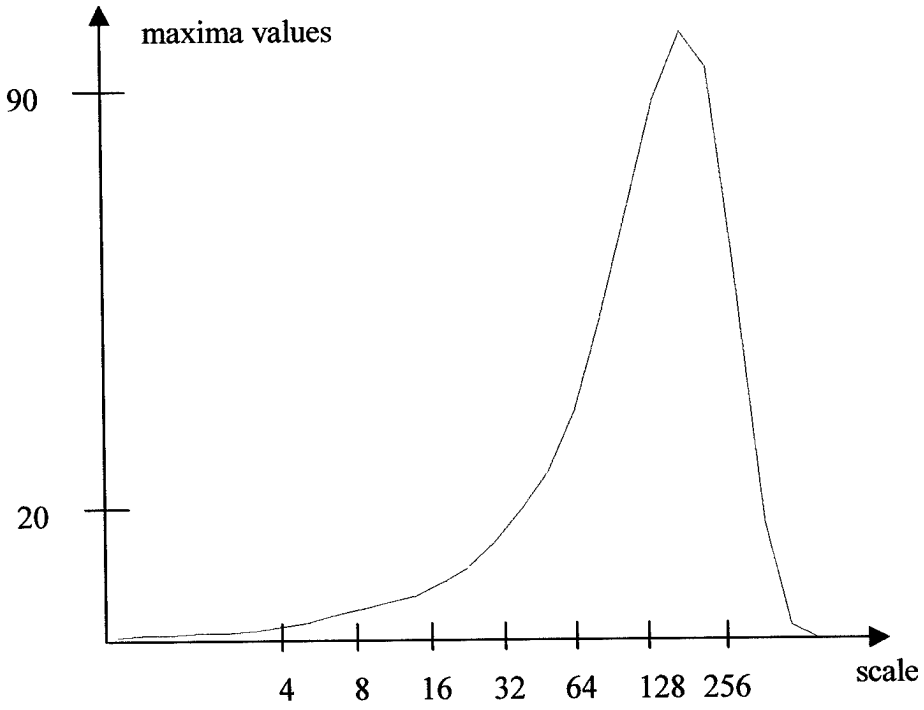


Figure 28: Evolution of the maxima of the CWT across scales (real mass). Max located at scale161.

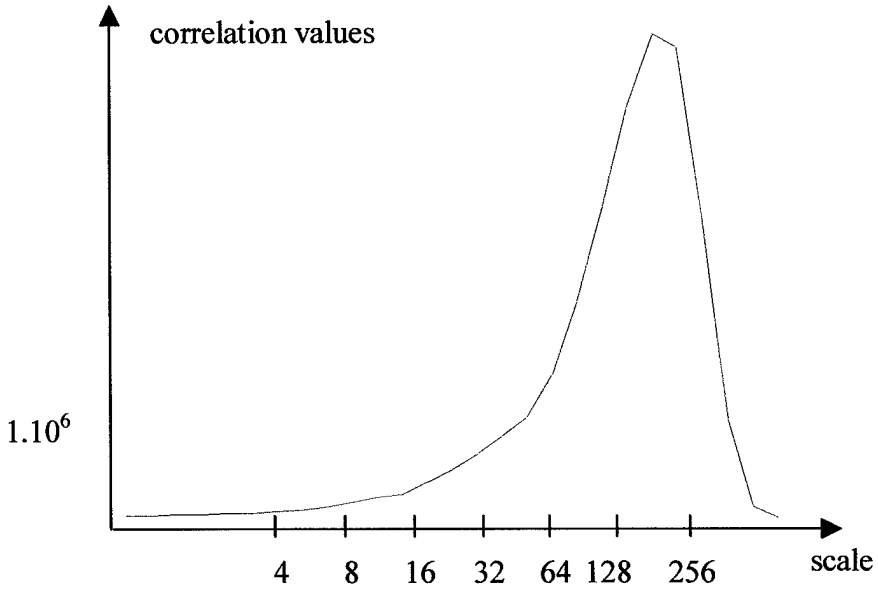


Figure 29: Correlation between the original image and the biased values of the CWT2D (real mass) Max identified at scale 161.

