

# Implementación en Lenguaje de Alto Nivel de Tres Codificadores para Imágenes

Arturo Moreno, José Ismael de la Rosa, *Member, IEEE*, Jesús Villa, Ma. Araiza, y Ernesto García

**Resumen**—Este artículo tiene como objetivo presentar la implementación en lenguaje de alto nivel de tres técnicas de codificación de imágenes basadas en: 1) la transformada del coseno discreta (DCT, de sus siglas en inglés), que se implementa en la norma JPEG, 2) la transformada wavelet discreta (DWT, de sus siglas en inglés), en donde se utilizan varias bases como la de Haar y algunas biortogonales, entre otras, este codificador se implementa en la norma JPEG 2000, y 3) la transformada ridgelet discreta (DRT, de sus siglas en inglés) que forma parte de las herramientas más recientes de análisis multiresolución direccional para mejorar aún más la capacidad de reconstrucción de las imágenes desde el punto de vista cualitativo (mejor calidad visual). La interfaz implementada tiene un conjunto de funciones que conforman un ambiente gráfico abierto para la inclusión de más herramientas de procesamiento de imágenes con otras finalidades adicionales a la codificación, como lo son el filtrado y la restauración de imágenes utilizando algunas de las técnicas de transformación presentadas en este trabajo.<sup>1</sup>

**Temas Clave**— Codificación de imágenes, Transformada del coseno discreta, Transformada wavelet, Transformada ridgelet, Análisis multi-resolución y direccional.

## I. INTRODUCCIÓN

LA compresión de imágenes se define como el proceso de reducir la cantidad de datos necesarios para la representación de las mismas [17]. La compresión de datos nace en la etapa temprana de la computación, cuando era necesario el uso de métodos para la reducción del espacio necesario para almacenar o transmitir la información, ya fuera ésta en la forma de textos, programas, imágenes o cualquier otro tipo de información [31], [34]. La evolución en este sentido ha sido tal, que en este momento en el que los medios de almacenamiento se han vuelto económicos a comparación con los costos de almacenamiento en las primeras etapas de la computación, es necesario el aprovechamiento de todo el espacio disponible ya que ahora se desea tener sistemas que manejen video (imágenes en movimiento), por ello la mejora en los métodos de compresión es una línea de trabajo con un futuro muy prometedor desde el punto de vista tecnológico y de investigación [6], [7], sobre todo con la reciente introducción del análisis multi-resolución direccional [2], [3], [4], [8], [9], [10], [12], [13], [14], [18] en donde tenemos técnicas como la transformada ridgelet, que veremos implementada en este trabajo, la transformada curvelet, la

transformada contourlet [11], bandlets [19], brushlets [25], entre algunas de las hoy denominadas X-lets.

La norma más reciente establecida por la ISO/ITU-T en lo referente a la codificación y compresión de imágenes fijas es la JPEG 2000 (compresión con pérdidas y sin pérdidas). Dicha norma supera por mucho en calidad a la norma estándar durante los 90's JPEG [35]. Otros estándares recientemente actualizados son JPEG-LS, MPEG-4 VTC (Visual Texture Coding) y MPEG-4/7/21. Estas últimas utilizan el núcleo de la norma JPEG [39] (pero únicamente para el caso de la transmisión/recepción de video), es decir, la transformada discreta del coseno (DCT). En el trabajo de D. Santa-Cruz [35] se introducen de manera un poco más detallada los esquemas para JPEG y JPEG2000, en donde además se lleva a cabo una comparación con respecto a otros esquemas importantes, sobre todo en el caso de compresión sin pérdidas. El núcleo de la norma JPEG 2000 [37], [38], es la transformada wavelet discreta (DWT) [22], [23]. Esta norma utiliza además otras etapas como: la cuantización escalar, un modelado contextual, codificación aritmética y post-compresión.

