



**UNIVERSIDAD
NACIONAL
AUTÓNOMA DE
NICARAGUA,
MANAGUA**

UNAN - MANAGUA

**FACULTAD REGIONAL MULTIDISCIPLINARIA DE CHONTALES
FAREM-CHONTALES
PROGRAMA DE DOCTORADO EN MATEMÁTICA APLICADA**

RESUMEN DE TESIS

**ALGORITMOS HEURÍSTICOS DE COINCIDENCIA PARA LA ESTIMACIÓN DE
MOVIMIENTOS EN COMPRESION DE IMÁGENES**

Autor: M.Sc.: Fernando José Hernández Gómez

Director de Tesis: M.Sc.-Ph.D.: Antonio Parajón Guevara

Consejera de Estudios: Ph.D.: Irene Loiseau

Diciembre, 2017

Índice

1. INTRODUCCIÓN	1
1.1. MOTIVACIÓN	3
1.2. ESTADO DEL ARTE	4
1.3. PLANTEAMIENTO DEL PROBLEMA	5
1.4. JUSTIFICACIÓN DEL PROBLEMA	7
1.5. ÁREA DE CONOCIMIENTO	9
1.6. Objetivos	9
1.6.1. Objetivo General	9
1.6.2. Objetivos Específicos	9
2. PROCESAMIENTO DE IMAGÉNES Y VÍDEO	10
2.1. Transformada Discreta del Coseno	10
3. ALGORITMOS DE ESTIMACIÓN DE MOVIMIENTOS	11
3.1. Métodos de Coincidencia por Bloques	11
3.2. Complejidad Computacional	12
4. METODOLOGÍA APLICADA	13
4.1. Modalidad de la Investigación	15
4.2. Herramienta computacional utilizada	15
5. IMPLEMENTACIÓN Y RESULTADOS	18
5.1. Aplicación de la DCT	18
5.2. Compresión alcanzada de la transformada discreta del coseno	19
5.3. Transmisión progresiva transformada discreta del coseno	19
5.4. Recuperación de Imágenes de Color en el Dominio de la Wavelet	20
5.4.1. Resultados de las muestras de las imágenes recuperadas	21
5.5. Algoritmos de Coincidencia para la Estimación de Movimientos	22
5.6. Secuencia de prueba para el conjunto de algoritmos de estimación de movimientos	22
5.6.1. Resultados de Implementación de los Algoritmos Heurísticos de Coincidencia para la Estimación de Movimientos	23

6. Conclusiones	24
7. Perspectiva de la Investigación	25

1. INTRODUCCIÓN

La Optimización Combinatoria es una rama de la teoría de Optimización en matemáticas aplicadas y en Ciencias de la Computación, que se encuentra estrechamente asociada a la Investigación de Operaciones o Programación Matemática, teoría de algoritmos y teoría de la complejidad computacional. Los algoritmos desarrollados resuelven instancias de problemas que suelen ser difíciles o en algunos casos imposibles de resolver; en general se explora un espacio de soluciones, relativamente grande, para ellas. Estos algoritmos lo logran reduciendo el tamaño del espacio, explorando y explotando al máximo las propiedades de forma eficiente y rápida. Con el desarrollo del estudio de la teoría de la complejidad computacional, es posible entender la importancia de los mismos.

Dentro del campo de la Investigación de Operaciones o programación matemática ha surgido una serie de procedimientos heurísticos y metaheurísticos para dar solución a estos problemas, los cuales son flexibles al involucrar características específicas y permiten la interacción con el decisor para buscar aproximaciones a la solución ideal, sobre la base del desarrollo tecnológico en el área de la computación, pues generalmente los procedimientos heurísticos son iterativos y requieren de una gran cantidad de cálculos.

El término de “Heurística” es un concepto que se originó en la Grecia clásica, de la palabra griega *heuriskein* que significa encontrar o descubrir; según la historia se deriva de *eureka*, famosa exclamación atribuida a Arquímedes.

De esta manera, se define heurístico como un procedimiento en que la solución no se determina en forma directa, sino mediante ensayos, pruebas y reensayos; obteniéndose un alto grado de confianza en las soluciones de calidad a un costo computacional razonable, aunque no se garantice la solución óptima o su factibilidad, en algunos casos, no se llega a establecer lo cerca que se encuentra de la solución óptima.

Siempre en la búsqueda de mejores soluciones sobre el espacio de soluciones y de buena calidad, la investigación en este campo en los últimos años, ha centrado su atención en el diseño de técnicas de propósito general para orientar la construcción de soluciones a partir de las diferentes heurísticas. Estas técnicas se denomina metaheurística y son estrategias para diseñar y/o mejorar los procedimientos heurísticos orientados a obtener un alto rendimiento en las soluciones de problemas que son difíciles de resolver.

En los últimos años, ha aparecido una serie de métodos bajo el nombre de metaheurísticas con el propósito de obtener mejores resultados que los alcanzados por los métodos heurísticos tradicionales. Estos surgen como un intento de producir una sinergia, es decir, que se pueden alcanzar mejores resultados haciendo uso de una combinación de métodos, proporcionando un marco general para crear o desarrollar nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos, entre otros.

El sufijo *meta* significa *más allá*, las metaheurísticas son estrategias para diseñar o mejorar los procedimientos heurísticos con miras a obtener un alto rendimiento. El término metaheurística fue introducido en [14] por (Glover, 2006), y a partir de entonces han aparecido muchas propuestas para el diseño de mejores procedimientos de solución a problemas combinatorios.

Mediante el estudio de la **teoría Complejidad Computacional** es posible comprender la importancia de estos tipos de problemas que en su mayoría son difíciles de resolver o imposible; pertenecen a la clase de problemas de optimización combinatoria $NP - Hard$, conforme crece el tamaño de los casos. Además, cabe señalar, que no existe una metodología única que los resuelva de forma eficientemente y exacta, o bien, si existe el tiempo de cómputo es muy elevado en casos a gran escala o en el caso que el espacio de búsqueda sea demasiado grande. Dichas instancias a menudo tienen ramificaciones teóricas - prácticas muy importantes, apareciendo en un gran número de aplicaciones en problemas que se presentan en la industria, logística, ciencias, ciencias médicas, ingenierías y en la administración, es decir en la vida real como en casos de estudios académicos, que se analiza desde el punto de vista teórico.

Es difícil realizar un análisis simple de un algoritmo que determine la cantidad exacta de tiempo que éste requiere para ser ejecutado, porque depende en gran parte de la eficiencia en la implementación del algoritmo y de los procesos de ejecución de la computadora. Además, conocer el tiempo exacto que tardará el cómputo o cálculo en generar los resultados, es una tarea difícil de encontrar, la principal tarea consiste en calcular la cantidad de operaciones que se realiza de acuerdo con los datos de entrada del problema a tratar, esto se conoce como “cálculo de la función temporal” y es utilizado desde el punto de vista teórico dentro de la teoría de complejidad computacional en [10] por (Cook,1971).

Una vez que se cuenta con un algoritmo que funciona de manera idónea, la siguiente tarea es definir los criterios que permitan medir su rendimiento o comportamiento, es decir, se ha de considerar el uso eficiente de los recursos y la simplicidad del algoritmo. Un algoritmo sencillo no le quita el crédito o desmerita la calidad, ya que su simplicidad facilita su mantenimiento, su verificación y su eficiencia.

Al hablar del uso eficiente de los recursos, éste puede medirse en función de dos indicadores: *el espacio*, cantidad de memoria que utiliza, y *el tiempo* que tarda en ejecutarse. Si para resolver un problema P , un algoritmo A , necesita de poca memoria de ejecución en un equipo con un pequeño número de instrucciones comparado con el resto de los algoritmos conocidos que resuelven el P , entonces se puede afirmar que A es más eficiente que los restantes, cuando se resuelve P .

Las principales líneas de estudio en el campo de la Visión Artificial son: generar descripciones inteligentes y útiles de: escenas, films, secuencias visuales sobre imágenes y vídeos, haciendo énfasis en características esenciales, objetos relevantes que aparecen en ellas. Algunos de los problemas que se estudia en esta área son:

- La detección de movimiento, cuyo objetivo principal consiste en detectar la estimación

movimiento en la escenas de una película, o pequeños films, compresión de imágenes digitales.

- Detección y localización de objetos en movimiento. Es uno de los problemas más complicados para la reconstrucción de escenas de películas, incluyendo también la detección y predicción de futuras trayectorias.
- Obtención de propiedades 3 – D de objetos a partir del movimiento. Es uno de los problemas clásicos de Visión Artificial.
- Un cuarto grupo de estudio que está en relación con el primero: Estimación de movimiento en problemas de compresión de imágenes y vídeo.
- Detección, reconocimiento, identificación, seguimiento de objetos, calibración de cámaras y visión estéreo, etc.

1.1. MOTIVACIÓN

Toda la idea detrás de la compresión de imágenes, secuencias o vídeo, consiste en la codificación de la Estimación de Movimiento con el frame ¹ actual con respecto al frame previo. Se crea entonces una imagen compensada por movimiento para el frame actual que está construida de bloques de la imagen desde la frame anterior. Se envían los vectores de movimiento para los bloques utilizados para la estimación de movimiento, así como la diferencia de la imagen compensada con el frame actual; también se codifica y envía *JPEG*. La imagen codificada que se envía, se decodifica en el codificador y se utiliza como marco de referencia para los frame siguientes. El decodificador invierte la proceso y crea un marco completo.

En general, para poder emplear los métodos, técnicas, modelos matemáticos y algoritmos de estimación de movimiento entre pares de imágenes, se establecen los siguientes supuestos sobre la secuencias de las imágenes adquiridas:

- Color/brillo del píxel constante, en el caso de imágenes a color.
- Los píxeles en una misma superficie se mueven de forma similar, coherentemente.
- Movimientos “suaves” y continuos;

La creciente complejidad de los sistemas de Visión Artificial hace que la posibilidad de probar fácilmente distintos algoritmos y métodos alternativos para llevarlos a cabo, suponga un ahorro considerable a la hora de diseñar e implementar dichos sistemas de visión artificial.

Algunas de sus aplicaciones principales destacan:

- Detección de movimiento aplicado a seguridad,
- Robótica y navegación Robótica,
- Reconocimiento óptico de caracteres,
- Construcción de modelos 3 – D ,

¹“es cada una de las imágenes instantáneas en las que se divide una película de cine que dan sensación de movimiento al ser proyectadas secuencialmente” (Konigsberg, 2004, p. 235).

