

Proyecto integrador DSP

Filtrado de ruido con transformada de wavelet

Gatica, Isaias
 Martin, Santiago
 Saez, Lautaro Andrés
 Vidman, Xavier Harry

Resumen—En el siguiente trabajo se estudiarán los conceptos de las funciones wavelets, transformada de wavelet, y cómo utilizarlos para filtrar ruido de señales unidimensionales y bidimensionales. En este informe se centrará el estudio en filtrado de imágenes. Para el filtrado de ruido se hará un análisis de multiresolución y un estudio de los umbrales más utilizados, con el fin de poder decidir qué umbral usar para la señal o imagen que se quiera estudiar. Como adelanto, el umbral es el parámetro que decide cómo se filtrarán los coeficientes que caracterizan a la imagen. Se describirá una implementación paso a paso de cómo realizar el filtrado de ruido, para luego poder utilizarlo en imágenes con ruido artificial y real. Luego, se podrá comparar con otros métodos de filtrado la eficiencia de utilizar wavelets para esta aplicación.

I. Introducción

Partiendo de una perspectiva histórica, se puede decir que la teoría de wavelets comenzó con los estudios de Alfred Haar en 1909. En ese entonces, la línea de pensamiento era puramente matemática, buscando como objetivo el dividir una función temporal en diferentes componentes de traslación y escalamiento. En su honor, se nombró a la primera wavelet con su nombre: Haar wavelet.

Los estudios de Alfred Haar, seguido de trabajos de Dennis Gabor, que colaboró con la construcción de más tipos de wavelets y trabajó en aplicaciones similares, se retomaron en los años 80. La evolución de la teoría de wavelets dio un salto importante durante esta década, gracias a que Stephane Mallat estableció una relación entre los filtros utilizados en el procesamiento de señales y las bases ortonormales de las wavelets. A partir de esto, Mallat desarrolló lo que se conoce como análisis de multiresolución, concepto que se explicará en el informe.

En 1988, Ingrid Daubechies publicó un conjunto de wavelets ortonormales que se convirtieron en el pilar de las aplicaciones wavelets.

En base al análisis de multiresolución, el estudio de la teoría de wavelets creció a tal punto de poder utilizarse para diversas aplicaciones. Las aplicaciones más importantes son la compresión de imágenes y el filtrado de ruido a través de la transformada wavelet. Otra aplicación importante es el poder identificar en qué momento se dan los picos de frecuencia de una señal. Con la transformada de wavelet esto es posible, ya que el análisis es temporal y frecuencial, de forma paulatina.

II. Marco teórico

Una wavelet es una “pequeña onda” que tiene su energía concentrada en un período de tiempo determinado, son de duración definida e irregulares, lo que les permite adaptarse y converger de mejor manera a la señal que se quiere analizar. Las familias de estas wavelet, también conocidas como “wavelets madres”, se definen como:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad a, b \in \mathbb{R} \quad ; a \neq 0. \quad (1)$$

Donde a y b son los parámetros de escala y traslación respectivamente. Se puede ver en la Fig.1 como trabajan los parámetros a y b . El parámetro b , asociado a la traslación, hace que la wavelet recorra a la señal $x(t)$ a través del tiempo. Mientras que el parámetro a , permite controlar el ancho de la wavelet.

Por otro lado, la transformada de wavelet queda definida como [1]:

$$C_f(a, b) = \int_{-\infty}^{+\infty} x(t) \overline{\psi_{a,b}(t)} dt \quad (2)$$

Donde $\psi_{a,b}(t)$ es la función que se utiliza para analizar a la señal $x(t)$ que se quiera estudiar. Realizando una analogía con Fourier, el trabajo que realiza $\psi_{a,b}(t)$, es el mismo que el de las exponenciales complejas $e^{-j\omega t}$. El resultado de esta transformada es una familia de coeficientes $C_f(a, b)$ que representan la señal teniendo en cuenta la wavelet madre utilizada. Es notorio que el resultado es bidimensional ya que a y b son parámetros independientes. Donde a está relacionado con la frecuencia que posee la señal y b nos brinda información del momento temporal que se analiza. En la Fig.2 se puede ver de mejor manera, donde el gráfico superior muestra cómo cambia la frecuencia de una señal en función de la traslación b , en específico en b_0 . Y el gráfico inferior muestra el resultado de la transformada, evidenciando en el tiempo b_0 el cambio de frecuencia.

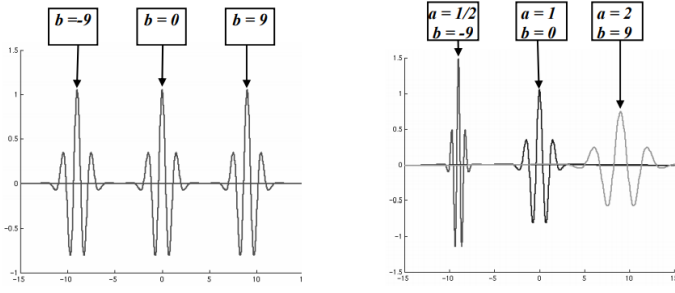


Figura 1: Traslación y escalamiento de una wavelet.

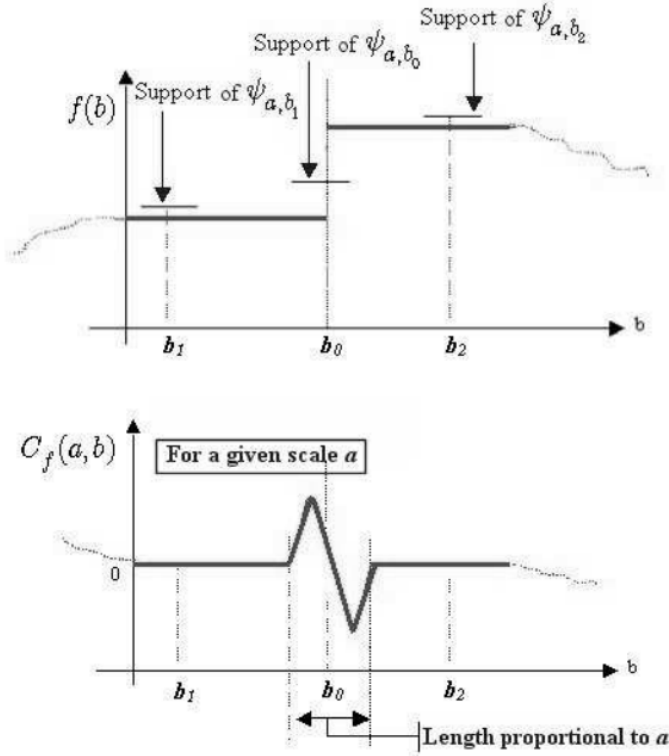


Figura 2: Representación del resultado de transformada wavelet en relación a los coeficientes de traslación y escala.