Uno de los casos que había preocupado bastante hasta hace poco más de diez años era la inconsistencia de la DWT para el análisis de señales bidimensionales (2-D), pues la DWT no es capaz de mapear adecuadamente discontinuidades a lo largo de una línea o curva en un espacio bidimensional, como el caso de los contornos (cerrados o abiertos) presentes en una imagen. Con la finalidad de atacar este problema D. Donoho y E. Candès [2] de la universidad de Stanford se han dado a la tarea de proponer nuevas transformaciones que recurren al uso mismo de la teoría de wavelets. Estas nuevas herramientas de análisis son conocidas como ridgelets y curvelets. Las ridgelets tratan de manera más efectiva las singularidades que se presentan en dos dimensiones (2-D). La idea principal es mapear las líneas de singularidades en singularidades puntuales utilizando para ello la transformada de Radón [16], [26]. Enseguida, se aplica la transformada wavelet (unidimensional) la cual si puede manipular de forma eficaz las singularidades puntuales que se encuentran en el dominio de Radón. La propuesta inicial se plasma en forma continua para funciones definidas en  $L_2(R)$ . La versión discreta de la transformada ridgelet implementada en este trabajo, es similar a la transformada finita ridgelet introducida por M. Do [8], y el esquema del codificador ridgelet se inspiró en el esquema utilizado para la transformada wavelet 2-D.

La aportación principal de este trabajo, es la construcción de toda una serie de funciones programadas en lenguaje C y empleadas para el diseño de los tres codificadores de imágenes fijas [15], [20], [29], en donde se emplearon las

<sup>1</sup>Los autores trabajan en la Universidad Autónoma de Zacatecas - Unidad Académica de Ingeniería Eléctrica, Lab. de Procesamiento Digital de Señales, Av. López Velarde No. 801, Col. Centro, C.P. 98068, Zacatecas, Zacatecas, Mexico. e-mails: morenob20@hotmail.com, vargasj@uaz.edu.mx

transformadas del coseno, la wavelet y la ridgelet (en forma discreta) como se describe en la sección II. Después, se presenta la interfaz ANAI mostrando algunos resultados de los codificadores programados y comparados en la sección III, donde los resultados obtenidos son consistentes con los reportados en la literatura existente referente a los esquemas de codificación. En la sección IV brindamos algunos puntos de conclusión sobre la implementación realizada y posibles trabajos a futuro.

## II. CODIFICADORES BASADOS EN TRANSFORMACIÓN

En las dos últimas décadas se han propuesto diferentes metodologías para la codificación y compresión de imágenes. En este trabajo se rescatan dos esquemas que forman parte importante de las dos normas más recientes a nivel mundial para la compresión de imágenes fijas. También se presenta una metodología adicional que aún se encuentra en investigación y que se basa en el principio de mejorado de la calidad visual a partir de un modelado geométrico direccional. A continuación se presentan de manera breve las tres metodologías abordadas y comparadas en el presente trabajo, las cuales forman parte de las técnicas de codificación por transformación [30], [31].

### II-A. Codificador de imágenes DCT

El esquema de compresión que se obtiene utilizando la DCT se implementa mediante los siguientes pasos [1], [17], [39]:

1. Se divide la imagen en bloques (el tamaño recomendado es  $8 \times 8$ ).
2. Se resta  $2^7 = 128$  a cada uno de los valores de la imagen.
3. A cada bloque se le aplica la DCT en 2 dimensiones.
4. Se divide cada elemento del bloque con su correspondiente elemento de una matriz de cuantificación. La mejor calidad de la imagen reconstruida se ve reducida de acuerdo al nivel de cuantización. A menor nivel de cuantización mayor calidad y viceversa. El nivel de cuantización se controla multiplicando por una constante la matriz de cuantización.
5. Se comprime la información utilizando algún método de compresión clásico (Shannon- Fano, Huffman, o codificación Aritmética) [17].
6. Transmitir o guardar la información.

Para el caso de la decodificación, en donde se tiene que recuperar la imagen original, sólo es necesario aplicar los pasos en orden inverso:

- El primer paso es reconstruir la matriz cuantizada utilizando la lista obtenida al aplicar la DCT, así como cuantización y diezmado.
- Se multiplica por la matriz cuantizada.
- Se aplica la transformada inversa del coseno discreto.
- Se redondean los valores y se les suma a cada uno  $2^7$ .
- Finalmente, se obtiene la imagen reconstruida.

Al aumentar el nivel de compresión se tendrá una pérdida de calidad. Debido a esto, es importante cuantificar la cantidad de pérdida, a través de la medición del error, en donde se usará la relación señal a ruido pico (PSNR). También, es válido

mencionar que en los codificadores que se implementaron se dejó la posibilidad para análisis por bloques de  $16 \times 16$  y  $32 \times 32$ , lo cual nos permite ver lo que sucede cuando cambia solo este parámetro dentro de la construcción de los codificadores por transformación.

