CAPÍTULO 4

ALGORITMOS PARA REDUCCIÓN DE RUIDO EN SEÑALES

4.1 Introducción

Los algoritmos para reducir ruido en señales basados en el análisis utilizando wavelets fueron iniciados, principalmente por Donoho y Johnstone en los Estados Unidos, y por Kerkyacharian y Picard en Francia. Mayer considera que este tema es una de las más significativas aplicaciones de las wavelets [TAS00], [HER03]. Se pueden encontrar varias aplicaciones, tales como, la restauración de señales de 1-D y 2-D como en el caso de señales medicas, imágenes astronómicas y por supuesto en comunicaciones [ODE96], [MUK02]. Dado que este objetivo se puede cumplir de diferentes formas, por ello es que se han desarrollado diferentes algoritmos que permiten analizar una señal y posteriormente sintetizarla reduciendo el ruido que presentaba originalmente. El objetivo de este capitulo es presentar la metodología que emplea los algoritmos para reducir el ruido de las señales en este caso de imágenes usando la teoría de wavelets y exponer los algoritmos de mayor relevancia.

4.2 Umbral Suave y Umbral Duro.

Para iniciar será expuesta la variedad de algoritmos que existen, a cuáles podemos tener acceso es decir cuáles pueden ser implementados con la ayuda de Matlab® y posteriormente serán mencionados los que son más comúnmente usados para la reducción de ruido en imágenes.

Para comenzar, es necesario saber que los algoritmos se pueden clasificar en métodos de reducción de ruido lineales y no lineales [HES97]. Los lineales son aquellos que son independientes del tamaño empírico de los coeficientes de la señal. Basan su realización en el hecho de que el ruido se encuentra principalmente en los coeficientes de la escala fina, de este modo elimina todos los coeficientes con una escala más fina que cierto umbral (λ). Si los coeficientes wavelet están dados por $\{d_{j,k}\}$ obteniéndose la siguiente relación [MAT02]:

$$d_{j,k}\begin{cases} 0, & j \geq \lambda \\ d_{j,k}, & j < \lambda \end{cases} \tag{43}$$

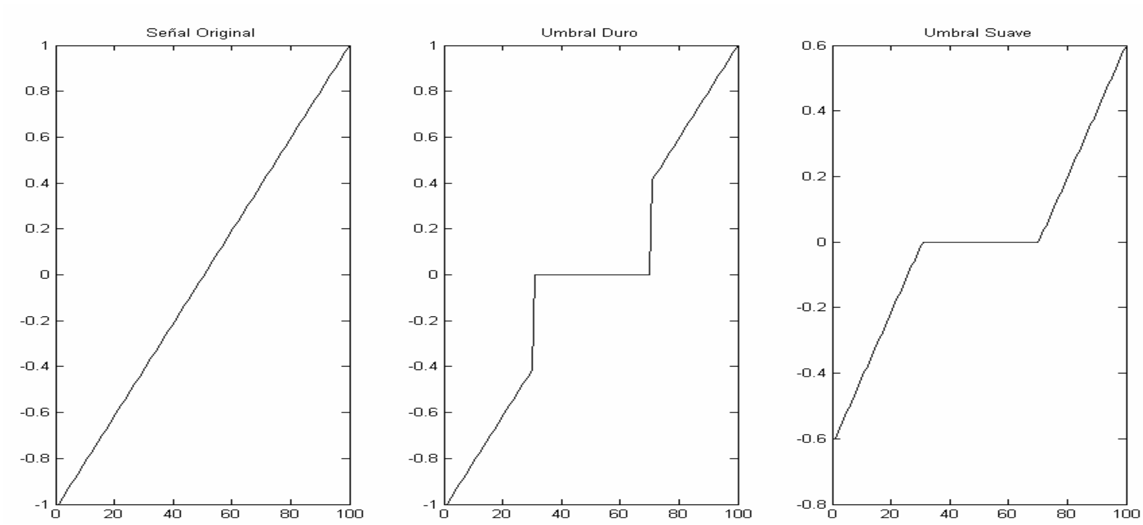
Mientras tanto, los métodos no lineales se basan en la idea de que el ruido se encuentra en cada coeficiente y esta distribuido sobre todas las escalas. Se pueden aplicar en dos versiones, la del umbral suave y la del umbral duro [MAT02], es decir: el umbral duro elimina los coeficientes que se encuentren por debajo del umbral que fue elegido. Esto es:

$$s(x)\begin{cases} s(x), & |x| > \lambda \\ 0 \quad , & |x| \leq \lambda \end{cases} \tag{44}$$