Por lo tanto, el uso de wavelets permite realizar un análisis paulatino en tiempo y frecuencia como se mencionó anteriormente. Cabe recalcar que existen diferentes tipos de wavelets madres, en la Fig.3 se pueden ver las más utilizadas.

En este informe se limita a estudiar el caso en el que $a = 2$ y $b = 1$, obteniendo la siguiente expresión:

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k) \quad (3)$$

Donde las funciones $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ forman una base ortonormal de $L^2(\mathbb{R})$. Con el fin de explicar este requerimiento, es necesario explicar un concepto llamado análisis multiresolución.

II-A. Análisis multiresolución

Un análisis multiresolución para $L^2(\mathbb{R})$ consiste en una secuencia de subespacios cerrados de $L^2(\mathbb{R})$, $\{V_j\}_{j \in \mathbb{Z}}$, y

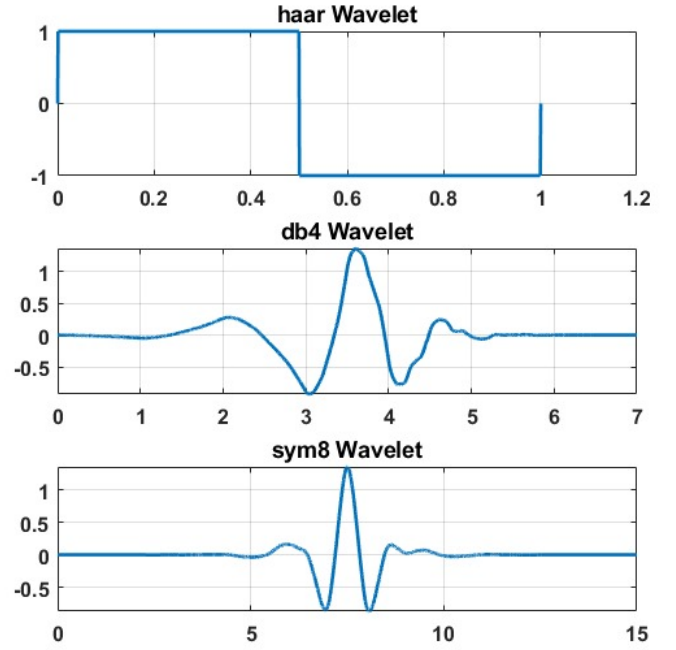


Figura 3: Tipos de wavelets más utilizadas.

una función $\phi \in V_0$ tal que se cumplan las siguientes condiciones:

- i. Los espacios V_j están anidados, es decir:

$$\dots \subset V_{-1} \subset V_0 \subset V_1 \dots$$

- ii. $\overline{\cup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R})$ y $\cap_j V_j = 0$
- iii. Para todo $j \in \mathbb{Z}$, $V_{j-1} = D(V_j)$
- iv. $f \in V_0 \rightarrow T_k f \in V_0, \forall k \in \mathbb{Z}$
- v. $\{T_k \phi\}_{k \in \mathbb{Z}}$ es una base ortonormal de V_0

Por otro lado, se define a W_j como el complemento ortogonal de V_j en V_{j-1} , que permite plantear la siguiente relación:

$$V_{j-1} = V_j \oplus W_j \quad (4)$$

W_j es llamado conjunto de detalle debido a que la proyección ortogonal de la señal en este espacio son los detalles, y V_j espacio de aproximación, ya que la proyección ortogonal de la señal sobre este espacio son las aproximaciones. Con lo cual, se puede reescribir la eq.4 como:

$$A_{j-1}(t) = A_j(t) + D_j(t) \quad (5)$$

Donde $A_j(t)$ se define de la siguiente forma:

$$A_j(t) = \sum_{k \in \mathbb{Z}} \beta_{j,k} \phi_{j,k}(t) \quad (6)$$

Donde:

$$\beta_{j,k} = \langle x(t), \phi_{j,k}(t) \rangle = \int_{\mathbb{R}} x(t) \overline{\phi_{j,k}(t)} dt \quad (7)$$

Estos coeficientes $\beta_{j,k}$ definen los promedios locales de la señal. Por otro lado, $D_j(t)$ se define a través de la siguiente expresión:

$$D_j(t) = \sum_{k \in \mathbb{Z}} \alpha_{j,k} \psi_{j,k}(t) \quad (8)$$

Donde los coeficientes $\alpha_{j,k}$ se relacionan con la diferencia entre promedios locales sucesivos (fluctuaciones de la señal), y quedan definidos de la siguiente forma:

$$\alpha_{j,k} = \langle x(t), \psi_{j,k}(t) \rangle = \int_{\mathbb{R}} x(t) \overline{\psi_{j,k}(t)} dt \quad (9)$$

La función $\psi \in L^2(\mathbb{R})$, y $\{T_k \psi\}_{k \in \mathbb{Z}}$ son una base ortonormal de W_0

Para dar un cierre a la idea de multiresolución se profundiza en la eq.4 que es el punto más fuerte de esta técnica. Se reescribe de la siguiente manera:

$$V_J = V_K \oplus W_K \oplus \dots \oplus W_{J+1}, \quad J < K \quad (10)$$

La eq.10 expresa que es posible pasar de un nivel de resolución J a cualquier nivel superior sumando los sucesivos detalles de los niveles inferiores. Es decir, la aproximación mejora a medida que el nivel J decrece. También se observa que cualquier nivel de resolución V_j se puede expresar como:

$$V_J = \sum_{j=J+1}^{+\infty} W_j \quad (11)$$

Por lo tanto cualquier nivel es igual a la suma de sus infinitos detalles. Análogamente cualquier función puede ser expresada como una suma de infinitos detalles:

$$L^2(\mathbb{R}) = \oplus_{j=-\infty}^{+\infty} W_j \quad (12)$$

A partir de las eq.11 y 12 se puede ver que:

$$L^2(\mathbb{R}) = V_J \oplus (\oplus_{j=-\infty}^J W_j) \quad (13)$$

Finalmente, como el espacio V_j se relaciona con el espacio detalle $A_J(t)$ y el $W_j(t)$ con los espacios de detalles $D_j(t)$:

$$x(t) = A_J(t) + \sum_{j=-\infty}^J D_j(t) \quad (14)$$

Por lo tanto, se define a cualquier señal $x(t)$ como la suma de una aproximación, que sería la parte más “burda” de la señal, con sus infinitos detalles, que serían “toques finales”. Esto se observa en la Fig.4, donde se utilizó una wavelet symmet 4 y 3 de niveles de descomposición. En rojo la señal original, en azul la aproximación y en verde las tres señales de detalle correspondientes.

