



XVII SIMPOSIO DE INGENIERÍA ELÉCTRICA (SIE-2017)

Título

Plataforma para codificación en HEVC/H.265

Title

HEVC / H.265 Encoding Platform.

Rafael Alejandro Olivera Solís¹, Vitalio Alfonso Reguera²,

1-MSc. Rafael Alejandro Olivera Solís, Universidad Central "Marta Abreu" de las Villas, Cuba, rolivera@uclv.edu.cu

2- Dr. C. Vitalio Alfonso Reguera, Universidad Central "Marta Abreu" de las Villas, Cuba, vitalio@uclv.edu.cu

Resumen:

HEVC/H.265 es el nuevo estándar de codificación de vídeo. Su objetivo principal es mejorar los niveles de compresión de vídeo. La compresión de video ha jugado un papel fundamental en el desarrollo de las aplicaciones multimedia. La necesidad de reducir el bitrate, y a su vez mantener su calidad lo mayor posible, ha traído consigo la utilización de herramientas que permitan un proceso de codificación eficiente y eficaz. Este trabajo propone una plataforma de codificación basada en herramientas de software libre y el uso de un clúster de computadoras, lo cual garantiza la eficiencia en el proceso de codificación.

Abstract:

HEVC / H.265 is the new video encoding standard. Its main objective is to improve the levels of video compression. Video compression has played a key role in the development of multimedia applications. The need to reduce the bitrate, while maintaining its quality as much as possible, has brought with it the use of tools that allow an efficient and effective coding process. This work proposes a coding platform based on free software tools and the use of a computer cluster, which guarantees efficiency in the coding process.

Palabras Clave: Compresión, FFMPEG, HEVC/H.265, HPC

Keywords: Compression, FFMPEG, HEVC/H.265, HPC.





1. Introducción

En la sociedad actual es innegable la importancia que tiene el vídeo como soporte de información, tanto en ámbitos empresariales (videoconferencias y seguridad) como de ocio (cine, televisión, entre otros). Redes sociales como *Facebook*, *MySpace* y servicios como *YouTube* realizan cada vez un mayor uso del video en su oferta. La difusión por satélite, la Televisión por Protocolo de Internet (IPTV) y la Televisión Digital Terrestre (TDT) se ven afectadas por el limitado ancho de banda, así como por el reducido espectro de la señal. Por tanto, es necesario emplear técnicas de compresión que reduzcan de forma drástica el régimen binario de la señal de video codificada, manteniendo la misma calidad visual. En 2013 se estandariza HEVC/H.265 (*High Efficiency Video Coding*), el formato de compresión de video de alta eficiencia. Este estándar fue desarrollado por ITU-T VCEG (*Video Coding Experts Group*) e ISO/IEC MPEG (*Moving Picture Experts Group*). Su objetivo principal fue mejorar la compresión de vídeo, en relación a los anteriores estándares [1], [2]. El nuevo estándar HEVC permite un importante avance en la compresión con respecto a sus predecesores [3]. Las aplicaciones para HEVC no sólo cubrirán el espacio de los conocidos usos actuales y capacidades del video digital, sino que también incluirán el despliegue de nuevos servicios y la entrega de una calidad de video mejorada, como el despliegue de imágenes de Televisión de Ultra Alta Definición (UHDTV) y video con mayor rango dinámico, una gama más amplia de color representable y mayor precisión de representación de lo que se suele encontrar hoy en día [3]. Desde sus inicios, la comunidad científica estuvo preocupada por el soporte, tanto de hardware como de software, que pudiese acometer esta tarea. En [4] implementan un codificador HEVC sobre los modernos procesadores Intel x86, donde se identifican los módulos más eficientes del codificador HM 6.2 a partir de la compensación del movimiento, el cálculo de suma de diferencia (SAD/SSD) y la transformación entera. Luego se explotan los métodos SIMD (*single-instruction-multiple-data*) para optimizar el rendimiento computacional de estos módulos. Se comprueba que estos procesadores x86 son más eficientes que los procesadores Intel i5-750 bajo estas condiciones. En [5], [6] se proponen estrategias y técnicas de procesamiento paralelo que mejoran sustancialmente el proceso de codificación en HEVC/H.265. En [7] se propone una estrategia para la aceleración del proceso de codificación paralela utilizando la estimación de movimiento con bloques de tamaños variables. Este trabajo evoluciona las plataformas anteriores pues incorpora una plataforma GPU de muchos núcleos con CPU multi-