### II-B. Codificador de imágenes DWT

La transformada wavelet en dos dimensiones se obtiene utilizando la transformada wavelet en una dimensión [5], [24]. Los pasos para calcular la transformada wavelet se observan en la figura 1 y a continuación se explica el algoritmo [17], [23].

1. La imagen se divide en renglones.
2. Se calcula la transformada wavelet para cada renglón (Aproximación y detalles), obteniéndose dos imágenes de  $\frac{N}{2} \times N$ .
3. Los bloques de  $\frac{N}{2} \times N$ , resultado del paso anterior, se organizan en columnas.
4. Se calcula la transformada wavelet para cada columna (Aproximación y detalles), obteniéndose cuatro imágenes de  $\frac{N}{2} \times \frac{N}{2}$ .
5. Las imágenes obtenidas se organizan como se muestra en la figura 1.

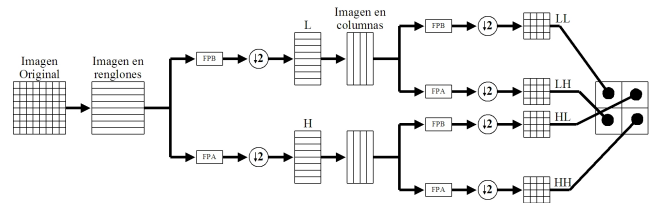


Figura 1. Transformada wavelet en dos dimensiones.

La transformada wavelet inversa en 2-D se calcula de manera similar; se hace uso de la transformada wavelet inversa en una dimensión. En la figura 2 se muestra el algoritmo utilizado. La descomposición en sub-bandas se lleva a cabo de izquierda a derecha. Luego, a la sub-banda LL se le aplica la transformada wavelet, con lo que se compacta aún más la energía en la banda de baja frecuencia. De igual manera, se obtiene el mismo resultado para una descomposición a tres niveles y así es posible seguir descomponiendo la imagen hasta el nivel máximo de descomposición que permita la imagen.

La transformada wavelet sólo compacta la energía. El diezmado y cuantizado realizan el trabajo de compactar la información (la información después de la compresión debe ser la misma o lo más semejante posible). Para ello se definirá a " $S_n$ " como la descomposición "n-ésima" dentro de la imagen y a " $S$ " como el número o nivel de descomposición y el número de descomposiciones " $S_{Max}$ " se calculará con la siguiente ecuación

$$S_{Max} = 3S + 1. \quad (1)$$

### II-C. Codificador de imágenes FRIT

Las transformadas ridgelet y curvelet son dos alternativas recientes utilizadas para las representaciones de sistemas multiescala. Se pueden emplear tanto para filtrado de señales,

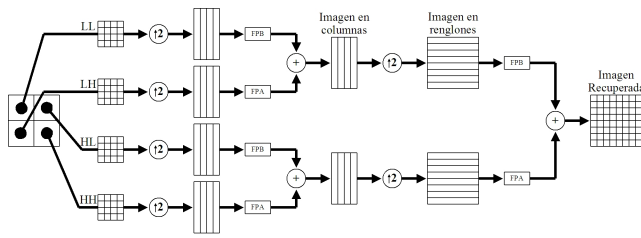


Figura 2. Transformada wavelet inversa en dos dimensiones.

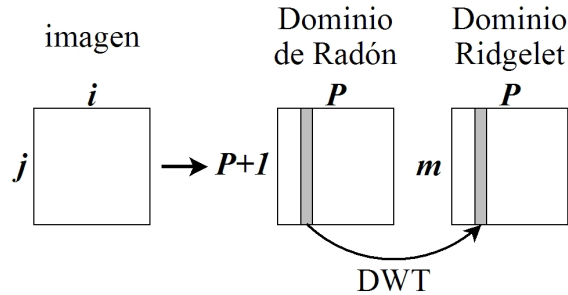


Figura 3. Transformada Ridgelet por medio de la transformada de Radón.