Thus we observed that the best scale to detect the real mass was the same for both methods (maxima and correlation).

Next, a real mass and it's associated binary mask, are displayed in Figure 30. We also plot the computed coefficients shown in Figure 31.

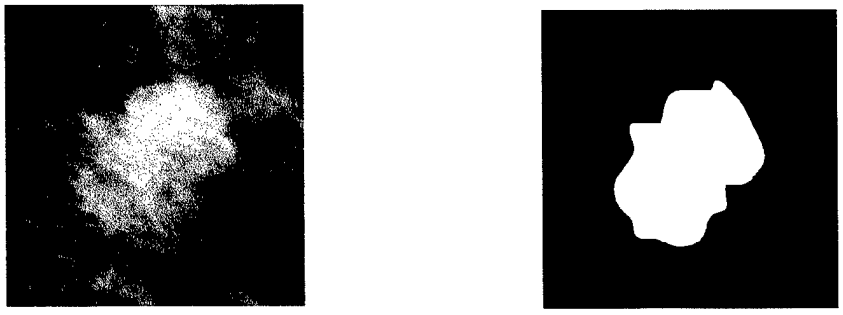


Figure 30: Real mass and phantom mask for detection by correlation.

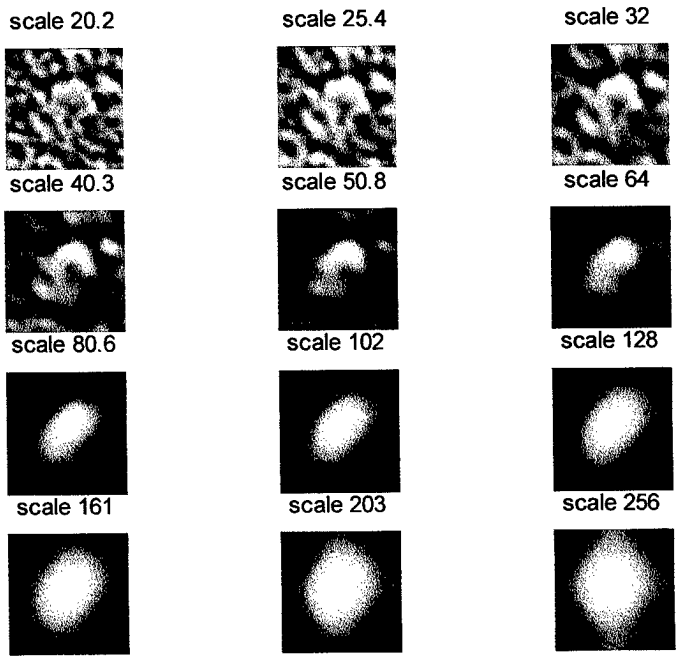


Figure 31: CWT_2D at octaves 1 to 5, five voices per octave.

Comment: According to these preliminary results, the correlation and the maxima do not give perhaps the best result for detection. Most likely, observers would not choose the scale identified as best (e.g. in our judgment, scale 40 for the medium mass appeared best, which was less blurry than scale 102 which was selected by these unsophisticated methods).

Our next experiment was to see if the variance of the noise changed these results. We previously considered a very noisy signal (variance 4) which does not capture the range of possible noise conditions. Thus, for the three different sizes of masses, displayed in Figure 34 the *maxima* of the coefficients, the *correlation*, Figure 35 for three cases were processed:

- White gaussian noise of variance 4 added to the original phantom mass,
- White gaussian noise of variance 1,
- No noise at all (phantom mass only).