- Aplicaciones médicas, biología, geología y meteorología,
- Seguridad automovilística,
- Captura de movimiento,
- Vigilancia en los parqueos, zonas residenciales, etc.
- Biométrica, control de calidad, etc.

El análisis de movimientos en la restauración y compresión de imágenes, secuencias de imágenes y vídeos es uno de los problemas principales en lo que respecta a la extracción de información, características y pistas relevantes para el análisis posterior de su contenido. La detección y segmentación de objetos se ven ampliamente beneficiadas si se cuenta con *algoritmos eficientes para estimación de movimientos*. Es por tal razón, que se tiene por objetivo obtener una buena correspondencia de puntos entre imágenes, secuencias de imágenes e imágenes consecutivas de un vídeo y así lograr la discriminación de los objetos según su movimiento.

1.2. ESTADO DEL ARTE

La operación más costosa desde el punto de vista computacional y con pocos recursos en todo el proceso de compresión es la recuperación de imágenes digitales y la Estimación de Movimiento de imágenes. Por tanto, este campo ha visto la mayor actividad e interés de investigación en las últimas dos décadas; en la que se diseña, implementa y evalúa los algoritmos de coincidencia de bloques fundamentales desde mediados de la década de 1980, los recientes algoritmos de comparación rápida de bloques del año 2002, hasta la actualidad.

Los métodos de estimación de movimiento más básicos se basan, en la diferencia entre imágenes consecutivas; la mayor parte de la totalidad de los trabajos sobre estimación de movimiento se centra sobre la hipótesis de restricción del flujo óptico propuesto en [20] por (Lucas, 1981) y en [15] se postula que la intensidad de un punto al que seguimos a lo largo de su movimiento permanece constante, desarrollado e implementado por (Herbst, 2013).

En [11] se realiza un estudio de la estimación de movimiento de coincidencia de bloques, el que desempeña un papel muy importante en la codificación de vídeo propuesto por (Cuevas, 2013). En este se plantea que, la estimación de movimiento se puede abordar como un problema de optimización, donde el objetivo es encontrar el mejor bloque de coincidencia dentro de un espacio de búsqueda. Además, se propone un nuevo algoritmo basado en la optimización de la Colonia de Abejas Artificiales (ABC) para reducir el número de ubicaciones de búsqueda en el proceso BM.

En este algoritmo, el cálculo del espacio de búsqueda se reduce drásticamente mediante la consideración de una estrategia de cálculo de aptitud que indica cuándo es factible calcular o sólo estimar nuevas ubicaciones de búsqueda. Dado que el algoritmo propuesto no considera ningún patrón de búsqueda fijo o cualquier otra suposición de movimiento como la mayoría de otros enfoques BM, existe una alta probabilidad de encontrar el mínimo real (Vector de mo-

vimiento preciso). Las simulaciones conducidas demuestran que el método propuesto alcanza el mejor equilibrio sobre otros algoritmos rápidos del BM, en términos de la exactitud de la estimación y del costo computacional.

En [30] se evaluó e implementa cuatro algoritmos de coincidencia de bloques utilizando la estimación de movimiento aplicados por (Yaakob, 2013). Además, se evalúan los efectos del tamaño del bloque empleado para encontrar el mejor algoritmo entre ellos y el más óptimo. Los resultados obtenidos sugieren que entre todos los algoritmos evaluados, *ARPS* tiene el mejor *PSNR* basado en el tiempo de cómputo.

En [12] se ofrece una clasificación de los métodos propuestos por (Dufaux y Moscheni, 2010) y los dividen en cuatro categorías: Las técnicas de gradientes, recursividad, comparación de bloques y las de dominio de frecuencias.

El seguimiento preciso del movimiento para un gran conjunto de puntos es una tarea que ha sido abordada en muchas ocasiones en [6] por (Brox, 2004 – 2014) y que aún es objeto de estudio para la comunidad científica.

También, a lo largo del estudio se revisaron y analizaron los algoritmos de coincidencia por bloques, la estimación de movimiento de búsqueda completa tratado en [7] por *Ratnam*, el algoritmo de búsqueda en cruz en [13] por (*Ghanbari*, 1990), el algoritmo de búsqueda de tres pasos en [18] por (Li, Zeng y Liou, 1994), el algoritmo de búsqueda de cuatro pasos en [?] por (*PoL.yMaW.*, 1996), el algoritmo de Búsqueda en Diamante en [9] por (*Cheung*, 2002), y la búsqueda de Patrón Adaptativo Rood en [21] por (Nie, 2002).

1.3. PLANTEAMIENTO DEL PROBLEMA

Una de las partes fundamentales del trabajo se centra en el campo de las Ciencias de la Matemática, Computación e Informática, ya que, ha sido tradicionalmente el diseño e implementación de algoritmos cada vez más eficiente para la solución de problemas, tanto de optimización como de búsqueda. La investigación de algoritmos exactos, heurísticos y metaheurísticos para resolver problemas de optimización combinatoria, puesto que nos enfrentamos a nuevos problemas de aplicación en las diferentes áreas del conocimiento, al mismo tiempo que disponemos de nuevos recursos tecnológicos y computacionales, tales como nuevos tipos de ordenadores con un poder de cálculo mucho mayor, redes y entornos virtuales, como el Internet, y clúster computadores con mejores procesadores.

La principal ventaja de la utilización de algoritmos exactos es que garantizan encontrar el óptimo global de cualquier problema, pero tienen el inconveniente que en problemas reales de complejidad computacional *NP* o *NP – Hard* su tiempo de ejecución crece de forma exponencial conforme crece el número instancia o espacio de soluciones analizados. Es por tal razón que se analizará los heurísticos que suelen ser bastante rápidos, pero la calidad de las soluciones encontradas suele ser bastante mala. Las metaheurísticas utilizadas ofrecen un

equilibrio entre ambos extremos; son métodos genéricos que ofrecen una buena solución y en muchos casos se obtiene óptimo global en un tiempo de cómputo razonable.

El desarrollo de Algoritmos de Estimación de Movimiento en Comprensión de Imágenes es un área de la visión artificial que ha tenido avances significativos en las últimas décadas. Dentro de sus aplicaciones pueden mencionarse: detección de movimiento aplicado a seguridad (reconocimiento de huellas dactilares, rostros, reconocimientos de iris y retina, vigilancia, etc.) de un banco o en las terminales autobuses, o de los aeropuertos, entre otros; el seguimiento visual, detección, reconocimiento, localización de objetos en movimiento, identificación de una serie de características locales (objetos, bloques, segmentos, etc.) en imágenes, incluyendo detección y predicción de trayectorias; obtención de propiedades en $2 - D$; la estimación de movimiento en problemas importante porque se determina la eficiencia de compresión de imágenes digitales, vídeo, control de calidad y la complejidad de un codificador de vídeo.

Los procesos visuales en el procesamiento de imágenes y vídeos son caracterizados utilizando modelos, técnicas heurísticas o metaheurística y estadísticos, algoritmos que describen el comportamiento de cada píxel a lo largo del tiempo. La teoría de NP-Complejidad sugiere que los algoritmos exactos y eficientes son poco probable que exista para la clase de problemas NP-difíciles. Una forma de hacer frente a la dureza NP es relajar el requisito de optimalidad y en su lugar buscar soluciones que estén probablemente cerca del óptimo. Esta finalidad principal de desarrollar los algoritmos de aproximación, que tienen por nombre heurísticos o metaheurísticos.

Con base a esto, estamos interesados en diseñar, implementar y proponer nuevas metodologías numéricas, modelos particulares, métodos específicos para resolverlo, algoritmos computacionales para su solución eficiente y aproximada mediante híbridos y heurísticos, para la más costosa de las tareas como: *estimación de movimiento* que corresponde a los algoritmos de búsqueda sobre el macrobloque de referencia, centradas en el análisis de los algoritmos de dominio de frecuencia y de dominio temporal.

El procedimiento consiste en encontrar un conjunto de esquemas permitidos, aquel que tenga un costo asociado y al reconstruir la imagen completa sea buena y de calidad visual. Cabe señalar que los errores que se presentan con frecuencia al reconstruir las imágenes son: *el efecto bloque y el efecto sombras*, debido a la cantidad de información analizada. La naturaleza de estos subproblemas hace que rara vez pueden ser tratados teóricamente debido a la gran cantidad de imágenes que se procesan y de las características estudiadas. Por lo tanto, nos vemos motivados a eludir el cálculo directo y considerar la utilización de ordenadores y algoritmos específicos que nos auxilien en nuestra tarea.

En el caso particular de una imagen de 512×16 píxeles, y con un valor de $p = 16$, el número total de operaciones por imagen es de $82,85 \times 10^8$. Con lo que para una secuencia con 30 *fps*, el número se incrementa a $98,55 \times 10^9$ operaciones, lo que resulta una cifra astronómica. Por tanto, si tomamos una imagen tamaño más grande, o un conjunto de imagen esto tiende hacer

impracticable.

Siendo ésta, una de las dificultades imprevistas, la cual radica en el tiempo que demora el ordenador en reconstruir la solución, puesto que muchos de estos problemas de optimización combinatoria pertenecen a la clase $NP - hard$ tratados en [4] por (Garey y Johnson, 1979). Es decir, el tiempo de cómputo y de construcción que se requiere se incrementa conforme crece el tamaño de la imagen, el número de imágenes analizadas o de las características en estudio.

Después de evaluar varias áreas de estudio, con la ayuda de mi director de tesis y mi consejera de estudio se eligió el tema: ***Algoritmos Heurísticos de Coincidencia Para la Estimación de Movimiento en Compresión de Imágenes***. Los algoritmos a tratar son los siguientes: Los algoritmos de dominio de frecuencia que se aplican sobre los coeficientes de las transformadas: **DCT, Wavelets**) y los algoritmos de dominio temporal (**block matching o de coincidencias**) tales como: el algoritmo de búsqueda exhaustiva, algoritmo de búsqueda en tres pasos, algoritmo de búsqueda en cuatros pasos, algoritmo de búsqueda en diamante, algoritmo de mejora del heurístico de búsqueda en tres pasos simple y eficiente, heurístico mejorado en la búsqueda de patrón en tres pasos.