núcleos. Solucionan el problema de la sincronización de los subprocesos entre las CPU y la GPU a partir de una estrategia basada en WPP. En [8] se presenta un codificador de software HEVC paralelo de alto nivel adecuado para aplicaciones de transmisión o de flujo de red. Se proponen 10 perfiles de codificación en tiempo real de videos 4K UHD a 60 fps, lo cual se logra a través de una eficiente plataforma de codificación paralela compuesta de CPUs interconectadas por la red de alta velocidad a través de paralelismo temporal de GOP (*Group of Picture*) y paralelismo espacial de imagen codificando a nivel de *slices*.

En los trabajos anteriores se puede constatar la evolución de las plataformas de codificación para este estándar. En la mayoría de las investigaciones se utilizan los codificadores HEVC tanto de la familia HM como las librerías x265 de código abierto. Ello hace del manejo del estándar HEVC/H.265 una tarea engorrosa para ordenadores simples, incluso con buenas prestaciones. Por este motivo, el empleo de un clúster de alto rendimiento (HPC), en el que se distribuya el procesamiento entre todos los procesadores de un mismo nodo, constituye una solución ante la problemática del procesamiento de la señal de video en el proceso de codificación. La utilización de la plataforma FFMPEG es otra de las soluciones poco exploradas en el manejo de este estándar. Esta herramienta fue la primera que contó con la librería x265, además de brindar múltiples bondades para el trabajo con la señal de video. En este artículo se propone una plataforma de codificación del estándar HEVC/H.265 utilizando la *Suite* FFMPEG en un clúster de alto rendimiento. El trabajo se organiza de la siguiente manera. A continuación, se caracteriza el estándar HEVC/H.265, haciendo énfasis en los nuevos elementos que contiene. Luego se realiza una descripción del clúster de alto rendimiento, así como sus elementos fundamentales. También se dan a conocer las principales características de FFMPEG. Por último, a manera de ejemplo, se realiza la codificación de materiales audiovisuales sin compresión utilizando H.265 y se compara con su similar en H.264.

2. Metodología

2.1 HEVC/H.265

Como parte de los avances en el campo de la codificación de fuente llevados a cabo por la comunidad científica, se aprueba la primera fase del estándar de compresión H.265/HEVC a inicios del 2013. Este se perfila como el sucesor natural del estándar de compresión H.264/AVC (*Advanced Video Coding*). En la figura 1 se muestra el diagrama en bloques de un codificador-decodificador HEVC/H.265.



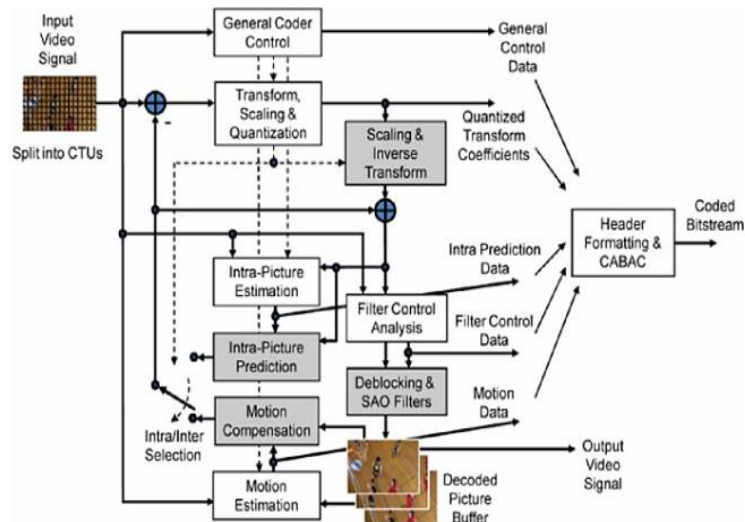


Figura 1. Diagrama en Bloques del codificador HEVC/H.265 [9].

HEVC/H. 265 mejora las herramientas existentes en los estándares previos, especialmente en H.264, logrando una mejora en la calidad percibida y permitiendo una reducción de bitrate por encima del 50% con respecto a H.264. H.265 pretende ser altamente eficiente en dos sentidos: lograr eficiencia en formatos de muy alta resolución como UHDTV (*Ultra High Definition Television*), y hacerlo también en entornos de muy baja tasa binaria, como es el caso de servicios *streaming*, *WebTV* y *OTT (Over the Top Content)*. H.265 también utiliza la estimación y la compensación de movimiento para aprovechar la similitud temporal de las secuencias, y además emplea la transformada discreta del coseno, con cierta variación, con el objetivo de aplicar la cuantificación en el dominio transformado, y codificar estadísticamente dichos coeficientes.