A partir de esta idea, se ve que es factible aplicar la transformada de wavelet para eliminar el ruido de una señal. Por lo tanto, si se busca una forma de filtrar los coeficientes a partir de parámetros relacionados con el ruido de la señal, se podrían eliminar las componentes

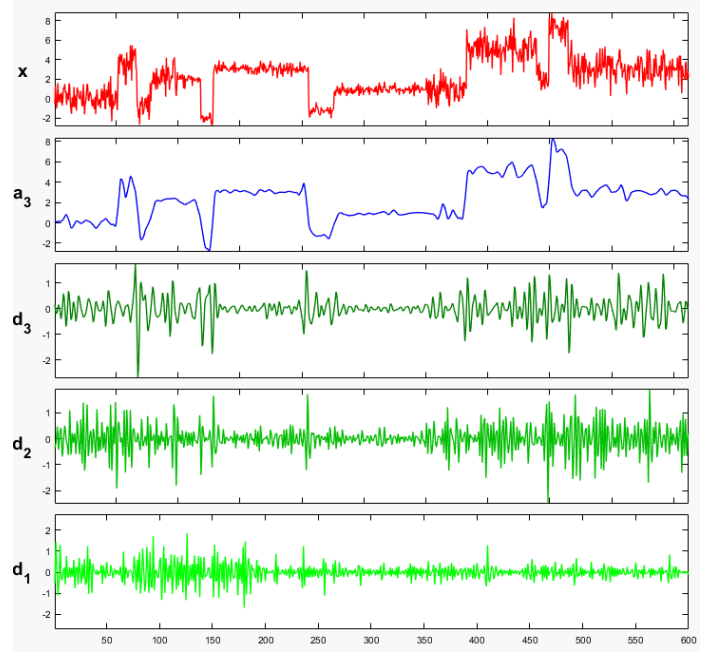


Figura 4: Descomposición de nivel 3 de una señal utilizando una wavelet sym4.

no deseadas. Luego, a la hora de reconstruir la señal, se obtendría una versión sin ruido. Estos parámetros son llamados umbrales y se definirán más adelante.

Teniendo en mente la idea del filtrado, y aún con el inconveniente de que el producto interno para calcular los coeficientes (eq.7 y 9) requiere un alto costo computacional, se plantea una estrategia basada en un algoritmo que relacione las bases ortonormales de wavelets y los bancos de filtros [1]. Dicho algoritmo permite calcular los coeficientes a^j y d^j a partir de los coeficientes de aproximación de una escala anterior de resolución, es decir los α^{j-1} , siendo estos los coeficientes que representan de manera perfecta la señal a analizar. Para llegar a esta relación se parte de la eq.5 donde:

$$A_j(t) = \sum_{k \in \mathbb{Z}} a_k^j \phi_{j,k}(t) \quad (15)$$

$$D_j(t) = \sum_{k \in \mathbb{Z}} d_k^j \psi_{j,k}(t) \quad (16)$$

con

$$a_k^j = \langle A_j(t), \phi_{j,k}(t) \rangle \quad (17)$$

$$d_k^j = \langle A_j(t), \psi_{j,k}(t) \rangle \quad (18)$$

Si se desea obtener a_k^j y d_k^j a partir de A_{j-1} , se realiza el producto interno A_{j-1} y la función de escala y wavelet.

$$\langle A_{j-1}(t), \psi_{j,k}(t) \rangle = \langle A_j(t) + D_j, \psi_{j,k}(t) \rangle \quad (19)$$

$$\langle A_{j-1}(t), \phi_{j,k}(t) \rangle = \langle A_j(t) + D_j, \phi_{j,k}(t) \rangle \quad (20)$$

Operando matemáticamente se llega a que:

$$a_k^j = \sum_{p \in \mathbb{Z}} a_p^{j-1} \langle \psi_{j-1,p}(t), \psi_{j,k}(t) \rangle = \sum_{p \in \mathbb{Z}} a_p^{j-1} \sqrt{2} a_{p-2k} \quad (21)$$

$$d_k^j = \sum_{p \in \mathbb{Z}} a_p^{j-1} \langle \psi_{j-1,p}(t), \phi_{j,k}(t) \rangle = \sum_{p \in \mathbb{Z}} a_p^{j-1} \sqrt{2} b_{p-2k} \quad (22)$$

Donde:

$$\sqrt{2} a_{p-2k} = \langle \phi_{j-1,p}(t), \phi_{j,k}(t) \rangle \quad (23)$$

$$\sqrt{2} b_{p-2k} = \langle \phi_{j-1,p}(t), \psi_{j,k}(t) \rangle \quad (24)$$

Se observa que estos coeficientes son productos internos de funciones ya conocidas, la función wavelet y su función de escala. Estos valores ya fueron calculados y se los asume como conocidos, por lo cual si se consideran como los coeficientes de filtros se puede definir a $LoD = \{[LoD]_n\}_{n \in \mathbb{Z}}$ y a $HiD = \{[HiD]_n\}_{n \in \mathbb{Z}}$ como el filtro pasabajos y un filtro pasaaltos respectivamente.

Por lo cual:

$$[LoD]_n = \sqrt{2} a_{-n} \quad (25)$$

$$[HiD]_n = \sqrt{2} b_{-n} \quad (26)$$

Quedando entonces definidos los coeficientes en un nivel j como:

$$a_k^j = \sum_{p \in \mathbb{Z}} a_p^{j-1} [LoD]_{2k-p} \quad (27)$$

$$d_k^j = \sum_{p \in \mathbb{Z}} a_p^{j-1} [HiD]_{2k-p} \quad (28)$$

Estos coeficientes presentan un aspecto muy similar a la suma de convolución entre el coeficiente del nivel anterior con el filtro correspondiente. Para que esta sea la convolución se introduce la operación de submuestreo, definida como:

$$y = dec(x) \Leftrightarrow y_n = x_{2n} \forall n \in \mathbb{Z} \quad (29)$$

usando las eq.27, 28, 29 y la convolución se llega a:

$$a^j = dec(a^{j-1} * LoD) \quad (30)$$

$$d^j = dec(a^{j-1} * HiD) \quad (31)$$

Lo que permite ver que cada señal se descompone en 2 señales, la de aproximación y detalle de la mitad de la dimension de la señal original.