La importancia de elegir el tema se debe a que es original y de alcance a nuestra realidad, pues aborda problemas que aportan resultados prácticos para la solución de los mismos. De igual manera, el enfoque del tema corresponde a la disciplina de los estudios de doctorado en las universidades a nivel nacional e internacional, refleja interés social y tecnológico, puesto que gira alrededor de una de las problemáticas del mundo moderno, que es la optimización de los recursos de compresión, transmisión y almacenamiento de información visual.

La finalidad de esta tesis se centra en el diseño e implementación de algoritmos de coincidencia basados en la Estimación de Movimiento, que requieren de las mejores implementaciones de algoritmos de dominio frecuencias basados en las transformadas de los coeficientes a través de *DFT*, *DCT* y *Wavelets*, así como los algoritmos de coincidencia que pertenecen al dominio temporal (FS, TTS, 4SS, DS, MHTSS, MHCS), que ha sido escasamente abordada en Nicaragua desde el punto de vista teórico y computacional.

1.4. JUSTIFICACIÓN DEL PROBLEMA

El potencial de la investigación radica en intentar resolver este problema que es de **carácter teórico - aplicado**, ya que al diseñar, implementar y evaluar los algoritmos de coincidencia para la estimación de movimientos con el fin obtener una mejora mínima en la solución, dar otras alternativas; en las que se puede llegar a obtener buena **eficiencia algorítmica**, siendo de **relevancia social** ya que estas mejoras generan desde el punto económico grandes ganancias a las industrias cinematográficas, sistema de seguridad informáticas o instituciones afines.

Otra de las razones que nos motiva el estudio, es su **utilidad metodológica**, puesto que puede servir de bases para posteriores investigaciones en Nicaragua, ya que aún no se han realizado,

es un tema poco investigado y de interés para la comunidad científica, servirá de soporte para ciertos cursos que se podrían implementarse en las diferentes universidades del país, por lo que es un campo nuevo o nulo dentro del ámbito de la Educación en Nicaragua, las distintas universidades de nuestro país no ofertan ni un curso o conferencias relacionada con el tema en estudio, en que la nueva generación haga uso y que sea de utilidad para la sociedad.

También, la importancia e interés del estudio radica en que se dejará un soporte del trabajo realizado tanto de la parte teórica como de la parte experimental, así como sus implementaciones en el lenguaje de programación *MATLAB* que le pueda ser de utilidad o de apoyo a los diferentes estudiantes de las carreras de Ciencias de la Computación, Sistema, Informática, Matemática, y otras carreras afines, que se encuentren pensando en trabajar en esta área y que sea asequible para ellos y al público en general.

También, tiene *Implicaciones prácticas*, debido a que sus resultados permite resolver un conjunto de problemas del área de investigación de operaciones, visión artificial, diversas área del conocimiento y de la vida cotidiana. Por ejemplo, los resultados obtenidos se aplican a problemas de la sociedad (*como la mejora de una fotografía antigua, o reconstrucción algunas que se encuentre en mal estado, es decir agregar ciertos coeficientes para mejorar su calidad visual, en el caso de salas de cine proyectar una imagen o película con mayor amplitud nos generan ciertos efectos de mala calidad, o en caso de una vídeo llamada, etc.*). Además, es aplicable en el análisis de imágenes médicas, ya que, puede convertirse en una herramienta de apoyo para el diagnóstico en las que necesitamos mayor visión y precisión. De la misma manera, nos puede servir de soporte en los diferentes campos en los que se necesita resguardar evidencias tales como: *la genética, la botánica, la zoología, la patología, la medicina forense, el diagnóstico por imágenes, así como instituciones de la salud, vigilancia y el ejército*. También, es aplicable desde el punto de vista de la mercadotecnia, la publicidad, los cuales desean vender y ofertar sus productos a los clientes de una forma visual y eficaz, etc.

De igual manera, tiene una **una viabilidad o factibilidad de ejecución**, siendo una de las razones principales que motivan el estudio, puesto que el único equipamiento necesario para llevar a cabo la investigación es una computadora personal con buen poder de cálculo, acceso a internet, y bibliotecas, bibliotecas virtuales facilitadas por docentes investigadores de otras universidades nacionales e internacionales, para la revisión de la literatura y la redacción del mismo, y de esta manera contribuir en desarrollo de investigaciones científicas de las diferentes universidades del país, siendo muy escasas.

1.5. ÁREA DE CONOCIMIENTO

De acuerdo con el reglamento del Sistema de Estudios de Posgrado y Educación Continua de La Universidad Nacional Autónoma de Nicaragua, *UNAN – MANAGUA* (2011), el área de conocimiento en el que versa el presente trabajo es el **Ciencias Aplicadas**, centrada en la **línea de Investigación Modelación y Programación Matemática**, puesto que, obedece a la naturaleza del estudio teórico de las propiedades de los modelos, métodos, técnicas matemáticas, algoritmización y herramientas generales de Optimización Combinatoria.

1.6. Objetivos

1.6.1. Objetivo General

Diseñar e Implementar Algoritmos Heurísticos de Coincidencia para la Estimación de Movimiento en compresión de imágenes.

1.6.2. Objetivos Específicos

1. Revisar el estado del arte de los diferentes Algoritmos de Coincidencia para la Estimación de Movimiento en compresión de imágenes.
2. Analizar la teoría matemática necesaria para el diseño e implementación de los Algoritmos de Coincidencia para la Estimación de Movimiento en compresión de imágenes.
3. Diseñar Algoritmos basados en el dominio de frecuencia (DCT, Wavelets) y Heurísticos (FS, TTS, 4SS, DS, MHTSS, MHCS) de Coincidencia para la Estimación de Movimiento en compresión de imágenes.
4. Implementar los Algoritmos del dominio de frecuencia (DCT, Wavelets) y Heurísticos (FS, TTS, 4SS, DS, MHTSS, MHCS) de Coincidencia para la Estimación de Movimiento en compresión de imágenes, escritos en el lenguaje de Programación Matemático *MATLAB*.
5. Comparar los resultados obtenidos mediante la experimentación computacional de los Algoritmos de Coincidencia para la Estimación de Movimiento en compresión de imágenes.

2. PROCESAMIENTO DE IMÁGENES Y VÍDEO

En este capítulo se expone las ideas relevantes del procesamiento de imágenes y vídeo. Se realiza una breve reseña histórica para comprender la génesis del procesamiento de imágenes, se mencionan algunos autores principales que brindaron aportes esenciales al desarrollo, se abordaron los conceptos básicos que se tendrán en cuenta a lo largo esta investigación tales como: la compresión y la transmisión de imágenes digitales, los tipos de redundancia existentes, las diferentes técnicas de compresión, la codificación de transformación por bloques, los procesos de transformación y cuantización de las imágenes, la representación matemática y sus propiedades, las diferentes medidas o métricas existentes para el proceso de compresión de imágenes digitales y vídeos. Se describen los métodos de determinación de flujo óptico, se hace mención de la transformada de Walsh-Hadamard, y los aspectos generales de la transformada de Fourier y principales aspectos de la transformada discreta del Coseno, la Wavelets, así como los detalles al implementar este tipo de métodos.

También, se muestra las métricas para la estimación de movimientos basados en los algoritmos de **block-matching**, en la que se determina el vector de movimiento (mv_x, mv_y) óptimo a partir de una *función de costos*. Esta función nos indica el grado de coincidencia entre la señal o frame original I_C y la señal reconstruida o el frame residual I_R , los parámetros m, n representan el largo y ancho de la imagen. Las diferencias entre las funciones de costos varían en términos de la complejidad y eficiencia de las métricas a utilizar, entre ellas están:

2.1. Transformada Discreta del Coseno

Se trabajo en la aplicación de las transformadas al procesamiento digital de imágenes ha crecido considerablemente en las últimas décadas. La transformada discreta del coseno (DCT: Discrete Cosine Transform), es la más empleada en las aplicaciones de compresión de imágenes y vídeo.

Se han desarrollado algoritmos rápidos o eficiente de la transformada discreta del coseno, propuestos por (Chen and Smith, 1997) en [8] en la que proporciona un factor de mejora en su complejidad computacional, cuando se compara con los algoritmos tradicionales de Transformada Discreta de Coseno.

Se analizaron las propiedades únicas que posee la DCT permiten obtener altos índices de compresión a muy bajo costo, puesto que tienen una buena propiedad de compactación de energía. La decorrelación de los coeficientes es una de las tareas fundamentales para compresión, ya que, el posterior tratamiento de cada coeficiente se puede realizar de forma independiente, sin pérdida de eficiencia al cuantificar los coeficientes que se eligen de forma visual.

De forma general, se estudiaron dos grupos de técnicas de compresión: sin pérdidas y con pérdida. La codificación sin pérdidas, la descompresión tiene como objetivo recuperar exactamente la información original de la imagen de tal forma que el porcentaje de compresión sea

buena, mientras que la compresión con pérdidas el proceso de compresión permite una pérdida de información que no se recupera en la descompresión.

3. ALGORITMOS DE ESTIMACIÓN DE MOVIMIENTOS

La **estimación de movimientos** es la parte principal del proceso de reconstrucción de imágenes digitales y codificación de vídeo. La **implementación de algoritmos eficientes** nos permite **reducir la redundancia temporal** en imágenes, secuencias de imágenes o vídeo, representa el $\frac{2}{3}$ del **costo de complejidad computacional** del proceso de reconstrucción y codificación de vídeo. Sin embargo, el problema tiende hacerse más complejo a medida que aumentan el número de áreas o regiones que se analizan y/o se mueven, aumenta el tamaño de las imágenes. Por otra lado, la codificación de segmentos de bordes, aumenta la complejidad computacional y el tamaño del código a implementar. Además, se aplica sobre diferentes soportes de modelo de movimiento, siendo utilizados: imagen completa, píxel a píxel, por bloques o por regiones de interés.

En la Estimación de movimiento por bloques la imagen se divide en bloques rectangulares disjuntos entre sí, de tamaño fijo. Se considera que sólo se producen traslaciones, siendo el modelo por excelencia utilizado con frecuencia en el proceso de compresión, principalmente por su simplicidad y sus buenos resultados de implementación. Uno de los **objetivos consiste en la búsqueda de la máxima coincidencia del movimiento real de los objetos, pero la tarea más fuerte es la reducción del error de predicción y la calidad visual de las imágenes**. Sin embargo, este presenta diferentes problemas ante movimientos más generales como una sencilla transformación afín que incluya escalado y/o rotación. Los modelos de movimiento se pueden agrupar en dos grandes clases: los modelos paramétricos y no paramétricos.