Para ello H.265 cuenta con diferentes estructuras novedosas que se analizarán a continuación.

- 1. Nueva Unidad de Codificación:** Define una nueva estructura de codificación que diverge de las anteriores unidades denominadas macro bloques con dimensiones de 16x16 píxeles. En este estándar la unidad de codificación se denomina *Coding Tree Block (CTB)* y sus dimensiones pueden ser desde 8x8 hasta 64x64 píxeles. En HEVC, una imagen se divide en diferentes CTU (*Coding Tree Units*) que pueden presentar un tamaño máximo de LxL píxeles, para el caso de la luminancia L puede tomar los valores de 16, 32 ó 64, con los tamaños más grandes que permiten típicamente una mejor compresión [10].



2. **Unidad de Predicción:** H.265 define una nueva unidad de predicción (PU) que pueden tener dimensiones desde 32x32 hasta 4x4, lo que representa la mitad que la menor de las unidades de codificación. Al igual que su antecesor H.264, pueden aplicarse de modo *intra-frame* o *inter-frame*, utilizando divisiones distintas.
3. **Predicción Intra-Frame:** HEVC introduce un nuevo método de predicción *intra-frame*, el intra-angular, el cual define 33 predicciones direccionales para todos sus posibles tamaños de PU el cual se diferencia de los 2 predictores direccionales que define H.264 para tamaños de 16x16, y 8 predictores para 8x8 y 4x4. Además, se definen los modos intra-planar, donde se supone una superficie de amplitud con una pendiente horizontal y vertical derivada de los límites, y el modo intra-DC donde se asume una superficie plana con un valor que coincida con el valor medio de las muestras de contorno [11].
4. **Predicción Inter-Frame:** La estimación de movimiento en H.265 se lleva a cabo con la precisión de $\frac{1}{4}$ de píxel con filtros interpoladores de *7-taps* y *8-taps*, y se incorpora además la AMVP (*Advanced Motion Vector Prediction*) para lograr la señalización de sus vectores de movimiento mediante el cálculo de los vectores de movimiento más probables obtenidos desde los bloques vecinos [12].
5. **Unidades de Transformación:** Se define una nueva TU (*Transform Unit*). Se aplica de igual forma la DCT (*Discrete Cosine Transform*) con una ligera modificación con respecto a la utilizada en H.264, también permite bajo ciertas condiciones la utilización de la DST (*Discrete Sine Transform*). En el caso de H.265 se amplían los tamaños de transformación desde 4x4 a 32x32.
6. **Sample Adaptive Offset (Compensación Adaptable de la muestra):** Nueva herramienta aplicada luego del *In-loop filter* con el objetivo de lograr mejorar la calidad subjetiva de la imagen decodificada, aplicando pequeños *offset* a los píxeles decodificados, en función de la zona a la que pertenecen [2].

Debido a la alta complejidad de los algoritmos introducidos por H.265, el estándar integra novedosas herramientas con el objetivo de frenar la codificación de la imagen. HEVC permite la codificación del *frame* en dos nuevos tipos adicionales al *slice*, que existía previamente en los anteriores estándares, denominados *Tile* y Frente de Ondas o WPP (*Wavefront Parallel Processing*).

Tile: define regiones descifrables rectangulares de la imagen, típicamente con el mismo número de CTU, éstas pueden ser codificadas independientemente y comparten algunas





cabeceras de información. Los tiles tienen como propósito fundamental aumentar la capacidad para el procesamiento paralelo en lugar de proporcionar la capacidad de recuperación de errores. Múltiples tiles pueden compartir información de cabecera por estar contenida en el mismo *slice*. Alternativamente, un solo tile puede contener múltiples *slices*. Los tiles proporcionan cierto nivel de paralelización en los procesos de codificación y decodificación de vídeo que puede ser explotado en sistemas con arquitecturas de procesamiento paralelo siendo más flexibles que las *slices* de H.264 además de considerarse una tecnología menos compleja que FMO (*Flexible Macroblock Ordering*) [13].