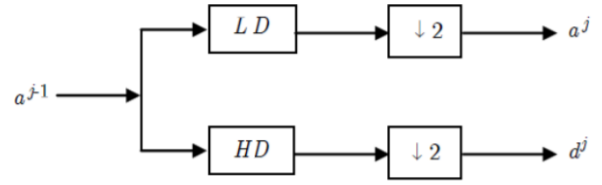


Figura 5: Descomposición con banco de filtros para 1D.

Es posible demostrar que se puede realizar el proceso inverso y reconstruir la señal aproximada, de la misma manera en que se descompuso, la suma de señales convolucionadas con filtros pasabajos y pasaaltos, matemáticamente se puede ver en la siguiente expresión:

$$a_k^{j-1} = \sum_{p \in \mathbb{Z}} a_p^j [LoR]_{k-2p} + \sum_{p \in \mathbb{Z}} d_p^j [HoR]_{k-2p} \quad (32)$$

Donde LoR y HoR son los filtros pasabajos y pasaaltos de reconstrucción.

Como se mencionó anteriormente, las señales a^j y d^j tienen una dimensión menor a la original que se desea reconstruir, por lo que es necesaria “rellenarlas” para que tengan la dimensión correcta.

Este desarrollo es posible extenderlo a dos dimensiones, permitiendo descomponer a las imágenes en cuatro señales diferentes que corresponden a la señal de aproximación y una descomposición de señales de detalles, una para detalles horizontales, verticales y diagonales. Lo mencionado se visualiza en la Fig.6.

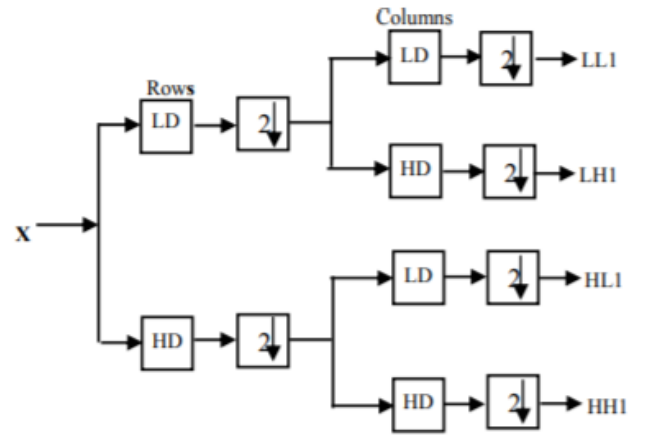


Figura 6: Descomposición con banco de filtros para 2D.

Donde las matrices finales representan:

- LL representa la aproximación.
- LH representa los detalles verticales.
- HL representa los detalles horizontales.
- HH representa los detalles diagonales.

Y la reconstrucción, para el caso de 2D se visualiza en la Fig.7.

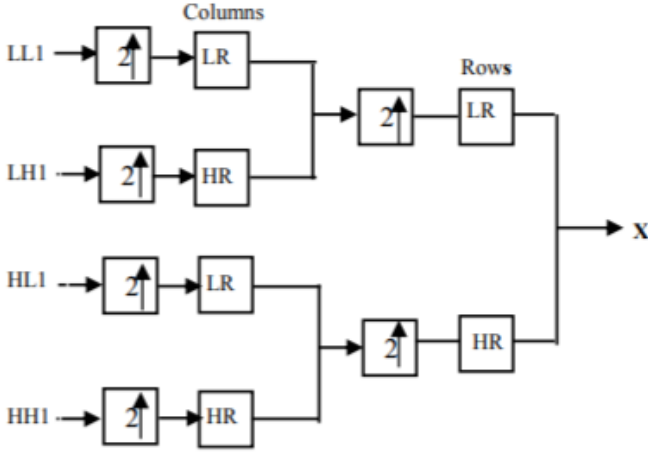


Figura 7: Recomposición con banco de filtros para 2D.

Donde los filtros LR =LoR y HR=HoR hacen referencia en los filtros pasabajos y pasaaltos descritos en la eq.32, y los bloques $2 \uparrow$ representan el rellenado de la señal para obtener la dimensión correcta en la reconstrucción.

II-B. Umbrales

Como se mencionó anteriormente, es necesario determinar parámetros para realizar el filtrado de los coeficientes. Estos parámetros se denominan umbrales. Con el fin de seguir la nomenclatura utilizada en la bibliografía aplicada, a este método se lo denominará umbralización. La umbralización tiene dos características importantes a definir: el tipo de umbralización y las distintas forma de calcular el valor del umbral. En la Fig.8 se observan los tipos de umbralización hard y soft. El valor del umbral se encuentra notado con τ . El tipo de umbralización hard anula todos los coeficientes que estén en el rango $[-\tau, \tau]$ y no modifica el resto. Lo mismo cumple el umbralizado soft con la diferencia que los coeficientes fuera de este rango se ven reducidos linealmente. El tipo hard reproduce de mejor forma los picos y discontinuidades si se piensa a la señal en una dimensión, o por ejemplo, los bordes si se piensa en imágenes. Por otro lado, el tipo soft genera resultados más suaves en la aproximación de la señal, lo que es satisfactorio si no se busca un resultado discontinuo [2]. En las imágenes este método genera un efecto de difuminado justamente para suavizar el resultado final.

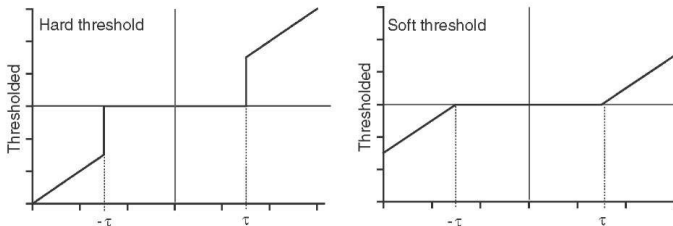


Figura 8: Tipos de umbralización más utilizados.