como para la compresión de éstas. La transformada ridgelet fué introducida en el trabajo de tesis de E. Candès en 1998 [2] bajo la supervisión del profesor D. Donoho. Dicha transformada fue introducida como una expansión de funciones esparcidas en espacios continuos que son lo suficientemente regulares (smoothness condition), lo cual permite alejarse de posibles discontinuidades [3], [4], [8], [9]. La transformada ridgelet discreta se obtiene para señales de tamaño finito y tiene un vínculo especial con la transformada de Radón finita (FRAT) [9]. La transformada ridgelet finita (FRIT por sus siglas en inglés) se obtendrá por medio del cálculo de la transformada discreta o finita de Radón (FRAT), a la cual se le aplicará la transformada wavelet discreta (DWT) en todas sus columnas. En la figura 3 se muestra el algoritmo, el cual queda descrito en los siguientes pasos:

1. Se restringe a la condición de que  $i = j$ .
2. Para que sea posible la aplicación de la transformada discreta de Radón se extiende la imagen al tamaño  $P \times P$ , donde  $P$  es un número primo.
3. Se aplica la transformada discreta de Radón a la imagen extendida.
4. Para la aplicación a las columnas de la transformada wavelet discreta se calcula el valor de  $m$  utilizando el algoritmo mostrado en la figura 4.
5. Se extienden las columnas al tamaño  $m$ .
6. A cada una de las columnas se les aplica la transformada wavelet discreta.

Para la codificación y cuantización, se aplicará el algoritmo de árboles de ceros (EZW) [36] a cada una de las columnas resultado de la transformada ridgelet, para lo cual se deberán re-organizar los coeficientes como se muestra en la figura 5. A cada columna re-organizada se le aplicará el algoritmo de árboles de ceros. Los coeficientes obtenidos se truncan de acuerdo al nivel de compresión deseado, con lo que se tiene el

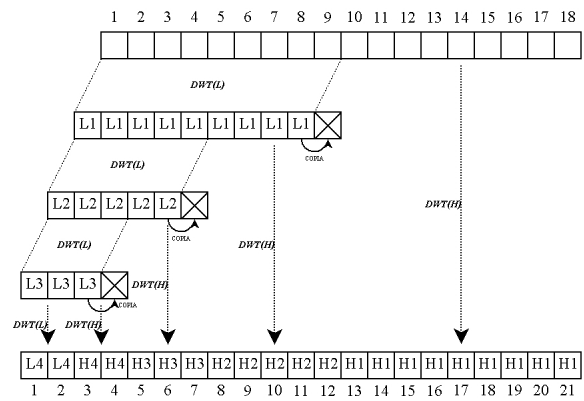


Figura 4. Transformada wavelet con número de muestras que no están en potencias de dos.

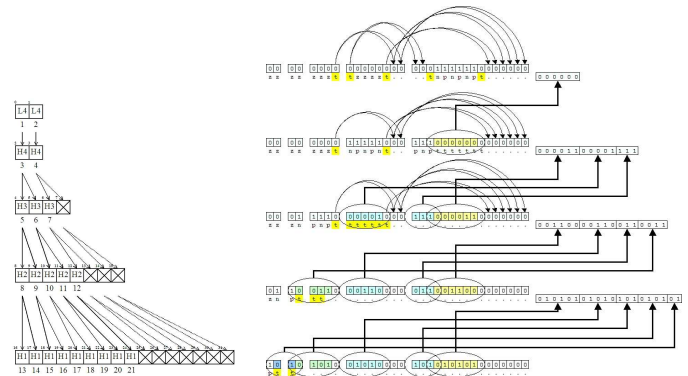


Figura 5. Re-organización de coeficientes y aplicación del algoritmo EZW.

esquema de compresión ridgelet completo (de manera similar que para la DWT).

### III. INTERFAZ IMPLEMENTADA Y COMPARATIVA DE ALGUNOS RESULTADOS

Las funciones de los codificadores se agruparon en una biblioteca de enlace dinámico (DLL) denominada ANAI.DLL, la cual se utiliza para probar los diferentes métodos programados para la construcción de los codificadores. Además, se construyó una interfaz gráfica programada en Visual Basic que hace uso de los codificadores presentados en este trabajo como se puede ver en las figuras 6, 7, 8, y 9. En la figura 6 se lleva a cabo la codificación por bloques, utilizando un bloque de  $32 \times 32$  y procesando la imagen de prueba Camera man con los tres codificadores. En el menú principal podemos desplazar cuatro sub-menús que contienen operaciones propias de cada uno de los codificadores, cada uno de los botones se utilizan para desplegar diferentes imágenes que van desde la imagen error, hasta la imagen reconstruida. Al igual que para la figura 6, en la figura 7 se presentan los resultados de codificación por bloques de la imagen Gold Hill, donde se puede apreciar el resultado de la reconstrucción del bloque perteneciente a la puerta de la segunda casa hacia abajo. Cabe mencionar, que también en los sub-menús podemos desplegar los PSNR de reconstrucción y seleccionar la tasa de compresión deseada.