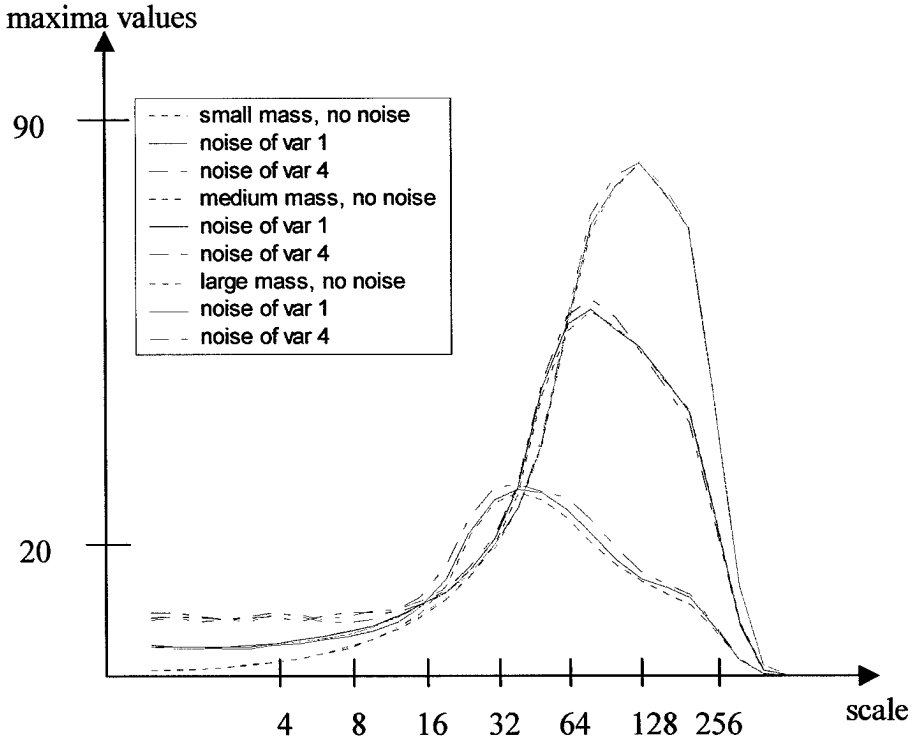


Figure 32: Evolution of the maxima of the cwt2d across scales, for three masses of different size and under different noise conditions.

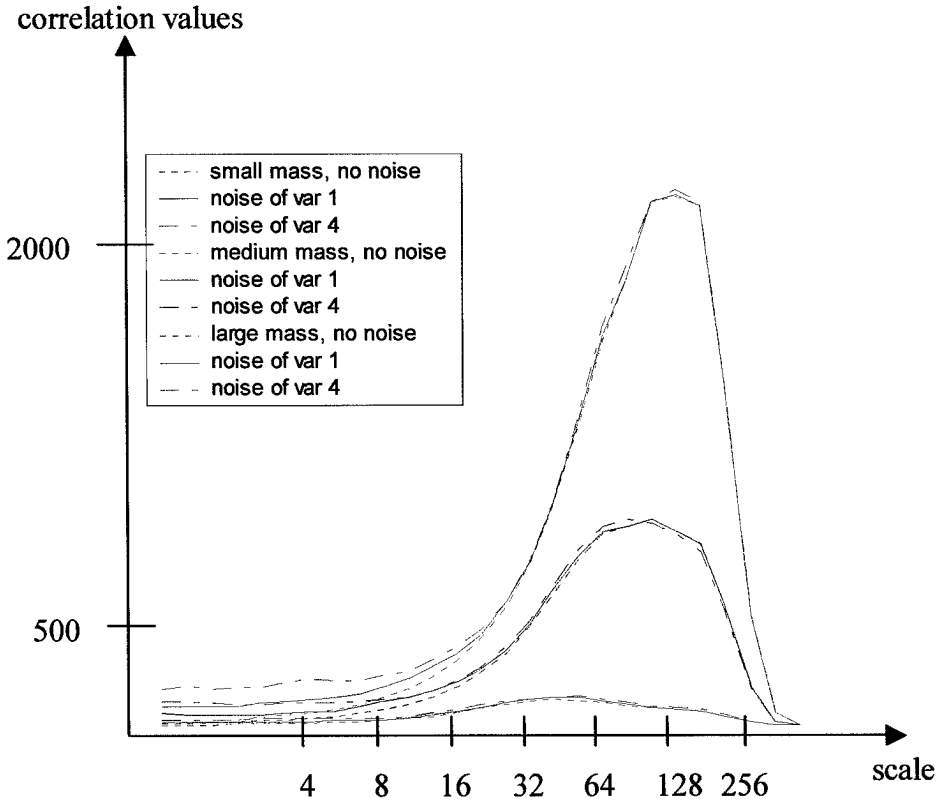


Figure 33: Correlation between the original image and the biased values of the CWT2D decomposition for three masses of different size under different noise conditions.