Ahora, para el cálculo del umbral τ existen distintos tipos de algoritmos. En este informe se estudiarán los siguientes [3]:

- VisuShrink.
- LevelShrink.
- BayesShrink.
- NormalShrink.
- AWT(Adaptative Wavelet Treshholding).

II-B1. VisuShrink: Este algoritmo, más conocido como umbral universal, está representado por la siguiente ecuación:

$$\tau_{univ} = \sigma_n \sqrt{2 \ln(n)} \quad (33)$$

Donde σ es la varianza del ruido y n es la dimensión de la señal. Esta ecuación se deduce a partir de que es sabido que el máximo valor de una señal que siga una distribución gaussiana con media cero y varianza σ^2 , ($x \sim N(0, \sigma^2)$) es menor que el valor $\sigma_n \sqrt{2 \log(n)}$, con una probabilidad que tiende a uno cuando n tiende a infinito. Por lo tanto, este umbral iguala a cero los coeficientes que tengan una alta probabilidad de ser ruido puro [4]. La simpleza del algoritmo lo hace de los más utilizados. Sin embargo, el solo tener en cuenta las características del ruido y la dimensión de la imagen, suele suavizar mucho la imagen. Esto se debe a que elimina muchos coeficientes de información útil al intentar eliminar el ruido presente. Antes de continuar con el siguiente umbral, es necesario mencionar dos aspectos importantes, el primero de ellos es por qué los umbrales dependen de la dimension de la señal. Mientras mayor sea la dimensión, más aumentará la redundancia de la señal. Por lo tanto, hay menos información nueva que la información de base, y desde el punto de vista de wavelet, esto no genera que se aumenten los coeficientes importantes. Por consecuente, existe una relación proporcional entre los coeficientes de ruido y la dimensión de la señal, lo que implica la necesidad de agregarlo al cálculo del umbral [5].

El segundo aspecto de importancia es que para la estimación de la varianza de ruido de la imágenes, se utilizan los coeficientes de mayor frecuencia que son los que presentan mayor energía de ruido. Estos coeficientes quedan representados en las matrices de detalle diagonal, ya que como se mostró en la Fig.6 se puede ver que la matriz de detalle diagonal se forma luego de filtrar dos veces la señal con un filtro pasa alto [6].

II-B2. LevelShrink: este umbral depende de la escala de nivel, a diferencia de VisuShrink, que aplica el mismo umbral a cada nivel de descomposición. Su expresión es la siguiente:

$$\tau_{level} = \sigma_n \sqrt{2 \ln(n)} 2^{-(J-j)/2} \quad (34)$$

Donde J es el número total de niveles de descomposición y j el nivel correspondiente. Si bien esto genera un umbral

que se adapta de mejor manera, sigue sin tener en cuenta los detalles de la señal a analizar.

II-B3. BayesShrink: la base de este umbral es minimizar la inferencia bayesiana. Sus ecuaciones son:

$$\tau_{bayes} = \begin{cases} \sigma_n^2 / \sigma_s & \text{si } \sigma_n^2 < \sigma_y^2 \\ \max(|y_i|) & \text{si } \sigma_n^2 \geq \sigma_y^2 \end{cases} \quad (35)$$

Donde σ_n^2 es la varianza del ruido, σ_y^2 es la varianza de la señal original y σ_s es la varianza de la señal pura que se estima con la siguiente ecuación:

$$\sigma_{sk} = \sqrt{\max \left(\left(\frac{1}{\text{size}(H_k)} \sum_{y_i \in H_k} y_i^2 \right) - \sigma_n^2, 0 \right)} \quad (36)$$

Donde H_k es la matriz de detalle a analizar. Este algoritmo tiene como ventaja respecto a los anteriores que utiliza información de la señal a analizar pero es necesario conocer información de la imagen original (sin ruido).

II-B4. NormalShrink: este algoritmo suple la desventaja de BayesShrink ya que no se necesita información de la señal sin ruido y presenta todas las ventajas incorporadas de utilizar información de la señal a analizar. Se lo define como:

$$\tau_{normal} = \frac{\beta_k \sigma_n^2}{\sigma_s} \quad (37)$$

Donde:

$$\beta_k = \sqrt{\frac{\ln(\text{size}(H_k))}{J}} \quad (38)$$

Y σ_s es el desvío estándar de la señal de cada nivel.

II-B5. AWT(Adaptative Wavelet Treshholding): por último, se presenta el algoritmo AWT el cual depende de la media aritmética y de la media geométrica de los coeficientes de la señal en cada nivel de análisis. Se lo define como:

$$\tau_{AWT} = c\sigma_n - |AM - GM| \quad (39)$$

Donde AM es la media aritmética, GM es la media geométrica y $c = 2^{J-j}$.

II-C. Estadísticos

Con el fin de poder comparar los resultados se definen los siguientes estadísticos:

- MSE(mean square error) definido por

$$MSE = \frac{1}{nm} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \quad (40)$$

- PSNR(Peak Signal to Noise Ratio) definido mediante la ecuación

$$PSNR = 20\log(\text{máx } I) - 10\log(MSE) \quad (41)$$

- SSIM(structural similarity) que se define por

$$SSIM(I, K) = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)} \quad (42)$$

Donde n y m son las dimensiones de la imagen, I es la imagen original sin ruido y K es la imagen filtrada. Para el SSIM μ_I es la media de la imagen original y μ_K es la media de la imagen a comparar. De manera análoga para σ_I y σ_K . El valor de σ_{IK} es la covarianza y los valores c_1 y c_2 son constantes.

III. Implementación

Para llevar a cabo el filtrado de ruido en imágenes es necesario realizar los siguiente pasos:

- 1) Calcular el nivel adecuado de descomposición según la wavelet elegida, y la dimensión de la imagen. Si bien se puede elegir un nivel arbitrario de descomposición, aumentarlo por encima del nivel óptimo no mejora el resultado. Esto solo produce un filtrado de la información útil de la imagen, suavizando los detalles.
- 2) Realizar los siguientes pasos:
 - i. Descomponer la imagen en sus matrices de detalle (horizontal, vertical y diagonal).
 - ii. Calcular el desvío estándar del ruido a partir de la matriz de detalle diagonal para cada nivel.
 - iii. Filtrar los coeficientes de la matrices de detalle obtenidas en el paso i., variando alguno de los siguientes parámetros:
 - Tipo de umbralización : hard o soft.
 - Algoritmos de umbralización.
 - Cantidad de niveles de descomposición.
 - Tipo de wavelet.
- 3) Reconstruir la imagen a partir de los coeficientes de detalle filtrados en los pasos previos y la matriz de aproximación.