Por otro lado, en el primer sub-menú tenemos la posibilidad



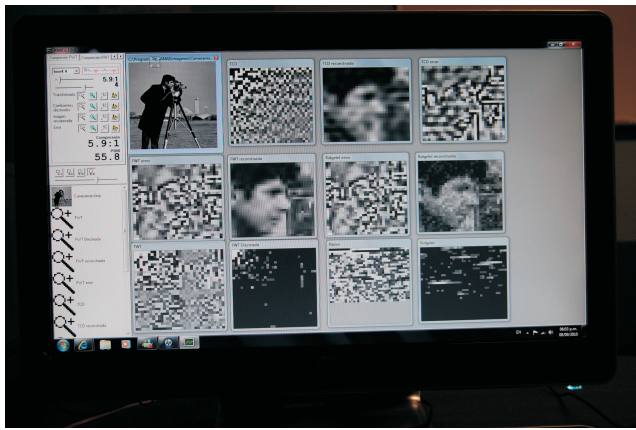


Figura 6. Interfaz ANAI procesando la imagen de prueba Camera man con los tres codificadores.



Figura 8. Interfaz ANAI procesando la imagen de prueba Lena con los tres codificadores.

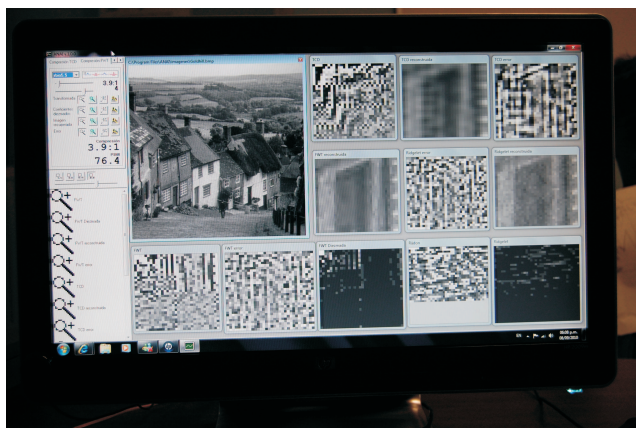


Figura 7. Interfaz ANAI procesando la imagen de prueba Gold Hill con los tres codificadores.

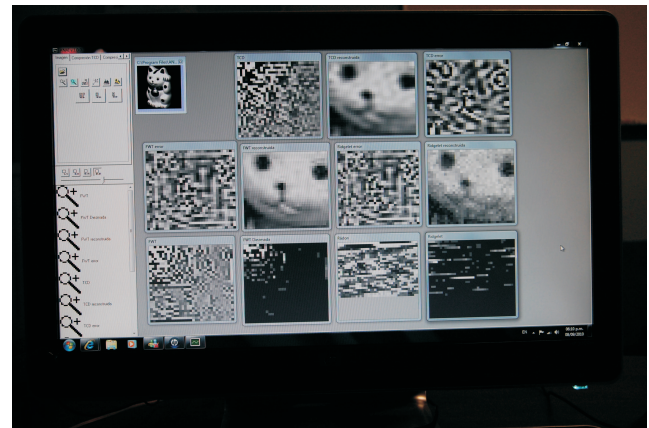


Figura 9. Interfaz ANAI procesando la imagen de prueba obj4\_5 (Gato) con los tres codificadores.

de cargar la imagen que vamos a procesar que debe de estar en formato BMP de 256 colores, también podemos desplegar histograma, tablas de valores de los codificadores Shannon-Fano (SF), Huffman (H), o codificación Aritmética (A). En la figura 8, podemos ver la codificación de una sub imagen de Lena, donde se muestra la reconstrucción del ojo izquierdo al centro, de arriba hacia abajo tenemos la reconstrucción con la DCT, luego la DWT, y finalmente la FRIT. Luego, al extremo derecho de la figura se muestran dos imágenes, arriba se tiene el ojo donde a cada pixel se le asocian los valores obtenidos de la codificación Huffman, y la imagen de abajo es la del ojo "original". Finalmente, en la figura 9 se presentan algunos resultados visuales (en este caso el bloque de  $32 \times 32$  corresponde a un segmento de la cara), que se detallan más adelante para toda la imagen completa obj4\_5 (que corresponde a la estatuilla de un gato) para el caso de los tres codificadores implementados y para cuatro diferentes tasas de compresión.