It appeared clearly that both methods were invariant to the amount of noise added to the phantom mass. In order to test for robustness in dealing with less symmetric masses, we then created a phantom mass with less smooth edges, as shown below in Figure 34.

(a) Original image.

(b) Noisy image (gaussian noise, variance 1).

Figure 34: An asymmetric mass model, with oblique edges.

We display below in Figure 37 the values of the coefficients computed from expansion.

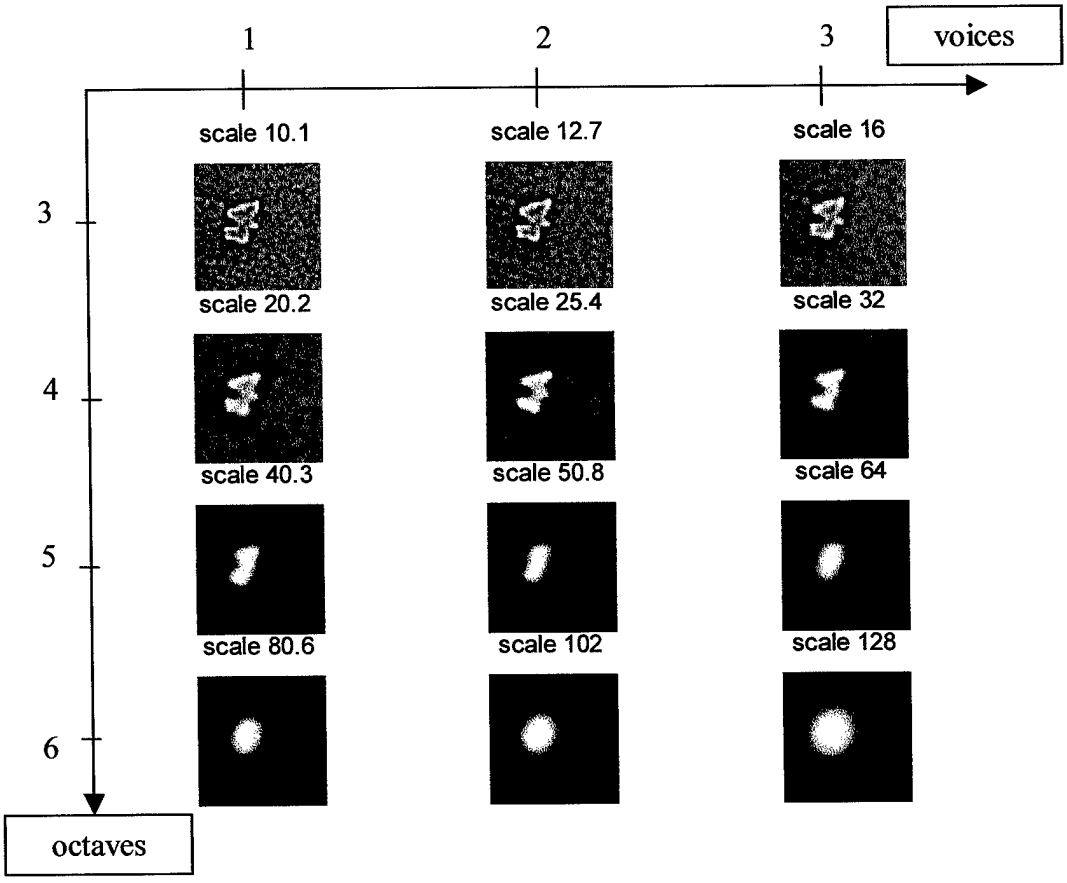


Figure 35: CWT_2D at octaves 3 to 6, three voices per octave.