IV. Resultados

A continuación, se presentan los resultados obtenidos en las imágenes de prueba, mostradas en la Fig.9. Todas se modificaron con un ruido gaussiano de $\sigma = 0,3$. (Las imágenes reales se encuentran en el anexo VI-A)

Luego de pruebas con diferentes wavelet, algoritmos de umbralización y variando la cantidad de niveles, se llegó a que el mejor resultado visual se obtuvo para:

- Wavelet = Sym8.
- Umbral = NormalShrink soft.
- Cantidad de niveles = 6.

Obteniéndose las imágenes filtradas y reconstruidas presentadas en la Fig.10.

Para corroborar este resultado, se presentan las tablas comparativas de las distintas combinaciones de parámetros descritos en la sección III y sus imágenes asociadas. Dichos resultados se organizan de la siguiente manera:

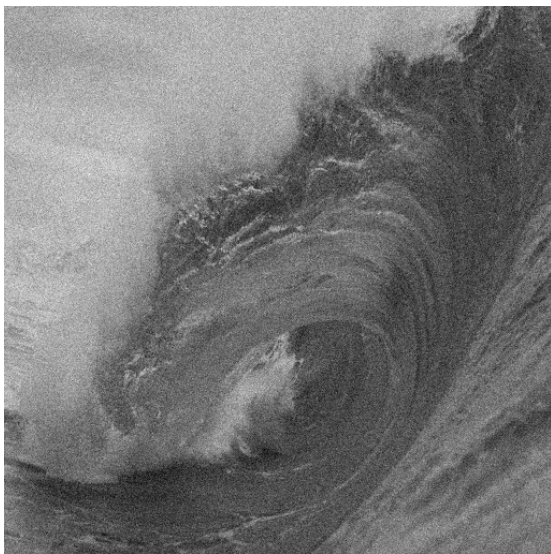
- IV-A Resultados obtenidos al variar la cantidad de niveles de descomposición.
- IV-B Resultados obtenidos para los dos tipos de umbralización.



(a) Lenna.



(b) House.



(c) Wave.

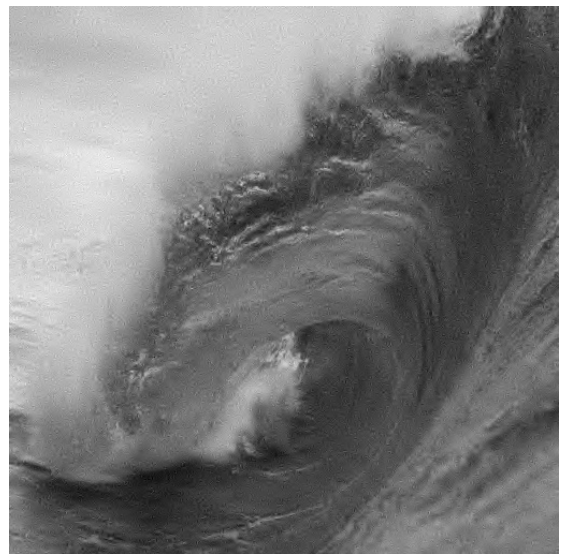
Figura 9: Imágenes de prueba escogidas.



(a) Lenna.



(b) House.



(c) Wave.

Figura 10: Imágenes filtradas.

imagen\niveles	Sin filtrar	1	2	4	6
Lenna	17.65	23.92	27.03	22.29	22.29
House	19.87	22.90	25.58	24.57	23.51
Wave	18.63	23.34	26.70	24.71	24.65

Tabla 1: PSNR utilizando wavelet Sym8, umbral normal-shrink, método soft.

imagen\niveles	Sin filtrar	1	2	4	6
Lenna	0.518	0.742	0.856	0.847	0.808
House	0.620	0.806	0.882	0.839	0.814
Wave	0.586	0.761	0.839	0.820	0.803

Tabla 2: SSIM utilizando wavelet Sym8, umbral normal-shrink, método soft.

IV-C Resultados obtenidos para los distintos algoritmos de umbralización.

IV-D Resultados obtenidos para cada tipo de Wavelet.

Los estadísticos que se utilizaron para comparar son el PSNR y el SSIM. El PSNR define la relación entre la máxima energía posible de una señal y el ruido que la afecta. Mientras menor es su valor, mayor es la cantidad de ruido en la imagen. Por otra parte, el SSIM mide la similitud entre dos imágenes. El resultado varía entre 0 y 1, donde $SSIM = 1$ indica que las imágenes son iguales. La ventaja de este estadístico recae en que los resultados coinciden con la percepción del ojo humano, lo que no sucede con el PSNR.

IV-A. Resultados obtenidos al variar la cantidad de niveles de descomposición.

A partir de los estadísticos vemos que el nivel 2 brinda mejores resultados numéricos. Sin embargo, si vemos las imágenes el nivel 6 presenta una mejora visual en cuanto al ruido presente.

IV-B. Resultados obtenidos para los dos tipos de umbralización.

Si analizamos la imagen resultante al utilizar el tipo de umbralización Hard, vemos que esta pierde intensidad de color en los píxeles. Esto se debe a las zonas discontinuas

imagen\nivel	Sin filtrar	soft	hard
Lenna	17.65	27.03	21.41
House	19.87	25.58	20.20
Wave	18.63	26.70	20.85

Tabla 3: PSNR utilizando wavelet Sym8, dos niveles de descomposición, umbral normalshrink.

imagen\nivel	Sin filtrar	soft	hard
Lenna	0.518	0.856	0.757
House	0.620	0.882	0.789
Wave	0.586	0.839	0.755

Tabla 4: SSIM utilizando wavelet Sym8, dos niveles de descomposición, umbral normalshrink.



(a) Nivel 1.

(b) Nivel 2.



(c) Nivel 4.

(d) Nivel 6.

Figura 11: Imágenes filtradas por nivel.



(a) Imagen filtrada soft.

(b) Imagen filtrada hard.

Figura 12: Imágenes filtradas con los métodos soft y hard.

que presenta (fig. 17b). En cambio el tipo Soft suaviza de forma mas suave generando una perdida menor de intensidad, esto también se ve reflejado en los mejores resultados de los estadísticos.