Las pruebas se realizaron con un total de 8 imágenes estándar como: baboon, camera man, barbara, lenaobj, obj1\_66, obj2\_32, obj4\_5, y 1.1.03 [27], [28]. Las imágenes fueron comprimidas a diferentes tasas para comparar






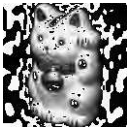
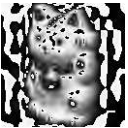
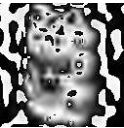


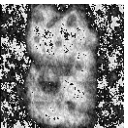
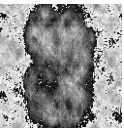
los algoritmos entre sí. Se realizó una comparación objetiva y subjetiva de cada una de las imágenes, presentando en cada prueba, primeramente la compresión basada en la transformada del coseno discreto (DCT), enseguida la transformada wavelet (DWT) y finalmente la compresión basada en transformada ridgelet (FRIT). La información en las pruebas de la Tabla I está descrita de la siguiente forma:

- Nivel de compresión logrado por la transformada del coseno discreto.
- PSNR de la imagen recuperada.
- Nivel de compresión logrado por la transformada wavelet.
- PSNR de la imagen recuperada.
- Nivel de compresión logrado por la transformada ridgelet.
- PSNR de la imagen recuperada.

La primera imagen mostrada de la transformada del coseno discreto corresponde a la compresión obtenida sin aplicar el diezmado de coeficientes a la imagen (primera columna a la izquierda, primera fila en la Tabla I). La segunda imagen corresponde al valor más cercano a una compresión de 20:1 con una cuantización entera. Asimismo, la tercera y cuarta imágenes corresponden a una cuantización con un nivel de

40:1 y 80:1 respectivamente. Para la compresión basada en la transformada wavelet, se muestran los niveles de compresión de 10:1, 20:1, 40:1 y 80:1, respectivamente. Finalmente la transformada ridgelet se probó con niveles de compresión de 1:1, 2:1, 5:1 y 10:1, obteniéndose resultados interesantes en su aplicación a cada una de las imágenes. A continuación, presentamos algunos resultados obtenidos para la imagen obj4\_\_5 (ver tabla I), el punto más interesante a resaltar es la imagen de la transformada ridgelet a una tasa de compresión de 1:1, en la cual se observa que el contorno del objeto se ha definido con bastante precisión, y se observa cómo la transformada wavelet tiene un rendimiento pobre cuando se encuentran dentro de la imagen demasiadas discontinuidades, aunque el PSNR se mantenga similar al de la DCT.

Tabla I  
PRUEBAS REALIZADAS A LA IMAGEN OBJ4\_\_5.

				
a)	3.2:1	21.8:1	41.2:1	81.3:1
b)	141.84	64.38	54.36	44.23
				
c)	10:1	20:1	40:1	80:1
d)	72.8	61.3	54.1	46.8
				
e)	1:1	2:1	5:1	10:1
f)	84.19	54.66	37.66	27.00

Los codificadores por transformación analizados en esta sección muestran ventajas y desventajas propias de las técnicas y métodos utilizados tanto para llevar a cabo la transformación (además se confirman algunas conclusiones establecidas en trabajos de J. Rolón [32] y en el trabajo reciente de J. Ma [21]), como las utilizadas para diezmar y cuantificar los coeficientes obtenidos mediante la transformación. Como se muestra a lo largo de las imágenes de prueba, el uso de un método u otro depende del tipo de imagen y el nivel de pérdida de información que sea permitido. En años recientes incluso, se ha demostrado que para el caso de la codificación de imágenes se pueden proponer técnicas eficientes como se muestra en [33], las cuales también se basan en el análisis wavelet.