We then computed plots for the maxima (Figure 36) and the correlation (Figure 39). We observed that the results were quite similar to the previous study of the three masses.

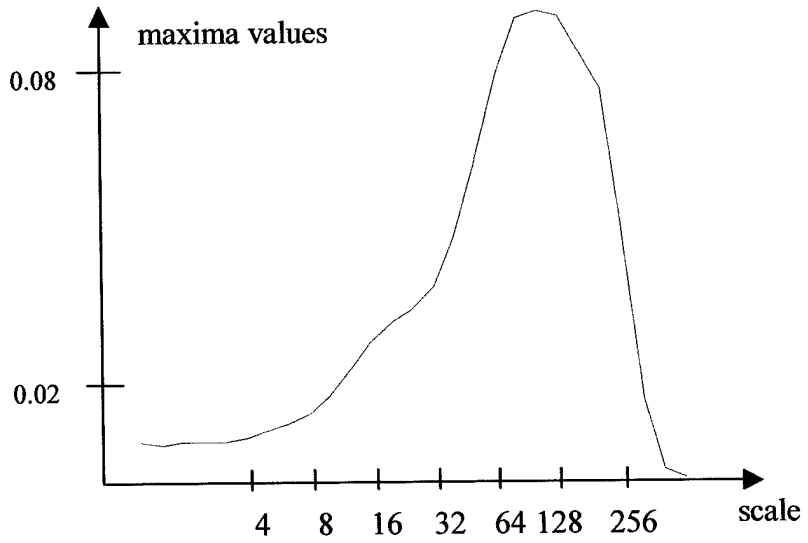


Figure 36: Evolution of the maxima of the CWT across scales.

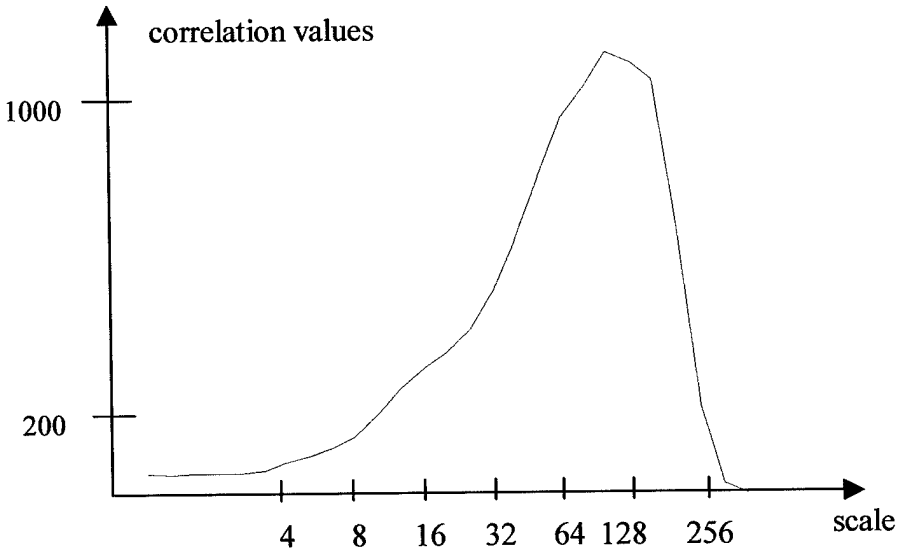


Figure 37: Correlation between the original image and the biased values of the CWT2D.

However, these two methods (maxima of the values of the coefficients and correlation between the original image and the values of the coefficients) give only a rough idea of the best scale for detection. Since both methods provided the similar results, we decided to proceed with our analysis using only the maxima method, due to its more efficient representation and faster computation time.

Key Research Accomplishments

a) During the first year of our study we first evaluated the existing tools in the research community to compute overcomplete expansions of multiscale signals. We modified under *Matlab* several existing *LastWave* algorithms, including the Discrete Wavelet Transform in two dimensions without downsampling, using the Algorithme à Trous algorithm.

b) We compared in one dimension the CWT and the DWT in order to show a proof of concept concerning any advantage of pursuing refinement of scale. We processed phantom masses, and 1D intensity profiles of real masses mammograms to evaluate feasibility. In order to identify the best scale, we evaluated the use of maxima of the coefficients and a correlated model using three masses of different size.