3.1. Métodos de Coincidencia por Bloques

Los métodos de coincidencia por bloques, cuyo término se deriva de inglés “Block matching” comprende toda una categoría de algoritmos que se comportan de forma similar dividiendo la imagen en regiones pequeñas denominados bloques, con frecuencia en cuadrados, rombos, entre otros. Estos intentan dividir las imágenes previa y actual en bloques.

Los algoritmos de dominio temporal se centran en la búsqueda de la máxima coincidencia del área de búsqueda, dentro de éstos forman parte los **algoritmos de coincidencia** que son objetos principales de nuestros estudio y los algoritmos basados en métodos de los gradientes. Estos centran su búsqueda en las coincidencias de la información (datos, características, objetos, etc.) del bloque actual con la del frame, bloque o marco, de referencia.

La idea más simple consiste en dividir una imagen en pequeños bloques rectangulares o cuadrangulares, de tamaño fijo, bajo el supuesto de que cada uno realiza una traslación independiente. Si estos bloques son suficientemente pequeños, se pueden aproximar movimientos

como la rotación de objetos grandes, o el zoom de la imagen, a partir de traslaciones de estas áreas o regiones, este método evita la codificación relativa a la segmentación y sólo es necesario codificar el vector de desplazamiento de cada bloque. La hipótesis sobre la que se sustentan estos métodos es que un bloque de tamaño $L \times L$ en la imagen $k - 1$ centrado en el punto (x_0, y_0) coincide con un bloque del mismo tamaño en la imagen k . Esto expresado matemáticamente quedaría

$$f(x, y, k) = f(x + d_1, y + d_2, k - 1) \quad \forall (x, y) \in B$$

Dicho de otro modo, cada uno de los bloques que componen cada una de las imágenes o frames de la secuencia de imágenes que posee un homólogo, decir, un bloque del mismo tamaño que representa la misma porción de la escena en otra imagen o frame consecutivo de la secuencia de imágenes.

Los algoritmos diseñados **block matching** (FS, TTS, 4SS, DS, MHTSS, MHCS), centran su búsqueda en la máxima coincidencia entre todos o algunos de píxeles en el bloque actual con el bloque de referencia sobre el área de búsqueda, basados en alguna métrica o función de costos que han sido abordada a profundidad en el capítulo anterior.

La mayor parte del estudio se centra tanto en las técnicas de estimación y compensación de movimientos como en el análisis de la segmentación de las imágenes para lograr un mayor nivel de estimación y compensación, para esto se estudia la segmentación por bloques fijos, por bloques variables y la segmentación basada en regiones tomando los objetos dentro de la imagen.

3.2. Complejidad Computacional

En el caso particular, para una imagen de 512×16 píxeles, y con un valor de $p = 16$, el número total de operaciones por imagen es de $82,85 \times 10^8$. Con lo que para una secuencia con 30 fps , el número se incrementa a $98,55 \times 10^9$ operaciones, lo que resulta una cifra astronómica. Por otro lado, si la fuente de información es de tamaño $N = 100$, y supongamos que una computadora realiza 10^{10} , es decir 10 giga de operaciones por segundo, encontrar una solución nos llevaría aproximadamente 4×10^{12} años para completarse sobre el espacio de búsqueda, mucho más tiempo que la edad actual del universo.

De esta manera, se observa cómo el método exhaustivo requiere un costo computacional muy alto, que puede impedir que sea utilizado en aplicaciones en tiempo real. Es por tal razón, que se diseñara y analizará los heurísticos que suelen ser bastante rápidos (eficientes) y de la calidad de las soluciones encontradas suele ser bastante buenas.

La teoría de la complejidad computacional plantea que un problema de decisión se puede clasificar en: la clase L, NL, P, NP, NP-Completos y NP-difícil, Co-NP. Sin embargo, para esta distinción es necesario considerar el modelo teórico de las máquinas de Turing (MT):

En la siguiente figura se resume lo anterior

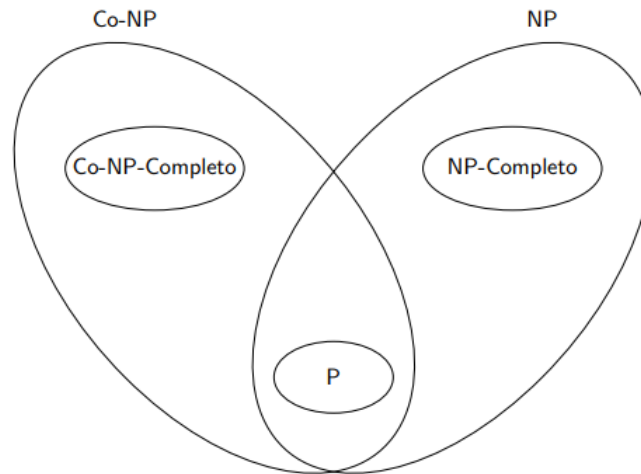


Figura 1: Complejidad teórica computacional

de lo anterior surgen las siguientes interrogantes:

$$P = NP, Co - NP = NP, P = Co - NP \cap NP,$$

que en la actualidad sigue siendo objeto de estudio para la comunidad científica.

4. METODOLOGÍA APLICADA

Para alcanzar los propósitos de la tesis se usó instancias o bases de datos de imágenes disponible en internet y que son de interés para la comunidad científica para la comprensión de secuencias de imágenes y tratamiento de vídeos en general, tratando de encontrar algunas soluciones exactas, y/o aproximadas utilizando heurísticos y sus mejoras para las soluciones presentadas por otros investigadores, para obtener primero la madurez necesaria del problema y de los subproblemas y elaborar, diseñar e implementar los algoritmos.

Se procedió a diseñar los diferentes algoritmos, los de dominio de frecuencia que se basan sobre los coeficientes de las transformadas: **la transformada Discreta del Coseno y la Wavelets**, se eligieron estos ya que nos sirven de base y son los que mejor se adaptan al tratamiento de los **algoritmos de dominio temporal** y que dan respuesta al problema en estudios, diseñándose también los siguientes heurísticos: el algoritmo de búsqueda exhaustiva, algoritmo de búsqueda en tres pasos, algoritmo de búsqueda en cuatro pasos, algoritmo de búsqueda en diamante, algoritmo y los heurísticos mejorados: búsqueda en tres pasos simple y eficiente, algoritmo de búsqueda de patrones en tres pasos.

El diseño metodológico de este trabajo se aprecia en la siguiente figura:

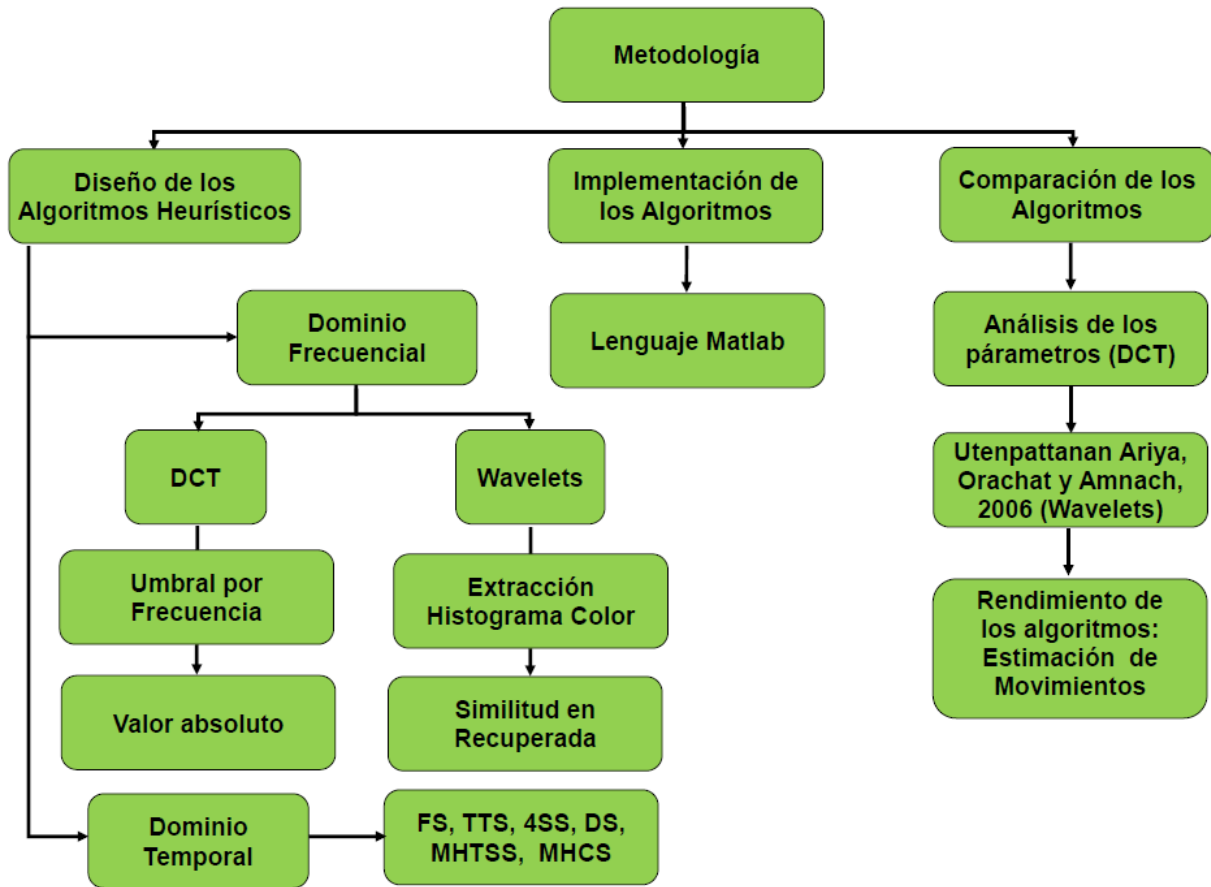


Figura 2: Esquema de la Metodología utilizada

Los métodos empleados en la investigación son: **Los métodos empíricos generales y puro**, puesto que la idea es partir de algoritmos de búsqueda completa, heurísticos ya conocidos por la sociedad científica, aplicarlos a distintas instancias del problema de compresión de imágenes estáticas, dinámicas y la estimación de movimientos. Además, se varía los parámetros de los modelos en las técnica de los heurísticos, realizándose híbridos de estos.