#### IV. CONCLUSIONES

Existen imágenes que por su composición sacan ventaja de ciertos modelos de algoritmos más que de otros y es posible lograr una cantidad de compresión mayor. Es interesante recalcar que algunos fenómenos indeseables para la compresión pueden ser aprovechados en el campo de filtrado de señales, búsqueda de contornos, segmentación de imágenes, etc.

La contribución principal de este trabajo es la implantación de los algoritmos mencionados dentro de este documento, en

una biblioteca de enlace dinámico (DLL) llamada ANAI.DLL, la cual puede ser utilizada para probar los diferentes métodos desarrollados. Además, se construyó una interfaz gráfica (ver por ejemplo figura 6) que hace uso de los algoritmos introducidos en el presente trabajo, lo cual permite interactuar con los 3 esquemas de compresión utilizados para corroborar los resultados del comparativo llevado a cabo. Como trabajo a futuro se plantea implementar las transformadas ridgelet y curvelet utilizando la transformada de Radón con recorrido óptimo de coeficientes, sobre todo para su implementación en tareas de filtrado y restauración de imágenes, permitiendo la propuesta de nuevos algoritmos. La implementación de estos métodos también podría llevarse a cabo mediante el uso de DSPs y FPGAs. Otra tarea pendiente concierne a la implementación de otras transformadas direccionales tales como la transformada contourlet para el mejor modelado de discontinuidades.

#### AGRADECIMIENTOS

Nuestros más sinceros agradecimientos al Programa de Mejoramiento al Profesorado (PROMEP) de la Secretaría de Educación Pública (SEP) de México, por el apoyo brindado para la realización de este trabajo, bajo el registro UAZAC PTC 24-103.5/03/1127.

#### REFERENCIAS

- [1] Austin D., <http://www.ams.org/featurecolumn/archive/image-compression.html>, American Mathematical Society, Oct. 2008.
- [2] Candès E. J., *Ridgelets: Theory and applications*, Ph. D. Thesis, Department of Statistics, Stanford University, 1998.
- [3] Candès E. J., and Guo F., "New multiscale transforms, minimum total variation synthesis: Applications to edge-preserving image reconstruction", *Journal of Signal Processing*, Vol. 82, pp. 1519–1543, Jan. 2002.
- [4] Candès E. J., and Donoho D. L., "Curvelets and curvilinear integrals", *Journal of Approx. Theory*, Vol. 113, pp. 59–90, 2000.
- [5] Daubechies I., *Ten lectures on wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, Pennsylvania, 1992.
- [6] De la Rosa J. I., Villa J. J., Araiza Ma. A., Miramontes G., and García E., "El rol de nuevas transformaciones de señales n-dimensionales en futuras normas para imágenes fijas y para secuencias de video," *Primer Congreso en Tecnologías Computacionales y Sistemas de Información*, León, Gto., Septiembre 7 - 9, 2005 (7 páginas, Memorias en CD).
- [7] De la Rosa J. I., *Ridgelets y curvelets: Una nueva alternativa a la transformada en ondículas para la compresión y restauración de señales n-dimensionales*, Reporte Técnico, Laboratorio de Procesamiento Digital de Señales, Universidad Autónoma de Zacatecas, 2006.
- [8] Do M. N., and Vetterli M., "Orthonormal finite ridgelet transform for image compression," *IEEE International Conference on Image Processing (ICIP)*, Vol. 2, pp. 367–370, Vancouver, Canada, Sept. 2000.
- [9] Do. M. N., *Directional multiresolution image representations*, Ph. D. Thesis, Department of Communication Systems, Ecole Polytechnique Federale de Lausanne (EPFL), November 2001.
- [10] Do M. N., and Vetterli M., "The finite ridgelet transform for image representation," *IEEE Transactions on Image Processing*, Vol. 12, No. 1, pp. 16–28, Jan. 2003.
- [11] Do M. N., and Vetterli M., "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, Vol. 14, no. 12, pp. 2091–2106, Dec. 2005.