Reportable Outcomes

During the first year, no manuscript, patents, or supplemental grants were obtained as a result of this study. We plan to have sufficient preliminary data to submit an abstract to the RSNA 2001 and a paper to Medical Imaging (SPIE) before the end the year.

Conclusions

Our study of one dimension cases answered the question of weather of not dyadic scales were sufficient to detect masses in a dense mammograms. We showed that reasonable approximations of mass shapes could be obtained through overcomplete expansions of a continuous wavelet transform that computed voices between the traditional dyadic scales.

We observed mathematical phantoms and real masses that a correlation method (between a model of a mass and the values of the computed coefficients) gave approximately the same results when compared to the maxima method (maximum of the coefficients at each scale). This is interesting since the correlation is not computable on real images. However, we realized that both methods provided as a "best scale" a much coarser scale than we would visually expect.

The next phase of this project will focus on Tasks 2 and 3. We have recently developed a simple scheme to detect masses using these representations. This method based on geometric properties of segmented masses within each expansion has recently shown to be remarkably stable. We will describe this approach and report on its evaluation in our next report.

References

[1] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: Siam, 1992.

[2] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic Press, 1998.

Appendix I

Matlab files – cwt2d cwt2d_thresh display_cwt2d

```
function [COEFS,nvoice,n]=cwt2d(im,noct,nvoice)

%
% cwt2d -- 2D Continuous Wavelet Transform
%
%  Usage
%    [COEFS,nvoice,n]=cwt2d(im,noct,nvoice)
%
%  Inputs
%    im  image to perform the transform on
%    noct   number of octaves wished (be careful, the first octave
%           processed is the coarsest!!)
%    nvoice number of voices per octave wanted
%  Outputs
%    COEFS  matrix with the coefficients of the cwt2d
%    nvoice number of voices per octave for this decomposition
%    n      size of the input image
%
%  Description
%    performs the continuous wavelet transform in 2D
%    the wavelet used is the Mexican Hat
%    the cwt2d is performed between scale=2^(1/nvoice) and
     scale=2^noct.
%    The octave 'i' corresponds to scales beetween 2^(i+1/nvoice) and
2^(i+1)
%
%  See Also
%    display_cwt2d, cwt2d_thresh
%
%  the matrix COEFS is ordered from high to low frequencies


close all;
clear COEFS;

S=size(im);
n=S(1);
n1=S(2);
if n ~= n1
   disp('Error : image must be square!');
   break
end
J1 = log2(n);
J2 = ceil(J1);
if J1 ~= J2
   disp('Error : the size of the input image must be a power of two!');
   break
end
```

```
xi    = [ (0: (n/2)) (((-n/2)+1):-1) ] .* (2*pi);