También, se utilizó **el método de análisis síntesis** puesto que particionó un problema complejo en otros más pequeños, después combinar los resultado de algoritmos obtenidos con el fin de equiparar o mejorar los resultados obtenidos, además abstracción y concreción puesto que hay que demostrar y comprobar cada uno de los resultados algorítmicos sean válidos. Debe de señalarse y dejar bien claro que en está investigación el más utilizado es el **método algorítmico**, puesto que se ha de implementar algoritmos heurísticos de búsqueda en el lenguaje de programación de alto nivel y trabajar con diferentes experimentos de laboratorio computacional para aprovechar las ventajas de los mismos.

Los métodos empleados en la investigación son: **Los métodos empíricos generales y puro**, puesto que la idea es partir de algoritmos de búsqueda completa, heurísticos ya conocidos por la sociedad científica, aplicarlos a distintas instancias del problema de compresión de imágenes

estáticas, dinámicas y la estimación de movimientos. Además, se varía los parámetros de los modelos en las técnica de los heurísticos, realizándose híbridos de estos.

También, se utilizó **el método de análisis síntesis** puesto que particionó un problema complejo en otros más pequeños, después combinar los resultado de algoritmos obtenidos con el fin de equiparar o mejorar los resultados obtenidos, además abstracción y concreción puesto que hay que demostrar y comprobar cada uno de los resultados algorítmicos sean válidos. Debe de señalarse y dejar bien claro que en está investigación el más utilizado es el **método algorítmico**, puesto que se ha de implementar algoritmos heurísticos de búsqueda en el lenguaje de programación de alto nivel y trabajar con diferentes experimentos de laboratorio computacional para aprovechar las ventajas de los mismos.

4.1. Modalidad de la Investigación

Debido a la forma de la manipulación el objeto de análisis de esta investigación, se clasifica como **una investigación aplicada exploratoria**. Puesto que, se justifica por la búsqueda e implementación de técnica de compresión de imágenes, aplicación de heurísticas en la búsqueda de la información y de los algoritmos de coincidencia de bloque para la estimación de movimientos, que ya se ha mencionado y con una finalidad hacia un problema práctico y relativamente soluble en el espacio de soluciones. Por tal razón, dejamos a un lado, la creación de teoría de nuevos algoritmos y tratar obtener las soluciones optimas, es decir la solución exacta del problema en estudio. Sin embargo, el hecho de construir un algoritmo sencillo o bien una implementación de las misma nos garantiza un valiosos aportes al conocimiento teórico - aplicado, y sobre todo en Nicaragua que es un área inexplorada y con mucho camino que recorrer.

4.2. Herramienta computacional utilizada

Para las diferentes ejecuciones de las técnicas y algoritmos implementados se necesitó del uso adecuado de la herramienta *MATLAB* versión *R2015b* y su biblioteca de visión artificial para la implementación de los algoritmos elegidos para el procesamiento de imágenes o secuencias, la estimación de movimientos aplicando las transformada discreta del coseno y wavelets, y los heurísticos.

A continuación, mostramos los códigos realizados en Matlab de las principales funciones, procedimientos utilizadas en el proceso de la transformada discreta del coseno, entre ellas tenemos:

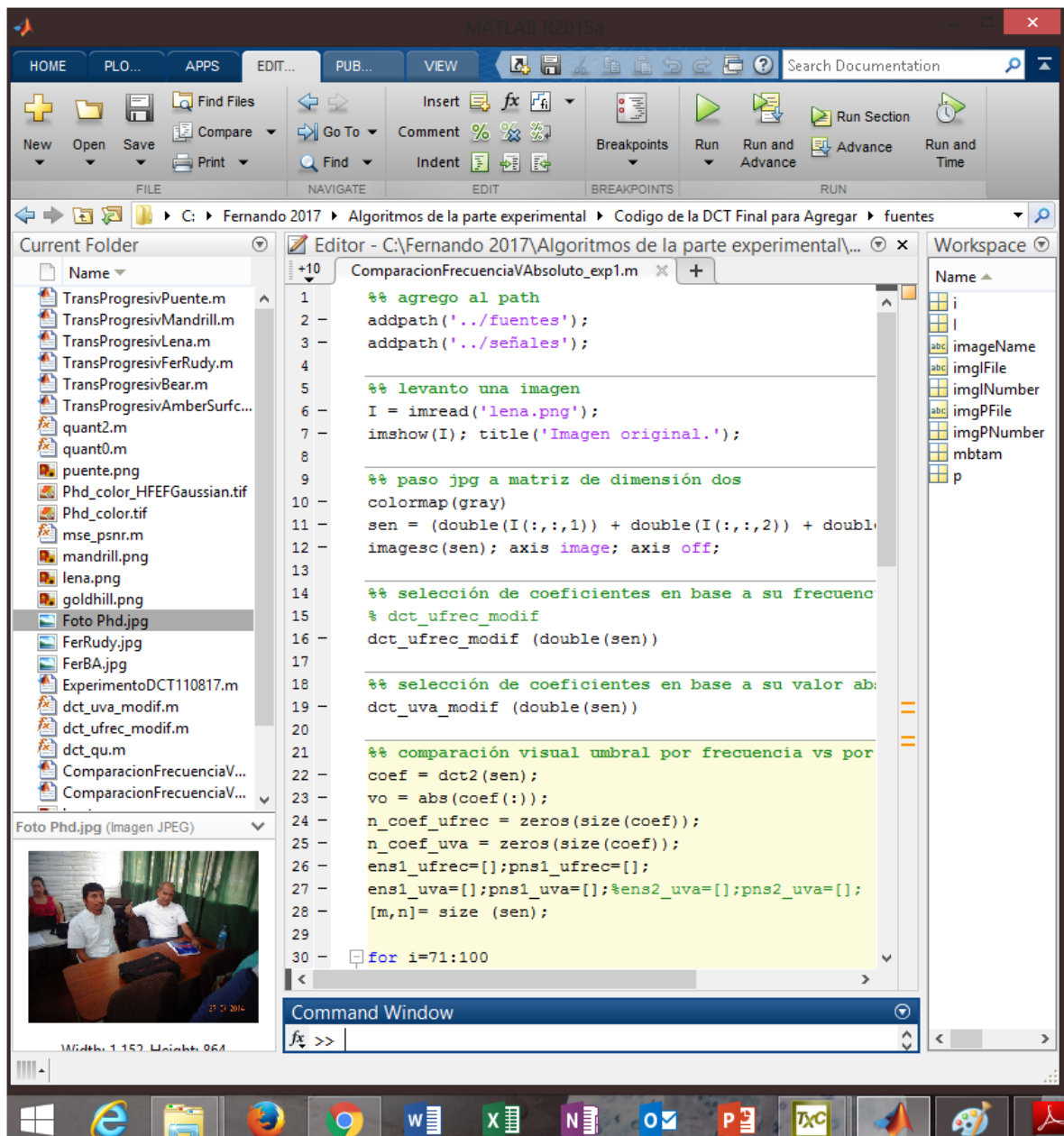


Figura 3: Comparación de la DCT

A continuación, se muestra los códigos de las principales funciones, procedimientos para la implementación de la Wavelets:

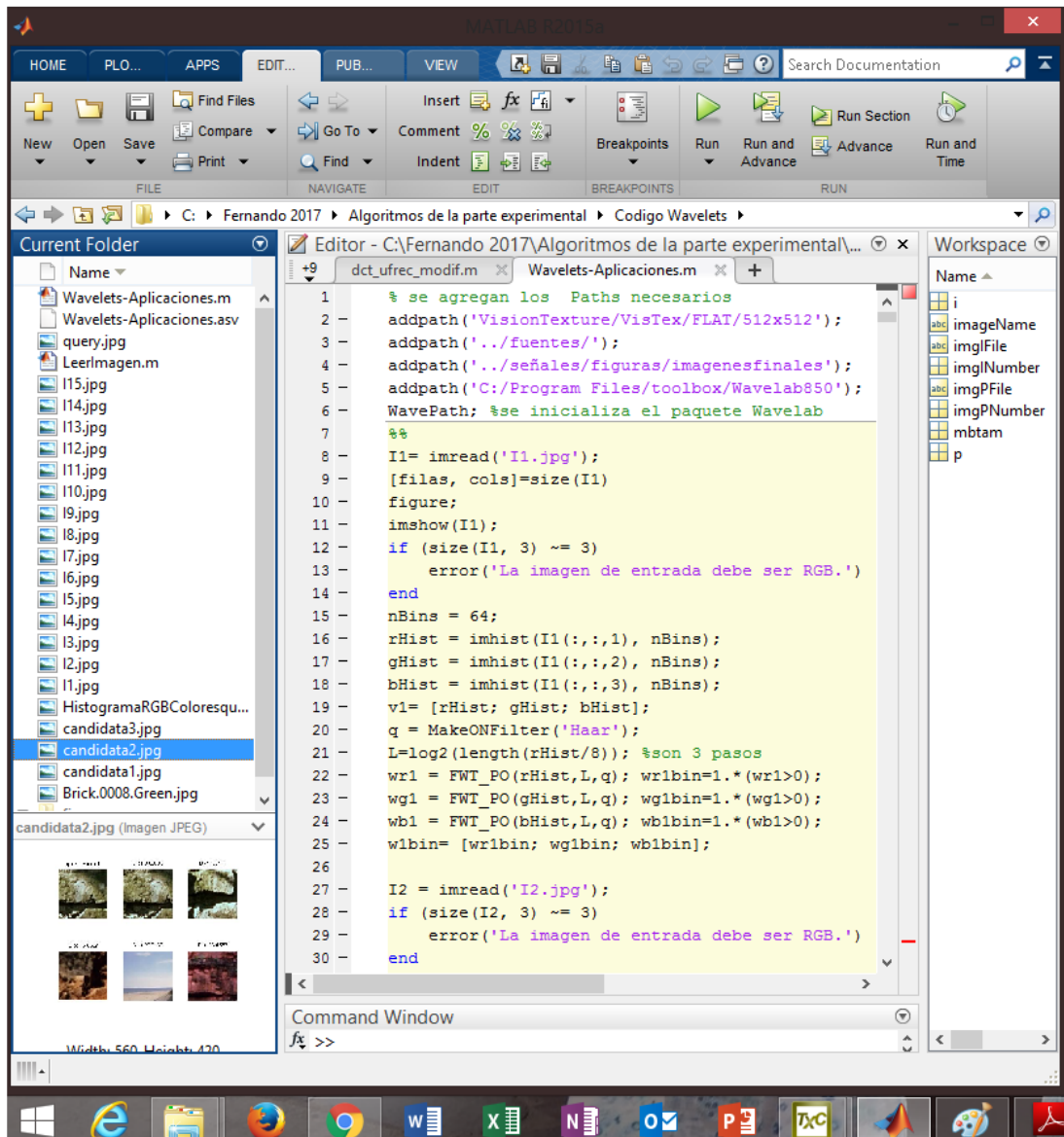


Figura 4: Aplicación de los coeficientes de la transformada Wavelets

A continuación, se detalla las principales funciones utilizadas en el proceso de implementación de los algoritmos heurísticos de coincidencia para la estimación de movimientos en comprensión de imágenes, entre ellas tenemos:

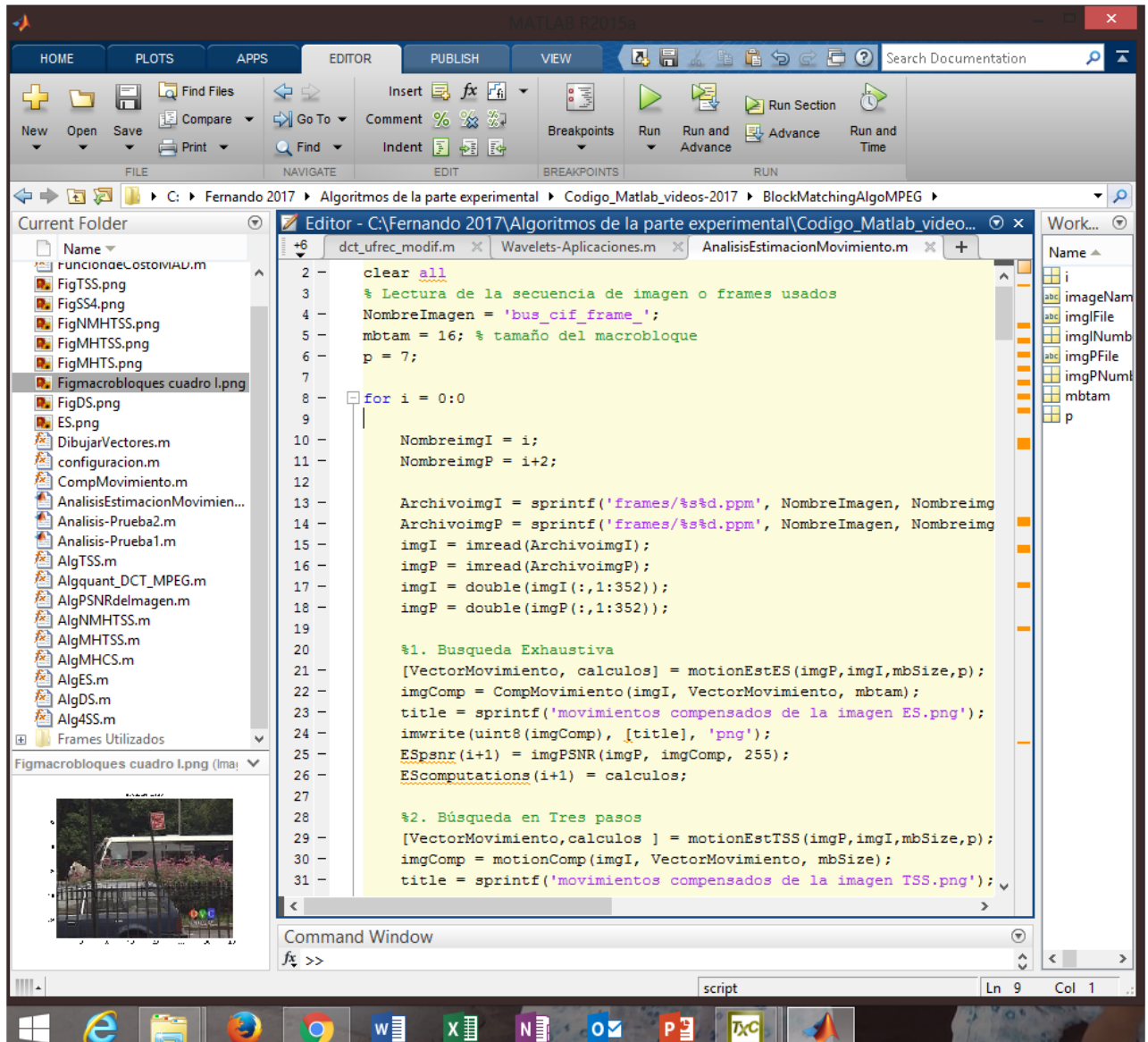


Figura 5: Implementación de los algoritmos heurísticos

5. IMPLEMENTACIÓN Y RESULTADOS

5.1. Aplicación de la DCT

Para la implementación de los algoritmos se tenían dos alternativas, la primera es a partir de la teoría, bases matemática y programarlos en su totalidad; la segunda opción es utilizar alguna de las librerías o funciones específica del software Matlab, ya que el desarrollo matemático como lo pudimos notar a lo largo de los capítulos anteriores es muy extenso y la codificación en

programas de software nos restaría mucho tiempo, necesario para poder alcanzar los objetivos de la tesis.

Las funciones a utilizadas se implementaron y se codificaron en el software *Matlab* que tendrá como entrada una variable de tipo matriz que representa a una imagen en niveles de gris (blanco y negro) o a color con sus respectivos niveles. En el caso que se utilice una imagen a color y se necesite en sus niveles de gris, se utilizará la función de *Matlab* *rgb2gray()*. A esta matriz de entrada, que al ser en niveles de gris será bidimensional, se le calculará la *DCT* – *2D* mediante la función *dct2()*, y finalmente se representarán gráficamente sus coeficientes mediante la función *mesh()* para medir el desempeño cuando no se logra apreciar la calidad visual. Para realizar la selección de coeficientes de la *DCT* por frecuencia o con base a su valor absoluto, se implementan las funciones *dct_ufrec_modif* y *dct_valor_absoluto_modif* que utiliza los percentiles (garantizados por la función *prctile*) en vez de dividir el el rango de los coeficientes de la *DCT* en intervalos iguales, puesto que estos no siempre nos garantizan que estemos seleccionando realmente un buen porcentaje de coeficientes de la *DCT* que deseamos.

Además, notamos el efecto bloque o cuadro una diferencia bien pronunciada que se origina al dividir la imagen en bloques de diferentes tamaños, al aplicar la la transforma a cada bloque, eligiendo el mismo porcentaje de coeficientes de la *DCT* en ambos casos en reconstrucción de la imagen. Por otro lado, detectamos que tanto al dividir la imagen en bloques de tamaño 2×2 como al dividirla en bloques de 8×8 , al aplicar la *DCT* a cada bloque y reconstruir la imagen, la misma aparece pixelada, sobre toda la imagen si los bloques considerados son de 2×2 , independientemente de que si aplicamos un cuantizador uniforme a toda la imagen, si seleccionamos coeficientes en base a su frecuencia o en base a su valor absoluto.

Cabe señalar que al tomar el 16 % y 25 % de los coeficientes de la *DCT* se sigue apreciando este mismo artifact o efecto al reconstruir la imagen, pero este hecho sólo se logra apreciar si observamos fijamente la imagen. Presentamos la imagen de *Lena* reconstruida tomando de cada bloque de 8×8 los coeficientes *DCT*(0, 0) y *DCT*(0, 1).

5.2. Compresión alcanzada de la transformada discreta del coseno

Todo lo analizado en los ítems anteriores nos permitió concluir que la compresión alcanzada utilizando la selección por valor absoluto sería mayor o igual que la alcanzada utilizando la selección por frecuencia, ya que, podemos considerar que todos los coeficientes de la *DCT* son del mismo orden (quizás el único coeficiente de mayor valor respecto a los demás es el primer coeficiente *D*(0, 0), es una de la razones principales por la que debe de ser incluido en cualquier selección que se realice, ya sea, tanto por frecuencia como por valor absoluto).

5.3. Transmisión progresiva transformada discreta del coseno

Para el análisis de este apartado anexamos seis archivos *.m*, es la extensión, que al ejecutar las implementaciones de los algoritmos producen vídeos. Estos archivos son: *Trans-*

ProgresivAmberSurfc.m, *TransProgresivbear.m*, *TransProgresivFerRudy.m*, *TransProgresivLena.m*, *TransProgresivMandrill.m* y *TransProgresivPuente.m*.

En los primeros cinco experimentos logramos observar como la imagen se reconstruye mejor a medida que el porcentaje de coeficientes de la DCT aumenta. Como ya mencionamos, la reconstrucción es más rápida seleccionando los coeficientes de la DCT por valor absoluto, pero en ambos casos lo que observamos es que la imagen va siendo reconstruida desde los detalles más gruesos a los detalles más finos, es decir, al principio vamos obteniendo como un bosquejo de la imagen y a medida que se agregan coeficientes de la DCT van apareciendo los detalles, como se ve sobretodo en *TransProgresivMandrill.m*, *TransProgresivLena.m* en donde los detalles más finos que son los pelos del bigote del mandril, así como las plumas del sombrero de Lena aparecen como últimos detalles en la reconstrucción de la imagen.

La justificación de esto es directa en la reconstrucción por frecuencia, ya que elegimos de las frecuencias más bajas a las frecuencias más altas y precisamente siendo los detalles más finos en la reconstrucción son aportados por las frecuencias más altas.

Otro punto de interés que observamos es que a partir de un porcentaje no tan alto de coeficientes los cambios en la reconstrucción de la imagen son imperceptibles a nuestro ojo debido a que a partir de cierto porcentaje no muy alto de coeficientes la imagen reconstruida nos resulta igual a la original. Sin embargo la imagen se va modificando aunque nuestro ojo no sea capaz de capturar estos detalles. Esto podemos visualizarlo con la función *surf* de *Matlab* que gráfica la relación entre cada píxel de la imagen y su valor de gris. Por ejemplo, proponemos ver el vídeo generado por el archivo *TransProgresivAmberSurfc.m*. Al observar la reconstrucción de la imagen vemos que a partir de cierto momento nuestro ojo no capta diferencias entre las imágenes reconstruidas y la original, la función *surf* permite observar que sigue habiendo cambios. Además, se pudo apreciar que al tomar un área de interés y ampliar la imagen (zoom), los detalles se observan a gran escala; pronunciándose en gran medida el efecto bloque según el tamaño de bloques elegidos. Así como, el efecto sombra.