Donde $s(x)$ es la señal que esta siendo analizada y λ es el umbral que se ha elegido previamente. El umbral suave, también conocido como Shrinkage Denoisig [QUI00], es una extensión del umbral duro, eliminando los elementos de valor menor al umbral y enviando a un valor determinado el resto de ellos. Este proceso se realiza de la siguiente manera:

$$s(x)\begin{cases} Sgn(x)\left(|x| - \lambda\right), & |x| > \lambda \\ 0 \qquad\qquad\qquad , & |x| \leq \lambda \end{cases} \tag{45}$$

Como podemos ver en la Figura 4.1 se muestra un ejemplo en el que una función es tratada con ambos métodos, nótese cómo el umbral duro crea discontinuidades en $s(x) = \pm\lambda$, mientras que el umbral suave no lo hace.



a) b) c)

Figura 4.1. Umbral Duro y Umbral Suave de una señal

Para llevara a cabo la selección de los algoritmos que se exponen a continuación, se ha tomado en cuenta la herramienta que será empleada en la simulación que es el Toolbox de Matlab® llamado Wavelet Toolbox; además de la antigüedad de cada algoritmo así como su forma de trabajo, ya que son la base para otros que trabajan bajo el mismo principio.

4.3 Análisis de Algoritmos

Para iniciar con el análisis de algoritmos debemos conocer la forma en que trabajan para lograr reducir el ruido de una señal. Para ello debemos partir de lo siguiente.

$$p(n) = s(n) + \sigma e(n) \tag{46}$$

Donde $s(n)$ es la señal de interés, $e(n)$ es ruido blanco Gaussiano con una media igual a cero, estacionario y de segundo orden, lo que significa que todas las componentes de ruido tendrán la misma desviación estándar, lo que se refleja en el estimador σ [JAN99] y $p(n)$ es la señal resultante de la mezcla de la señal y el ruido, el objetivo de todos los algoritmos que serán estudiados en éste capítulo es recuperar la señal $s(n)$ suprimiendo el ruido presente en la señal $p(n)$ o bien lograr una estimación lo más cercana posible a la señal $s(n)$.

El procedimiento para eliminar el ruido de la señal se divide en tres etapas [TAS00] para las cuales se requiere definir cierta nomenclatura. En este caso, este procedimiento es presentado para el análisis con la DWT, pero puede ser aplicado para cualquiera de sus variantes e incluso este procedimiento también es válido para un ser llevado a cabo con la CWT; como veremos en el siguiente capítulo. Para referirnos a la DWT, se ha definido $S(a,b) = DWT\{s(n)\}$, y su inversa la IDWT será $s(n) = IDWT\{S(a,b)\}$. De este modo la metodología es [HER03]:

► Descomposición. Elegir una wavelet $w_{a,b}$ con la cual se va a trabajar y elegir un nivel de descomposición J que representa los niveles a calcular o la resolución.

Posteriormente calcular la DWT de la señal $s(n)$. Esto es:

$$S(a,b) = DWT\{s(n), w_{a,b}, J\} \tag{47}$$

► Umbral. Ésta es la parte medular del proceso, ya que el umbral lo podemos definir de a cuerdo con [JAN99] como:

Un umbral puede ser visto como un parámetro suavizador que controla entre suavidad y apariencia de la señal.