line=J1-1;
raw=nvoice-1;
scale=1;

imfft=fft2(im);

for jo=1:noct
   for jv=1:nvoice
      qscale=scale*(2^(jv/nvoice));
      omega1 = xi./(qscale);
      for i=1:n
         window(:,i)=-exp(-
((omega1(i))^2+omega1'.*omega1')/2).*((omega1(i))^2+omega1'.*omega1');
      end
      %renormalization
      window=window./qscale;
      wfft=imfft.*window;
      w=ifft2(wfft);
      image=-real(w);
      image=image+abs(min(min(image)));
      COEFS((line*n+1):(line+1)*n,(raw*n+1):(raw+1)*n)=image;
      raw=raw-1;
   end
   line=line-1;
   scale=scale*2;
   raw=nvoice-1;
end
```

```
function
[COEFS_thresh]=cwt2d_thresh(COEFS,n,nvoice,octave,thresh,alpha)

%
%  cwt2d_thresh-- thresholds the coefficients of the cwt2d
   decomposition
%
%  Usage
%    [COEFS_thresh]=cwt2d_thresh(COEFS,n,nvoice,octave,thresh,alpha)
%  Inputs
%    COEFS  matrix with the coefficients of the decomposition output by
      cwt2d
%    n      size of the processed image output by cwt2d
%    nvoice number of voices of the decomposition output by cwt2d
%    octave octaves you want to threshold.
%     remember that octave i correspond to scale 2^(i+(1/nvoice)) to
      2^(i+1)
%    thresh see formula below
%    alpha  see formula below
%  Outputs
%    COEFS_thresh matrix with the thresholded coefficients of the cwt2d
%
%  Description
%    thresholds the coefficients of the cwt2d decomposition with an
     adaptative threshold.
```

```
%    at octave 'o' and voice 'v' the value of the threshold is :
thresh*2^(alpha*(o-1+v/nvoice))
%
%  See Also
%    cwt2d, display_cwt2d
%

noct=length(octave);
for jo=1:noct
   for jv=1:nvoice
      image=COEFS(((octave(1)+jo-1)*n+1):((octave(1)+jo)*n),((jv-
1)*n+1):(jv*n));
      threshold=thresh*2^(alpha*(octave(1)+jo-1-1+jv/nvoice));
      image=(image>threshold).*image;
      COEFS_thresh(((octave(1)+jo-1)*n+1):((octave(1)+jo)*n),((jv-
1)*n+1):(jv*n))=image;
   end
end
```

```
function display_cwt2d(COEFS,n,octave,nvoice)

%
%  display_cwt2d-- display of the 2D Continuous Wavelet Transform
   computed with the cwt2d function
%
%  Usage
%    display_cwt2d(COEFS,n,octave,nvoice)
%  Inputs
%    COEFS  matrix with the coefficients of the decomposition output by
     cwt2d or cwt2d_thresh
%    n      size of the processed image output by cwt2d
%    octave octaves you want to display.
%     to display octaves 0 to 3 type 0:3
%     remember that octave i correspond to scale 2^(i+(1/nvoice)) to
      2^(i+1)
%    nvoice number of voices of the decomposition output by cwt2d
%
%  See Also
%    cwt2d, cwt2d_thresh
%

noct=length(octave);
k=1;

for jo=1:noct
   for jv=1:nvoice
      subplot(noct,nvoice,k)
      image=COEFS(((octave(1)+jo-1)*n+1):((octave(1)+jo)*n),((jv-
1)*n+1):(jv*n));
      imshow(image,[])
      scale=2^(octave(1)-1+jo)*2^(jv/nvoice);
      title(['scale ',num2str(scale,3)]);
      k=k+1;
   end
end
```