5.4. Recuperación de Imágenes de Color en el Dominio de la Wavelet

El algoritmo empleado para calcular el descriptor binario de Haar, consiste una serie de pasos que se mencionan a continuación:

1. Se obtiene el **histograma color** utilizando espacio de color **RGB**.
2. Se determinan los **coeficientes de Haar** del histograma mediante el uso de la transformada wavelet.
3. Se **Binariza** los coeficientes de Haar.
4. Se aplica la distancia de **Hamming**.

En el caso de la utilización de los histogramas de color de la query, en los diferentes experimentos se logró ver que los tonos claros de la imagen corresponden a los valores de la derecha histograma.

A partir de los histogramas de las imágenes, podemos decir que el detalle de la imagen presenta una tonalidad dominante oscura, puesto que se concentra en las sombras. También, podemos notar que existe un detalle de una imagen con tonalidad dominante clara, en las partes con más iluminaciones y el detalle de una imagen con tonalidad media, que corresponde a tonos medios. Una imagen con una gama tonal completa tiene píxeles en todas las áreas. Al identificar la gama tonal de la imagen nos ayuda a determinar las correcciones tonales correspondientes.

5.4.1. Resultados de las muestras de las imágenes recuperadas

Para la realización de los experimentos, primero tratamos de obtener algunas imágenes digitales, con el fin de obtener la muestra, seleccionamos 3 imágenes grandes de texturas del conjunto de imágenes que se encuentran en [29] página web del grupo de investigación (JAMS Z. WANG) con las siguientes características:

- Que 2 de ellas tengan el mismo tipo de textura, y 2 el mismo color.
- Se quiere que 2 de las 3 tengan una textura similar, pero de distinto color.
- Y que 2 de las 3 tengan texturas diferentes de color similar. Son 3 imágenes de textura (i.e, pasto verde, pasto rojo, ladrillo rojo) ya que, cumplen con estas características.
- De cada una de las 3 imágenes se extraen 3 bloques de tamaño 128×128 , es decir se obtiene 9 imágenes de menor tamaño, pero con características similares.
- Se agregan 4 imágenes pequeñas de tamaño 128×128 , que tengan el mismo color predominante que las grandes, por ejemplo 2 de un color y 2 de otro.
- Se toma una imagen pequeña de cada imagen grande y se realiza una rotación con alguna aplicación o software como: photoshop o con MATLAB en 45 y 90 grados. O bien se rota la imagen grande y se extrae una de las imágenes pequeñas, es decir resultando 6 imágenes chicas. Con todo este proceso construimos nuestra base de datos de imágenes, las cuales son 19 en total, las que nos servirán para realizar los diferentes experimentos, y realizar las diferentes medidas y comparaciones hasta obtener la imagen recuperada más similar a la original.

Después, de haber realizado las diferentes corridas o simulaciones correspondientes a la *query image I*, nos quedamos con el siguiente grupo de imágenes las que presentan menor distancia entre las muestras construidas, se obtuvo qué algoritmo encuentra las dos imágenes más próximas a la imagen de consulta, encontrando una imagen similar en el paso 2 con una distancia 13.0208, y que coincide con la imagen dividida de la esquina superior con respecto a la imagen de consulta o de prueba.

En el segundo grupo de experimentación que corresponde a la *query image II*, obtuvimos qué algoritmo encuentra dos imágenes similares en el paso 1 y 2 con una distancia 0.52083 y 1.0417, y una de ellas coincide con la imagen de consulta o de prueba.

En el tercer grupo de experimentación, correspondiente a la *query image III* el algoritmo halla las imágenes similares en el paso 1 y 2 con una distancia 3.125 y 3.125, y que coincide

con la imagen de consulta o de prueba.

Lo cual podemos concluir, que si tomamos en cuenta la imagen original al grupo de prueba, el algoritmo encuentra la imagen original y ubica las más parecidas a ella. Además, afirmamos que:

1. Dos imágenes comparadas a través de sus histogramas globales de color pueden no estar semánticamente relacionadas a pesar de compartir una distribución similar de color.
2. Los descriptores binarios de la Wavelet de Haar de las imágenes de interés se compara con la de consulta utilizando la distancia de Hamming.
3. Las imágenes más similares son recuperadas y ordenadas según la distancia a la de consulta, como se observó en los diferentes experimentos realizados.

5.5. Algoritmos de Coincidencia para la Estimación de Movimientos

La implementación de algoritmos de estimación de movimiento es una de las tareas claves del proceso de compresión de la imágenes y secuencia de ellas, nos lleva un buen porcentaje en el tiempo de procesamiento, como se ha mencionado en el capítulo 3. Los algoritmos de coincidencia por bloques son desarrollados con la finalidad de reducir la complejidad computacional de los algoritmos, debido a su eficiencia y el buen rendimiento. Las estrategias de búsqueda de ciertas características u objetos, es uno de los factores principales en el desarrollo de algoritmo de estimación de movimiento ya que tiene el potencial de proporcionar un buen rendimiento.

Los algoritmos propuestos se modifican en base a la estrategia de búsqueda, según el enfoque de algoritmos de estimación de movimientos estándar. Estos se mejoran considerablemente mediante la implementación de algoritmos heurísticos y sus híbridos, proporcionando mejores resultados para PSNR sin aumentar el número de cálculos.

5.6. Secuencia de prueba para el conjunto de algoritmos de estimación de movimientos

Los experimentos realizados para el análisis del desempeño del conjunto de algoritmos descritos en el capítulo 3, consistieron en codificar una de secuencias de imágenes de prueba *bus_cif_frame*, selecciona del conjunto de imágenes que se encuentran en [24] página web de Video Trace Library (ARIZONA STATE UNIVERSITY), con ciertas características específicas, para realizar las comparaciones del rendimiento de los diferentes algoritmos, tomando las mediciones del número de puntos de búsqueda, del error en la estimación y compensación de movimientos generados.

Además, se describe las condiciones de los experimentos, en las que se utilizan 150 frame de las secuencias vídeo seleccionado de la misma página. Después, de las diferentes ejecuciones de los algoritmos, decidimos tomar un tamaño bloque de 16×16 píxeles y con un parámetro de búsqueda $p = \pm 7$, puesto que nos presenta mejores rendimientos y calidad visual en la

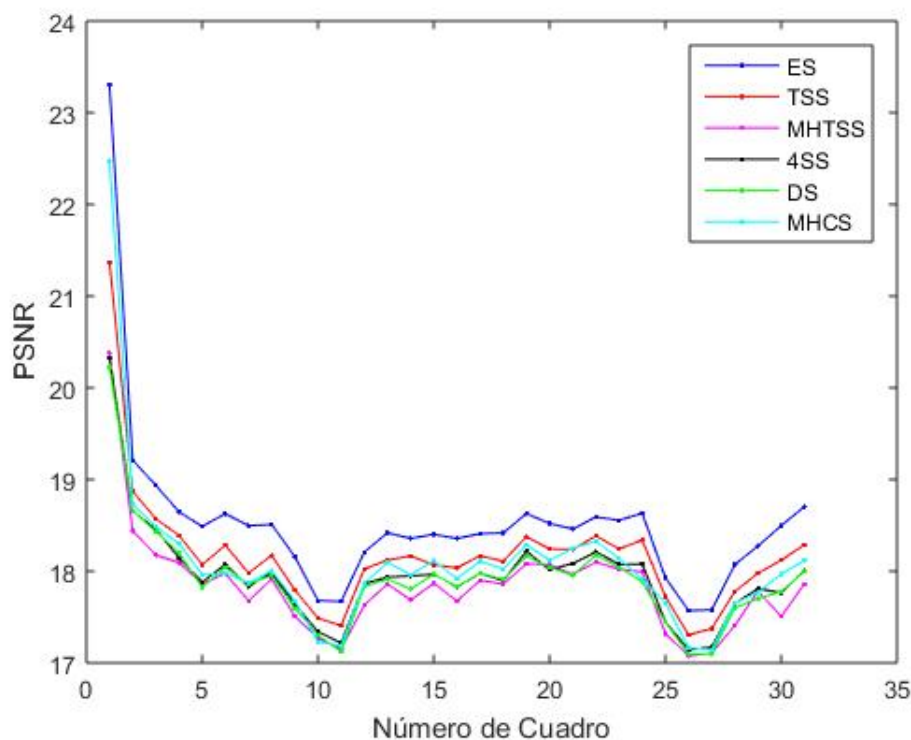
reconstrucción. Además, se presentan y comentan los resultados obtenidas tras la ejecución de los diferentes algoritmos desarrollados e implementados en el software *Matlab*. Los que se detallan a continuación:

- Algoritmo de búsqueda exhaustiva.
- Algoritmo de búsqueda en tres pasos.
- Algoritmo de búsqueda en cuatro pasos.
- Búsqueda en diamante.
- Algoritmo metaheurístico búsqueda en tres pasos simple y eficiente.
- Algoritmo metaheurístico mejorado en el patrón de búsqueda en tres pasos.

Los diferentes algoritmos se basan en las funciones de costo o métrica tratadas en el capítulo 3, que calculan el error entre dos macrobloques, de las que se eligieron *MAD* abordada en la sección 2.2.11 y la función de costos *MSE* abordada en 2.2.9, la elección de ellas se debe a que son las más conocidas y sencillas de implementar.

5.6.1. Resultados de Implementación de los Algoritmos Heurísticos de Coincidencia para la Estimación de Movimientos

La evaluación de la heurística mejorada según el patrón de búsqueda en cruz *MHCS* implementada nos da un rendimiento medio, con respecto al resto de los algoritmos, y se comporta bien, tanto, computacionalmente cómo en términos de calidad visual, según las diferentes experimentaciones realizadas. Esto observa a continuación:



Todo lo anterior lo podemos fundamentar mediante la visualización del histograma de ren-

dimiento de todos los algoritmos, notamos que el algoritmo *ES* es el que utiliza mayor costo computacional, según la métrica empleada. Los algoritmos heurísticos *TSS*, *4SS* y *DS* utilizan aproximadamente el mismo tiempo de ejecución, es decir se comportan de forma similar, los metaheurístico *MHTSS* y *MHCS* trabaja de forma media y sus implementaciones son en promedio mejor con respecto a los demás algoritmos, debido a las mejoras de los parámetros variados.