Para cada coeficiente de 1 a J seleccionar un umbral λ y aplicar umbral suave o duro a cada coeficiente. Con esto se eliminarán los coeficientes de más baja energía [CRI96]. Para esto definiremos un operador $u\{\ \}$ que determine el cálculo de umbral λ (48) y otro operador $D\{\ \}$ que haga el proceso de reducción de ruido (49) para mantener los coeficientes wavelet limpios $\tilde{S}(a,b)$. Estos son [HER03]:

$$\lambda = u\{S(a,b)\} \tag{48}$$

$$\tilde{S}(a,b) = D\{S(a,b), \lambda\} \tag{49}$$

► Reconstrucción. Calcular la IDWT utilizando las modificaciones que se han hecho a los coeficientes.

$$\tilde{s}(n) = IDWT\left\{\tilde{S}(a,b)\right\} \tag{50}$$

Ahora que se conoce la forma general en la que trabajan los algoritmos de reducción de ruido en señales es necesario señalar la importancia que tiene la elección del tipo de umbral y la forma en que se calcula dicho umbral. A continuación, se llevará a cabo un análisis profundo e individual de cómo es que cada método realiza el cálculo del umbral, que es la segunda fase del procedimiento mostrado anteriormente, que es en realidad la parte que hace la diferencia entre cada método. Tal y como lo señala Krim [KRI99], pues el cálculo de umbral de las wavelets es un problema de valoración de orden, entre la exactitud del balanceo contra la captura de un porcentaje de la señal a tanto como sea posible, dejando fuera de ella el ruido tanto como sea posible.

Existe una gran variedad de métodos para elegir el valor del umbral λ y pueden ser agrupados en dos categorías que son los umbrales globales y los umbrales dependientes de nivel ó recursivos. Los primeros son aquellos que eligen un valor fijo de λ para ser aplicados globalmente a todos los coeficientes de wavelets $d_{j,k}$ obtenidos de la transformación $S(a,b)$ conforme fue explicado en el Capítulo 3 para el caso de la CWT y en el Capítulo 2 para la DWT. Los umbrales dependientes de nivel son aquellos que tienen la posibilidad de un valor diferente de umbral λ_a para cada nivel de resolución (escala) dado [ANT01].

Para iniciar con ésta descripción debemos saber que todos los umbrales requieren de una estimación de los niveles de ruido para lo cual no se utiliza la desviación estándar usual de los valores de los datos. Este algoritmo fue propuesto por Donoho y Johnstone en 1994 [DON94] de acuerdo con [SAR03]. Lo que sus autores consideran es un estimador en el dominio de las wavelets y sugieren un estimador robusto que está basado en la desviación media absoluta y toma en cuenta sólo los coeficientes empíricos de wavelets en los niveles de escala más finos, ya que estos son los que concentran la mayor cantidad de ruido [ANT01], [HER03]. Éste estimador de los niveles de ruido está dado por:

$$\sigma = \frac{\text{median}\left(\left\{\left|S(J-1,b)\right| : b = 0,1,...2^{J-1}-1\right\}\right)}{0.6745} \tag{51}$$

Donde J representa el número de niveles de decimado. Este estimador se aplica en los diferentes métodos para reducir el ruido. Ahora que se han asentado las bases, se darán a conocer cada uno de los algoritmos y la forma en los que trabajan sobre la señal.

4.4 Algoritmo Minimax

Siguiendo el orden cronológico, de acuerdo con Antoniadis [ANT01] y [JAN99], el primer algoritmo en surgir fue el Minimax. Utiliza el principio Minimax el cual se utiliza en estadística para diseñar estimadores. Ya que la reducción de ruido de la señal puede ser vista como el estimador de una regresión de una función desconocida, el umbral Minimax λ_M es la opción que obtiene el mínimo de un conjunto de valores dados, del máximo error cuadrático medio, como lo llaman algunos autores [HER03].

Éste algoritmo fue desarrollado por Donoho y Johnstone en 1994 [DON94] y es un umbral que minimiza el término constante en el riesgo de la estimación de una función. El umbral Minimax propuesto, que depende del tamaño n de las muestras es:

$$\lambda_M = \sigma\lambda_n^* \tag{52}$$

Donde λ_n^* es definido como el valor de λ para el cual se cumple:

$$\Lambda_n^* := \inf_\lambda \sup_d \left\{ \frac{R_\lambda(d)}{n^{-1} + R_{\text{oracle}}(d)} \right\} \tag{53}$$

Donde $R_\lambda(d) = E(\delta_\lambda(d) - d)^2$ y $R_{\text{oracle}}(d)$ es el riesgo ideal calculado con ayuda de uno de los dos oráculos calculados. Uno es la Proyección Diagonal Lineal (DLP), el cual nos dice "mantener" o "aniquilar" cada coeficiente de la wavelet. Y el otro es el "*Shrinker*" Diagonal y Lineal (DLS), que nos dice cuánto encoger cada coeficiente de wavelet [ROS05]. El riesgo ideal de los oráculos es dado por [ROS05]:

$$R_{\text{oracle}}^{\text{DLP}}(d) := \min(d^2, 1) \tag{54}$$

$$R_{\text{oracle}}^{\text{DLS}}(d) := \frac{d^2}{d^2 + 1} \tag{55}$$

Donde $\min()$ denote el valor mínimo de d^2.