	Sin filtrar	universal	bayes	level	normal	awt
Lenna	17.65	25.86	25.71	25.40	27.03	25.24
House	19.87	22.91	23.32	23.19	25.58	23.41
Wave	18.63	26.74	26.70	26.86	26.70	25.56

Tabla 5: PSNR utilizando wavelet Sym8, dos niveles de descomposición, método soft.

	Sin filtrar	universal	bayes	level	normal	awt
Lenna	0.518	0.848	0.847	0.849	0.856	0.838
House	0.620	0.851	0.850	0.857	0.882	0.849
Wave	0.586	0.830	0.829	0.833	0.839	0.823

Tabla 6: SSIM utilizando wavelet Sym8, dos niveles de descomposición, método soft.

imagen\wavelet	Sin filtrar	haar	db4	sym8
Lenna	17.65	23.44	25.19	27.03
House	19.87	26.38	24.78	25.58
Wave	18.63	24.67	26.87	26.70

Tabla 7: PSNR utilizando dos niveles de descomposición, umbral normalshrink, método soft.

IV-C. Resultados obtenidos para los distintos algoritmos de umbralización.

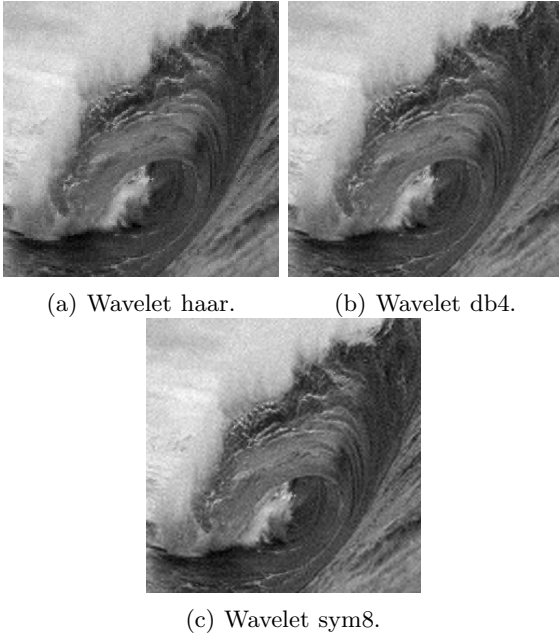


Figura 13: Imágenes filtradas con distintas wavelet.

IV-D. Comparación con otros métodos

En esta sub-sección comparamos el filtrado de wavelet con dos filtros ya implementados para la supresión de ruido.

imagen\wavelet	Sin filtrar	haar	db4	sym8
Lenna	0.518	0.819	0.853	0.856
House	0.620	0.848	0.875	0.882
Wave	0.586	0.805	0.836	0.839

Tabla 8: SSIM utilizando dos niveles de descomposición, umbral normalshrink, método soft.

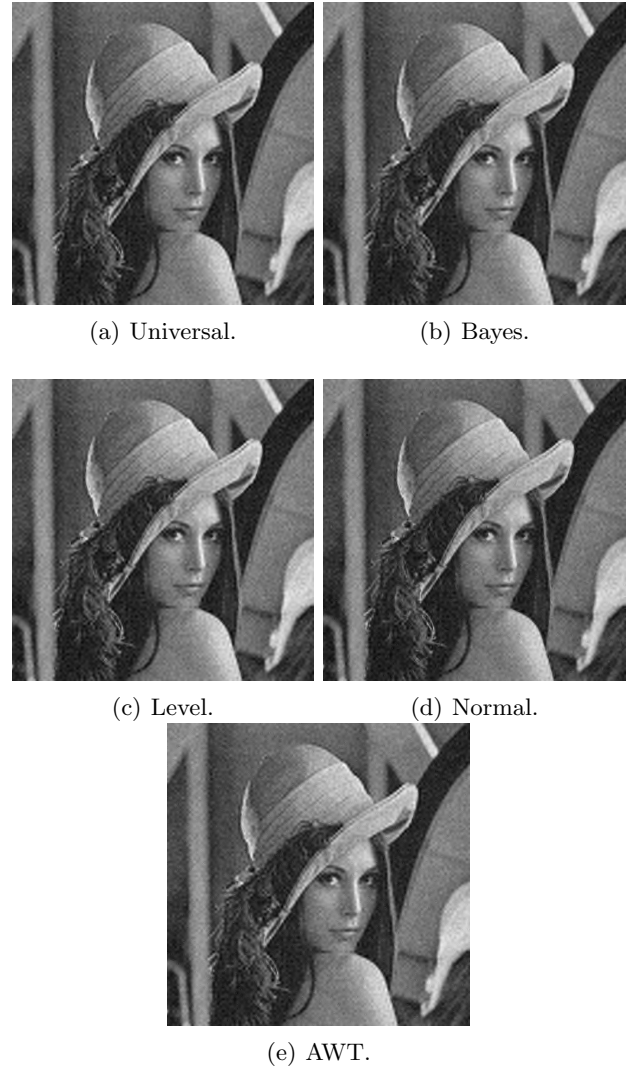
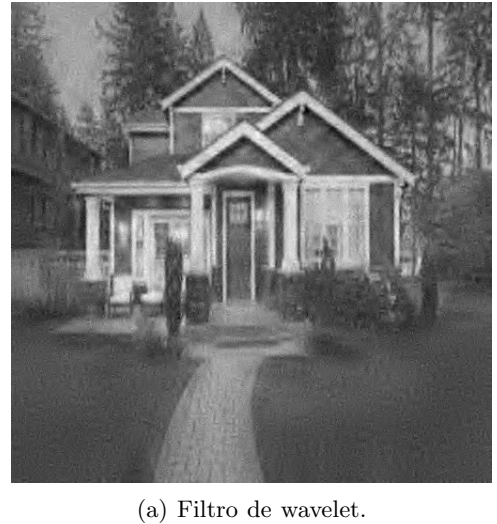


Figura 14: Imágenes filtradas con distintos algoritmos de umbralización.



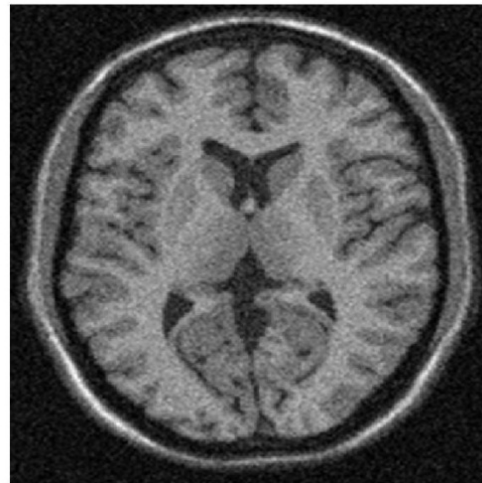
	Sin filtrar	wavelet	wiener	gaussian
Lenna	23.10	23.83	26.30	26.14
House	24.80	25.07	28.28	27.99
Wave	24.21	24.33	27.00	26.86

Tabla 9: PSNR para los distintos métodos de filtrado.