6. Conclusiones

En esta investigación, nos ocupamos de la estimación de movimiento $2-D$ disponible. Para este fin, se plantearon dos líneas principales de estudio: los que están desarrollados sobre los métodos de dominio de frecuencia, empleando texturas estáticas y la implementación de los métodos de dominio temporal utilizando texturas dinámicas, para su aplicación en la solución de problemas complejos de interés, como es la estimación de movimientos y compensación en la compresión de secuencias de imágenes o vídeo.

Se reviso y realizó un estudio unificado de todas las medidas de compresión, de la transformadas utilizadas con frecuencias y centrando nuestro interés en la transformada discreta del coseno y de la Wavelets, que son esenciales para la parte experimental de los algoritmos de estimación de movimientos, en especial las variantes en los heurísticos, ofreciendo un modelo formal de ellas. Se desarrollo el modelo formal, permitiendo caracterizar las heurísticas, observando los puntos en común que poseen los diferentes algoritmos de compresión y de búsqueda en los macrobloques de referencia, y centrándonos principalmente en la mejora de la calidad visual, la cantidad de información comprimida o reconstruida y el comportamientos de los métodos en la búsqueda de similitudes entre píxeles o imágenes. Los algoritmos son suficientemente genéricos para abordar los modelos existentes y sus variantes, pero a la vez permiten controlar detalles de bajo nivel, que son especialmente interesantes en el tratamiento específico de las heurísticas. También, se estructuraron los resultados teóricos de la teoría matemática esencial para el diseño e implementación de los Algoritmos de Coincidencia para la Estimación de Movimiento en compresión de imágenes.

Se diseñaron e implementaron los códigos de la funciones, procedimientos fundamentales de los algoritmos del dominio de frecuencia (DCT, Wavelets) y Heurísticos (FS, TTS, 4SS, DS, MHTSS, MHCS) de Coincidencia para la Estimación de Movimiento en compresión de imágenes, escritos en el lenguaje de Programación Matemático *MATLAB*.

Los resultados generales muestran que al aplicar filtrado de los coeficientes de la DCT con base a valor es mejor que los coeficientes seleccionados por frecuencia, independiente del tamaño del macrobloque utilizado. En la que se realizaron diferentes corridas de ambos métodos, y sus posibles combinaciones, en las que se observó las mejoraras del desempeño, y la calidad visual de las imágenes reconstruidas. Este experimento nos sirve para afirmar que una selección por valor absoluto daría mejores resultados visuales ya que eliminan las altas frecuencias

y dejan las más bajas.

Los experimentos de variar los parámetros y dividir la imágenes en bloques de diferentes tamaños, nos permitió decidir que puede ser un buen tamaño los bloques de 8×8 , ya que obtuvimos un pixelado menor que al dividir la imagen en bloques de tamaño más pequeños. Para tamaños ≥ 8 mejora considerablemente el tamaño la calidad visual y el nivel de compresión de las imágenes analizadas.

En el caso de la parte experimental de la Wavelets pudimos apreciar que en las diferentes ejecuciones de los algoritmos implementados, que al incluir la imagen original a la base de datos de prueba el algoritmo implementado lo encuentra en el paso 1 o 2 con una distancia mínima, o muestra aquella que se encuentra a una distancia mínima. Además, dos imágenes comparadas a través de sus histogramas global de color pueden no estar semánticamente relacionadas a pesar de compartir una distribución similar de color. Los descriptores binarios de la Wavelet de Haar de las imágenes de interés se compara con la imagen de consulta utilizando la distancia de Hamming. Por tanto, las imágenes más similares son recuperadas y ordenadas según la distancia a la de consulta.

El tercer conjunto de experimentos realizados en este trabajo, se centra en la implementación y evaluación del rendimiento del conjunto de algoritmos heurísticos de coincidencia para estimación de movimientos de búsqueda rápida, bajo el contexto de los codificadores normalizados *MPEG – JPEG*. Para concluir, los resultados demuestran que los algoritmos que presentan un mayor rendimiento son los heurístico mejorados *MHCS* y *MHTSS* debido a que son más eficiente y por su calidad visual, y en menor medida en orden creciente de complejidad computacional: *DS*, *4SS*, *TS*, y *ES*. En base al análisis realizado en este trabajo se considera estos algoritmos como los mejores en promedio.

7. Perspectiva de la Investigación

Como trabajo futuro, sería muy útil para ambas líneas tener una implementación más eficiente de los heurísticos, trabajar aún con la variación de los parámetros y las medidas de desempeño, pensar en el diseño de metaheurísticas para lograr mejores resultados. Además que se debe aprovechar el auge de los hardware en paralelo o la utilización de clúster de computadoras para agilizar los procesos de rendimiento, o en el caso que se trabaje con ordenadores de índole personal y con un buen poder de cálculo. Ya que una de las desventaja observadas de la líneas de investigación tratada, es que estos algoritmos de búsqueda son muy exigentes en términos de rendimiento computacional, desde el software utilizado que requiere de una buena capacidad de memoria caché y un buen procesador para su ejecución, ya que los hacen impracticable para aplicaciones en tiempo real, cuando la imagen o secuencia tiene mucho información que analizarse. Queda abierto el trabajo de los códigos en *C++* para la mejora del rendimiento de los mismos.

Referencias

- [1] Alegre, E., Sánchez, L., Fernández, R. Á., and Mostaza, J. C. (2003). “*Procesamiento Digital de Imagen: fundamentos y prácticas con Matlab*”. Universidad de León.
- [2] Ayer, S., Sawhney, H. (1995). “*Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding*”. ICCV, pp. 777-784.
- [3] Blum, C., and Roli, A. (2003). “*Metaheuristics in combinatorial optimization: Overview and conceptual comparison*”. ACM Computing Surveys (CSUR), 35(3), 268-308.
- [4] Book, R. V. (1980). Michael R. Garey and David S. Johnson, “*Computers and intractability: A guide to the theory of NP completeness*”. Bulletin (New Series) of the American Mathematical Society, 3(2), 898-904.
- [5] Bovik, A. C. (2010). “*Handbook of Image and Video Processing*”. Academic Press.
- [6] Brox, T., Bruhn, A., Papenber, N., and Weickert, J. (2004). “*High accuracy optical flow estimation based on a theory for warping*”. Computer Vision-ECCV, 25-36.
- [7] CH, C. S., Ratnam, J. V. K., and Student, P. G. “*Comparison of Fast Block Matching Algorithms for Motion Estimation*”.
- [8] Chen, W. H., Smith, C. H., and Fralick, S. C. (1977). “*A fast computational algorithm for the discrete cosine transform*”. IEEE Transactions on communications, 25(9), 1004-1009.
- [9] Cheung, C. H., and Po, L. M. (2002). “*A novel cross-diamond search algorithm for fast block motion estimation*”. IEEE transactions on Circuits and Systems for Video Technology, 12(12), 1168-1177.
- [10] Cook, S. A. (1971, May). “*The complexity of theorem-proving procedures*”. In Proceedings of the third annual ACM symposium on Theory of computing (pp. 151-158). ACM.
- [11] Cuevas, E., Zaldívar, D., Pérez-Cisneros, M., Sossa, H., Osuna, V. (2013). “*Block matching algorithm for motion estimation based on Artificial Bee Colony (ABC)*”, Applied Soft Computing Journal 13 (6), pp. 3047-3059.
- [12] DUFAUX, F. (2010). “*Motion Estimation Techniques for Digital TV: A Review and a New Contribution*”.
- [13] Ghanbari, M. (1990). “*The cross-search algorithm for motion estimation (image coding)*”. IEEE Transactions on Communications, 38(7), 950-953.
- [14] Glover, F. W., and Kochenberger, G. A. (Eds.). (2006). “*Handbook of metaheuristics*” (Vol. 57). Springer Science & Business Media.
- [15] Evan Herbst, Xiaofeng Ren, and Dieter Fox. (2013) “*Rgb-d flow: Dense 3-d motion estimation using color and depth*”. In Robotics and Automation (ICRA), IEEE International Conference on, pages 2276-2282. IEEE, 2013.
- [16] Jagalingam, P., and Hegde, A. V. (2015). “*A review of quality metrics for fused image*”. Aquatic Procedia, 4, 133-142.

- [17] Lee, B. (1984). "A new algorithm to compute the discrete cosine transform". IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(6), 1243-1245.
- [18] Li, R., Zeng, B., and Liou, M. L. (1994). "A new three-step search algorithm for block motion estimation". IEEE transactions on circuits and systems for video technology, 4(4), 438-442.
- [19] Lu, J., and Liou, M. L. (1997). "A simple and efficient search algorithm for block-matching motion estimation". IEEE Transactions on Circuits and Systems for Video Technology, 7(2), 429-433.
- [20] Lucas, B. D., & Kanade, T. (1981). "An iterative image registration technique with an application to stereo vision". In IJCAI, pages 674-679, 1981.
- [21] Nie, Y., and Ma, K. K. (2002). "Adaptive rood pattern search for fast block-matching motion estimation". IEEE Transactions on image processing, 11(12), 1442-1449.
- [22] Pourreza, H. R., Rahmati, M., and Behazin, F. (2002). "Adaptive Pixel Difference Classification, an Efficient and Cost Effective Algorithm for Motion Estimation". Threshold, 3(4), 3.
- [23] Reeves, C. R. (1996). "Modern heuristic techniques. Modern heuristic search methods", 1-25.
- [24] Reisslein, M. (2005-2011). "Video Trace Library ". Arizona State University, [online] Available: <http://trace.eas.asu.edu>.
- [25] Ruiz, V. González. (2000). "Compresión reversible y transmisión de imágenes". PhD thesis, Universidad de Almería.
- [26] Stiller, C., and Konrad, J. (1999). "Estimating motion in image sequences". IEEE Signal Processing Magazine, 16(4), 70-91.
- [27] Utenpattanant, A., Chitsobhuk, O., and Khawne, A. (2006, February). "Color descriptor for image retrieval in wavelet domain". In Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference (Vol. 1, pp. 4-pp). IEEE.
- [28] Verma, N., Sahu, T., and Sahu, P. (2012). "Efficient motion estimation by fast three step search algorithms". International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 1(5).
- [29] Wang, James Z. (2012-2017). "MODELING OBJECTS, CONCEPTS, AESTHETICS AND EMOTIONS IN BIG VISUAL DATA ". JAMES Z. WANG RESEARCH GROUP, [online] Available: <http://wang.ist.psu.edu>.
- [30] Yaakob, R., Aryanfar, A., Halin, A. A., and Sulaiman, N. (2013). "A comparison of different block matching algorithms for motion estimation". Procedia Technology, 11, 199-205.