Hasta este punto llegó el trabajo desarrollado por Donoho y Johnstone [DON94], pero dejaron establecidas las bases para posteriormente se siguiera investigando sobre ésta línea. Para la implementación, se ha buscado en Matlab® este algoritmo y de acuerdo a [MAT02] el programa utilizado para el cálculo del umbral con éste algoritmo funciona de la siguiente manera:

$$\lambda_M = \begin{cases} 0 & , \ n \leq 32 \\ 0.3936 + 0.1829\left(\frac{\log(n)}{\log 2}\right) & , \ n > 32 \end{cases} \tag{56}$$

En este caso lo que se hace es considerar la longitud de la señal con ruido y en caso de que esta sea menor a 32 muestras el umbral es cero en caso contrario se aplica la ecuación correspondiente para calcular el umbral indicado [HER03].

De acuerdo con la clasificación explicada previamente este algoritmo corresponde a un umbral global, ya que el valor de λ_M es el mismo para todas las muestras analizadas.

4.5 Algoritmo de Umbral de Forma Fija

También fue propuesto por Donoho y Johnstone [SAR03] y ha recibido varios nombres como: Umbral Universal, Umbral de forma fija Visushrink, dependiendo de cada autor.

Surge como una alternativa de uso de los umbrales minimax. En ésta ocasión sus creadores utilizan una forma fija de umbral, la cual está determinada por:

$$\lambda_{UNI} = \sigma\sqrt{2\log(n)} \tag{57}$$

Donde el umbral λ_{UNI} es función de n que es la longitud de la señal a limpiar y σ es el estimador de los niveles de ruido descrito en la ecuación (51). Esto ocasiona que el umbral varíe de acuerdo con la longitud de la señal. Para una señal de mayor tamaño el umbral será mayor que para una señal de menor tamaño como menciona Hess en [HES97]. El umbral universal puede ser determinado sólo dependiendo de la longitud de la señal y de la desviación estándar σ.

Este umbral universal de la ecuación (57) es sustancialmente más grande que el minimax, para cualquier valor particular de n. Como resultado la reconstrucción incluirá menos coeficientes; lo que representa una estimación que es el mejor balance para el proceso de limpieza, ya que distribuye mejor la estandarización de los coeficientes que en la estimación minimax [ANT01]. Otra característica importante es que en éste umbral tiene una alta probabilidad que garantiza que todas las muestras de DWT en las cuales la función analizada es exactamente cero serán estimadas como cero [HER03]. Éste algoritmo también es catalogado dentro de los métodos de umbral global.

Una cuestión importante que hay que aclarar es el por qué el umbral depende de la longitud de la señal. La explicación radica en que si agregamos más muestras aumenta la redundancia de la señal; esto es, hay menos información nueva en las muestras nuevas de la que ya hay en las primeras muestras [HER03]. En el dominio wavelet escala y traslación, esto significa que el número de los coeficientes importantes está creciendo escasamente y toda la información está concentrada en un número limitado de coeficientes. De este modo, el número total de coeficientes de ruido es proporcional a n al igual que el número de coeficientes que contienen la información, sólo que se guarda una relación que está determinada, precisamente por el umbral universal [JAN99].

Para el caso práctico se tiene que Matlab® [MAT02] utiliza exactamente la misma fórmula, pero en el software se toma la estimación de niveles de ruido $\sigma = 1$. Con esto tenemos que, si se considera como otros autores [TAS00] [QUI00] y [KRI99], entre otros σ es la desviación estándar de ruido, lo que simplifica aún más esta regla y por lo tanto el trabajo.