Appendix 2

Matlab files – dwt2d display_dwt2d

```
function [approx,vertical,horizontal,modulus,phase,n]=dwt2d(im,noct);
%
%  dwt2d-- Fast Dyadic Wavelet Transform (periodized, orthogonal) in 2
dimensions
%
%  Usage
%    [approx,vertical,horizontal,modulus,phase,n]=dwt2d(im,noct)
%  Inputs
%    im        2-d signal; size = 2^J = n
%    noct   Coarsest scale of the decomposition is 2^noct
%  Outputs
%    approx an n times n matrix giving the coarsest approximation
coefficients of the decomposition.
%    vertical  an n*noct times n matrix giving the vertical details
coefficients at all dyadic scales.
%    horizontal   an n*noct times n matrix giving the horizontal
details coefficients at all dyadic scales.
%    modulus   an n*noct times n matrix giving the modulus.
Modulus=sqrt(vertical.^2+horizontal.^2).
%    phase  an n*noct times n matrix giving the phase.
Phase=atan2(horizontal,vertical).
%    n    size of the input image
%
%    all these matrix are ordered from high to low frequencies
%    we calculate the wavelet decomposition from finer to coarser
scales.
%
%  See Also
%    display_dwt2d
%

S=size(im);
n=S(1);
approx=im;
[lodyadf,dlodyadf,hidyadf,dhidyadf] = MakeATrouFilter('Spline',3);

for d=1:noct

   %approximation coefficients
   A=approx;
   for i=1:n
      raw=A(i,:);
      for j=1:(2^(d-1)+2^(d-2))
         raw=lshift(raw);
      end
      A(i,:)=iconv(lodyadf,raw);
   end
   A=A';
   for i=1:n
      raw=A(i,:);
```

```
      for j=1:(2^(d-1)+2^(d-2))
         raw=lshift(raw);
      end
      A(i,:)=iconv(lodyadf,raw);
   end
   A=A';

   %vertical details
   V=approx;
   for i=1:n
      raw=V(i,:);
      V(i,:)=iconv(hidyadf,raw);
      for j=1:2^(d-2)
         p=V(i,:);
         V(i,:)=lshift(p);
      end
   end

   %horizontal details
   H=approx';
   for i=1:n
      raw=H(i,:);
      H(i,:)=iconv(hidyadf,raw);
      for j=1:2^(d-2)
         p=H(i,:);
         H(i,:)=lshift(p);
      end
   end
   H=H';

   approx=A;
   vertical((d-1)*n+1:d*n,:)=V;
   horizontal((d-1)*n+1:d*n,:)=H;
   f = zeros(1,2*length(lodyadf));
   f(1:2:2*length(lodyadf)-1) = lodyadf;
   f2 = zeros(1,2*length(hidyadf));
   f2(1:2:2*length(hidyadf)-1) = hidyadf;

   lodyadf = f;
   hidyadf = f2;
end

modulus=sqrt(vertical.^2+horizontal.^2);
phase=atan2(horizontal,vertical);
```

```
function display_dwt2d(vertical,horizontal,modulus,phase,n,oct);

%
%  display_dwt2d-- display of the Fast Dyadic Wavelet Transform
computed with the dwt2d function
%
%  Usage
%    display_dwt2d(vertical,horizontal,modulus,phase,n,oct)
%  Inputs
```

```
%    vertical  vertical details coefficients output by dwt2d
%    horizontal   horizontal details coefficients output by dwt2d
%    modulus   modulus output by dwt2d
%    phase  phase output by dwt2d
%    n      size of the processed image output by dwt2d
%    oct octaves you want to display. Octave i corresponds to scale
2^i.
%        to display the octaves 1 to 5 type 1:5
%
%  Description
%    To display the 2d Fast Dyadic Wavelet Transform
%
%  See Also
%    dwt2d
%

j=1;
for i=1:length(oct)
   tmp1=vertical((oct(1)-1+i-1)*n+1:(oct(1)-1+i)*n,:);
   max1=max(max(abs(tmp1)));
   tmp1=(tmp1+max1)/(2*max1);
   image((i-1)*n+j:i*n+(j-1),1:n)=tmp1;
   tmp2=horizontal((oct(1)-1+i-1)*n+1:(oct(1)-1+i)*n,:);
   max2=max(max(abs(tmp2)));
   tmp2=(tmp2+max2)/(2*max2);
   image((i-1)*n+j:i*n+(j-1),n+5:2*n+4)=tmp2;
   tmp3=modulus((oct(1)-1+i-1)*n+1:(oct(1)-1+i)*n,:);
   max3=max(max(abs(tmp3)));
   tmp3=(tmp3)/(max3);
   image((i-1)*n+j:i*n+(j-1),2*n+9:3*n+8)=1-tmp3;
   tmp4=phase((oct(1)-1+i-1)*n+1:(oct(1)-1+i)*n,:);
   max4=max(max(abs(tmp4)));
   tmp4=(tmp4+max4)/(2*max4);
   image((i-1)*n+j:i*n+(j-1),3*n+13:4*n+12)=tmp4;
   image((i*n)+j:(i*n)+j+4,:)=1;
   j=j+5;
end
image(:,n+1:n+4)=1;
image(:,2*n+5:2*n+8)=1;
image(:,3*n+9:3*n+12)=1;
imshow(image,[])


% at each scale we normalise the vertical and horizontal details and
  the phase as follows:
% matrix=(matrix+max(abs(matrix)))/(2*max(abs(matrix)))
% to display the modulus we do : modulus=1-modulus/max(abs(modulus));
```