	Sin filtrar	wavelet	wiener	gaussian
Lenna	0.647	0.870	0.843	0.835
House	0.740	0.906	0.895	0.886
Wave	0.693	0.887	0.862	0.853

Tabla 10: SSIM para los distintos métodos de filtrado.

Al comparar el PSNR se observa que el filtro de Wiener es mejor en los tres casos, pero al ver la Tab.10 se puede deducir que el filtrado con wavelet es mejor visualmente. Esto se debe, a como se menciono anteriormente, a que el SSIM es mejor estadístico para comparar imágenes.



(a) Imagen sin filtrar.



(b) Imagen filtrada.

Figura 16: Resonancia magnética.

IV-E. Aplicación a imágenes reales

Por último, con el fin de verificar el funcionamiento del filtrado, utilizamos imágenes que presentan ruido real. En la figura 17b se observa una foto tomada en el año 1925 al príncipe de gales. Esta imagen presenta ruido debido a la tecnología de esa época. Por otro lado, en la imagen 17b se observa una tomografía de un cerebro, la cual presenta un ruido particular llamado ruido Rician.



(a) Imagen sin filtrar.



(b) Imagen filtrada.

Figura 17: Foto del príncipe de Gales tomada en el año 1925.

En los resultados se observa que el filtro es capaz de eliminar el ruido en imágenes reales de buena forma, siempre teniendo en cuenta el difuminado que se le agrega a raíz de esto. En la imagen del príncipe se logró filtrar el ruido con mayor o menor intensidad según la cantidad de ruido presente en cada segmento de ella. Por otro lado, en la tomografía, el filtro fue capaz de dar buenos resultados para un ruido que presenta una distribución distinta a la distribución gaussiana.

V. Conclusiones

A partir de los resultados obtenidos se puede concluir de manera general que el método de filtrado de ruido utilizando la transformada de wavelet es una herramienta útil. Viendo las Fig.9 y 10 se observa que el resultado de filtrar el ruido agregado artificialmente es aceptable. Si bien se presenta un difuminado que provoca pérdidas de detalles, esto es normal en la mayoría de métodos que utilizan umbrales, justamente por la necesidad de

eliminar el ruido. Es importante recalcar la elección del nivel de descomposición y su relación con los estadísticos. En las Tab.1 y 2, el nivel 2 es el que brinda mejores resultados numéricamente. Sin embargo, a la hora de ver las imágenes, el nivel 6 muestra menor cantidad de ruido que la imagen del nivel 2. Esta discrepancia se puede deber a que si bien la imagen del nivel 2 presenta ruido visible, la intensidad de los píxeles no varía tanto respecto a la original como si lo hace el nivel 6, lo que genera en la forma de calcular los estadísticos un mejor resultado. Por otro lado, la razón por la cual el nivel seis elimina mayor cantidad de ruido es que el nivel de descomposición aumenta y esto permite filtrar una mayor cantidad de coeficientes de detalles. Se puede decir entonces, que se elimina una cantidad superior de ruido a costa de perder detalles. La wavelet utilizada sym8 es congruente que brinde mejores resultados ya que la forma de este tipo de onda se adapta de manera óptima a los detalles de la imagen por tener una forma más suave que la wavelet haar y db4. Por último, el algoritmo de umbralización normalShrink brinda un mejor filtrado debido a que utiliza información de la imagen y no solo de la varianza del ruido.

Además, es interesante analizar el resultado de la Tab.7. Para la wavelet haar la casa obtiene mejor PSNR, lo que es coherente por la forma de la haar y por los bordes rectos que presenta la casa. La forma discontinua y “dura” de esta wavelet no suaviza los bordes lo que permite que sobrevivan al filtrado. Sin embargo, los detalles de la imagen que no posean estas características se ven suprimidos, y genera una mayor discrepancia con la imagen original. Este resultado, es congruente con el de la SSIM (Tab.8).

Ahora, analizando los resultados con los métodos de filtrado gaussiano y wiener, se ve en la Fig.15 que estos dos métodos no quitan tanto el ruido como el método de la transformada wavelet. Sin embargo, presentan un menor difuminado. Si se observa la Tab.10, los resultados de la transformada wavelet dan un mejor SSIM comparado con wiener y gaussiano. Concluyendo, la transformada de wavelet utilizada para filtrar ruido actúa de manera óptima al compararlo con métodos especializados en el filtrado de ruido.

Por último, cuando se implementó el filtrado en imágenes reales que contengan ruido, es decir que no se le agregó ruido de forma sintética, el resultado es aceptable. En la imagen del Príncipe de Gales es destacable que en comparación con las imágenes sintéticas esta no posee el mismo nivel de ruido en todas sus regiones y por lo tanto en las zonas donde es mayor el ruido vemos un mayor difuminado en la imagen resultante. Además, en el caso particular de la resonancia, ésta presenta un tipo de ruido particular llamado ruido Rician y aún así el filtro fue lo bastante robusto para adaptarse brindando un buen resultado.

VI. Anexo

VI-A. Imágenes reales



(a) Lenna.



(b) House.



(c) Wave.

Figura 18: Imágenes originales.

Referencias

- [1] Michel Misiti-Yves Misiti-Georges Oppenheim-Jean-Michel Poggi - "Wavelets and their Applications"
- [2] Cristóbal E. Vega González - "Un metodo automático para la elección de la base de ondícula y la selección del umbral para la estimacion de señales"
- [3] José Antonio Guerrero Díaz de León - "Conjuntos Completos de Filtros Para la Eliminación de Ruido" en Imágenes
- [4] Fei Xiaoa,Yungang Zhanga - "A Comparative Study on Thresholding Methods in Waveletbased Image Denoising"
- [5] Algoritmos para reducción de ruido en señales - <http://catarina.udlap.mx>
- [6] D.Gnanadurai, and V.Sadasivam - "An Efficient Adaptive Thresholding Technique for Wavelet Based Image Denoising"