4.6 Algoritmo Rigorous SURE

Hasta ahora hemos analizado dos métodos que coinciden en ser de umbral global, pero éste algoritmo propuesto por Donoho y Johnstone [SAR03], introduce un esquema que utiliza los coeficientes de la wavelet de tal forma que para cada nivel a de escala tenga un umbral λ_a. Éste algoritmo o método recibe también el nombre de SureShrink porque se basa en la aplicación del Stein's Unbiased Risk Estimate (SURE), que es un estimador de umbral suave. También se conoce como Función de pérdida cuadrática. Lo primero que hace es obtener una estimación de riesgo para un valor de umbral en particular. Reducir el riesgo o pérdida en ese valor nos da una selección del valor del umbral [ROS05]. El valor final del umbral después del procedimiento matemático es [ROS05]:

$$\lambda_{SURE} = \arg\min_{0<\lambda<\lambda_{UNI}} SURE\left(\lambda, \frac{S(a,b)}{\sigma}\right) \qquad (58)$$

Donde $a = a_0,...,J-1,\ b = 0,1,...,2^J - 1, \lambda_{UNI}$ fue definido en la ecuación (61) $S(a,b)$ son los coeficientes de la DWT y σ está definido por la ecuación (51). El operador $\arg\min$ denota que se utilizarán los elementos menores del estimador SURE cuya relación matemática es la siguiente [ROS05]:

$$SURE(\lambda;S) = n - 2\clubsuit\{i : |S_i| \leq \lambda\} + \left[\min\left(|S_i|, \lambda\right)\right]^2 \qquad (59)$$

En donde ♣ denota la cardinalidad del conjunto $\left\{ i : \left| S_i \right| \leq \lambda \right\}$. Para el caso de los coeficientes DWT la variable S se cambia por el coeficiente $\frac{S(a,b)}{\sigma}$.

De acuerdo con [ANT01], este tipo de umbral tiene una desventaja, para los casos en que los coeficientes wavelet están muy dispersos, el ruido contribuye al perfil del algoritmo SURE cambiando el valor de la señal en las coordenadas donde la señal era igual a cero.

Este problema fue resuelto por los mismos autores haciendo un método que conjuntara los trabajos realizados previamente y que compartiera las características de ambos algoritmos, haciendo referencia al Umbral Universal y al SureShrink. De esta mezcla surge el método Heuristic SURE ó Hybrid SURE, del cual se encontrarán más detalles en [ANT01], [HER03].

Para la implementación, el Wavelet Toolbox de Matlab® también contiene un programa que contempla este método de cálculo de umbral que se llamará $SURE_M$, el cual está implementado de la siguiente manera [MAT02]:

$$SURE_M = n - 2\left\{1 : n\right\} + \frac{\sum_{a=0}^{n} \left| S(a,b) \right|^2 + \left\{ n-1 : 0 \right\} \left| S(a,b) \right|^2}{n} \tag{60}$$

$$\lambda_{SURE} = \sqrt{\left| S(a,b) \right|^2_{\min(SURE_M)}} \tag{61}$$

Se pueden notar las diferencias que existen con respecto a lo encontrado en la literatura sin embargo las diferencias de todos los algoritmos implementados en Matlab con respecto a la teoría, radica en que los algoritmos del programa suponen que se trabajará con ruido blanco Gaussiano y por ende se tiene la forma $N(0,1)$, donde se denota la distribución normal, la media es igual a cero y la desviación estándar $\sigma = 1$ por lo que muchos cálculos se simplifican [HER03]. Ahora bien, si se desea tomar en cuenta la σ, entonces existe una variable que permite el cálculo adecuado mediante la ecuación (51).

Quizá queda la duda de saber por qué sólo estos tres algoritmos han sido abordados y pues el motivo es que son de los más importantes que existen en la literatura. Hasta ahora se ha hablado de definiciones y métodos pero a partir de estos tres trabajos, han sido derivados muchos más cuyo objetivo es el continuar la búsqueda de nuevas ideas para la reducción de ruido en señales. Donoho y Johnstone son considerados como los pioneros en la utilización de las wavelets como herramienta para la reducción de ruido [MAL99].

Hasta este momento ha sido expuesto cómo es que trabajan los diferentes algoritmos para reducir el ruido en señales usando la teoría de las wavelets. Y también se ha mencionado que son la plataforma para muchos otros algoritmos que pretenden reducir el ruido de forma más eficiente, como el que será presentado en el siguiente capítulo.