- [12] Donoho D. L., and Duncan M. R., "Digital curvelet transform: Strategy, implementation and experiments", *In Proc. Aerosense 2000, Wavelet Applications VII*, SPIE, Vol. 4056, pp. 12–29, 2000.
- [13] Donoho D. L., *Digital ridgelet transform based on true ridge functions*, Beyond Wavelets, 2001.
- [14] Fadili M. J., and Starck J.-L., *Curvelets and ridgelets*, Encyclopedia of Complexity and Systems Science, R. Meyers (Ed.), Vol. 3, pp. 1718–1738, Springer New York, 2009.
- [15] Flores C., *Desarrollo e implementación de un algoritmo de compresión de imágenes a altas tasas utilizando la transformada de ondículas*, Tesis de Maestría, Centro de Investigación y Desarrollo de Tecnología Digital-IPN, Enero 2000.
- [16] Ginkel M., Luengo C. L., and Van Vliet L. J., *A short introduction to the Radon and Hough transforms and how they relate to each other*, Quantitative Imaging Group Imaging Science and Technology Department.
- [17] González R. C., and Woods R. E., *Digital image processing*, Ed. Prentice Hall, Second edition, 2002.
- [18] Jaffard S., *Wavelets, tools for science and technology*, SIAM, 2001.
- [19] Le Pennec E., and Mallat S., "Sparse geometric image approximation with bandlets", *IEEE Transactions on Image Processing*, Vol. 14, no. 4, pp. 423–438, Apr. 2005.
- [20] Lugo E., *Diseño de bancos de filtros para la compresión de imágenes digitales*, Tesis de Maestría, Centro de Investigación y Desarrollo de Tecnología Digital-IPN, Septiembre 1999.
- [21] Ma J., and Plonka G., "The curvelet transform: A review of recent applications," *IEEE Signal Processing Magazine*, Vol. 27, No. 2, pp. 118–133, March 2010.
- [22] Mac A., *The fast wavelet transform*, Dr. Dobb's Journal, April 1992.
- [23] Mallat S., *A wavelet tour of signal processing*, Academic Press, Second Edition, 1999.
- [24] Meyer Y., *Wavelets, algorithms and applications*, SIAM, 1993.
- [25] Meyer F. G., and Coifman R. R., "Brushlets: A tool for directional image analysis and image compression", *Applied and Computational Harmonic Analysis*, Vol. 4, No. 2, pp. 147–187, Apr. 1997.
- [26] Milanfar P., "A model of the effect of image motion in the Radon transform domain," *IEEE Transactions on Image Processing*, Vol. 8, No. 9, pp. 1276–1281, Sept. 1999.
- [27] Nene S. A., Nayar S. K., and Murase H., *Columbia Object Image Library (COIL-20)*, Technical Report CUCS-005-96, Columbia University, February 1996.
- [28] Nene S. A., Nayar S. K., and Murase H., <http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php>, Columbia University Image Library, Oct. 2008.
- [29] Orozco F. L., *Algoritmos de compresión y descompresión de imágenes digitales para dispositivos móviles*, Tesis de Maestría, CINVESTAV, Febrero 2006.
- [30] Poularikas A. D., *The transforms and applications handbook*, IEEE Press, 2000.
- [31] Rao K. R., and Yip P. C., *The transform and data compression handbook*, Boca Raton, CRC Press LLC, 2001.
- [32] Rolón J. C., *A note on the curvelet transform and its application to image coding*, Reporte Técnico, Dept. of Signal Theory and Communications, Universidad Politécnica de Catalunya, 2006.
- [33] Rolón J. C., Ortega A., and Salembier P., *Modeling of contours in wavelet domain for generalized lifting image compression*, *In Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1025–1028, Taipei, Taiwan, 2009.
- [34] Salomon D., *Data Compression*, Springer Verlag, 2004.
- [35] Santa-Cruz D., and Ebrahimi T., "A study of JPEG 2000 still image coding versus other standards," *Proc. of the X European Signal Processing Conference*, Vol. 2, pp. 673–676, Tampere, Finland, Sept. 5 - 8, 2000.
- [36] Shapiro J. M., "Embedded image coding using zerotrees," *IEEE Transactions on Signal Processing*, Vol. 41, No. 12, pp. 3445–3461, Dec. 1993.
- [37] Skodras A., Christopoulos C., and Ebrahimi T., "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, Vol. 18, No. 5, pp. 36–58, Sept. 2001.
- [38] Usevitch B. E., "A tutorial on modern lossy wavelet image compression: Foundations of JPEG 2000," *IEEE Signal Processing Magazine*, Vol. 18, No. 5, pp. 22–35, Sept. 2001.
- [39] Wallace G. K., "The JPEG still picture compression standard," Submitted for publication in *IEEE Transactions on Consumer Electronics*, Dec. 1991.