WPP: Es una herramienta utilizada por HEVC en el procesamiento en paralelo. Cuando WPP se encuentra habilitado, una *slice* se fracciona en filas de CTU. La primera fila puede ser procesada por un hilo de ejecución; en el caso de un segundo hilo de ejecución puede comenzar con la segunda fila después de haberse procesado las dos primeras CTU de la fila anterior. Al igual que los tiles, las WPP suministran una forma de paralelización en los procesos de codificación y decodificación. La diferencia radica en que con WPP se consigue un mejor rendimiento de compresión además de no permitir la introducción de determinadas aberraciones visuales como ocurre en los *tiles*.

2.2 HPC/Clúster:

Para el trabajo con HEVC es necesario, para realizar un proceso de codificación y decodificación lo más eficiente posible. Para ello es necesario la utilización de un clúster de computadoras. Se deben combinar las características de disponibilidad de almacenamiento, alta disponibilidad de recursos computacionales, balance de carga y alto rendimiento. En la Universidad Central "Marta Abreu" de las Villas (UCLV) se encuentra uno de los clústeres de computadoras de alto rendimiento (HPC) más potentes del país. Los clústeres de alto rendimiento utilizan los nodos para ejecutar cálculos simultáneos. Un clúster de alto rendimiento permite que las aplicaciones trabajen de forma paralela, mejorando así el rendimiento de éstas. Los clústeres de alto rendimiento son conocidos como clúster computacionales o computación de red [14]. En esta sección, el análisis se enfocará en las herramientas y procedimientos que posibilitan el correcto funcionamiento del clúster, específicamente en el gestor de recursos del HPC y el gestor de colas de trabajos. Estos son dos de los componentes esenciales en el funcionamiento del clúster de computadoras de la UCLV.





- **Gestor de Recursos *OpenPBS*:**

El gestor de recursos *OpenPBS* es el sistema de gestión de colas de trabajos. Es un gestor de recursos distribuidos que permite, que varios usuarios, grupos y proyectos puedan trabajar juntos usando una infraestructura compartida como, por ejemplo, un clúster de computación de alto rendimiento. En el entorno de *OpenPBS* la existencia de colas no tiene mucha importancia para el usuario. El usuario simplemente tiene que especificar los recursos que requiere su trabajo y *OpenPBS* se ocupa en colocarlo en la cola de trabajos más adecuada. *OpenPBS* presenta como funcionamiento principal los siguientes puntos resumidos a continuación:

1. Coloca trabajos en colas y planifica su ejecución junto con *Maui*.
2. Organiza los trabajos con el nodo de ejecución más apropiado.
3. Gestiona recursos.
4. Hace cumplir las políticas de asignación de trabajos (proyecto A es más importante que proyecto B).

- **Entorno de Usuarios (*Modules*):**

Diferentes programas como aplicaciones, compiladores o librerías requieren que las correspondientes variables de entorno estén definidas para que se ejecuten correctamente. Es posible definir estas variables de forma permanente en el perfil del usuario, pero eso suele crear problemas cuando se desea usar diferentes versiones de una aplicación, usar una vez el compilador *gcc* y en otro momento un compilador de Intel o crear varias versiones de programas con diferentes librerías MPI. La herramienta *modules* permite el cambio rápido de un entorno a otro debido a que permite cargar o borrar las variables necesarias para un programa ejecutando un comando sencillo y avisa si hay conflictos entre un entorno ya definido con otro que el usuario quiere cargar. También es posible que un módulo cargue otros módulos para facilitar el uso y poder crear paquetes de módulos.

- **Envío de trabajos:**

Los usuarios no pueden conectarse a los nodos computacionales, por lo que la ejecución de trabajos en los nodos computacionales tiene que realizarse, obligatoriamente, a través del sistema de colas Torque/PBS. El sistema de colas registrará por orden cada una de las solicitudes enviadas y las emitirá, para su ejecución en los nodos computacionales, cuando estén disponibles los recursos requeridos. El envío de trabajos se realiza a través



del comando **qsub**, cuyo argumento obligatorio es el nombre de un *script* de *shell*. El *script* tiene que disponer de permisos de ejecución. Dentro del *script*, el usuario debe indicar las acciones que se realizarán en los nodos, una vez que los recursos requeridos estén disponibles. Las opciones de configuración de Torque/PBS se dan en gran medida al sistema de colas PBS el cual permite a los usuarios configurar diferentes aspectos de la ejecución de los trabajos. Las instrucciones de configuración Torque/PBS se indican en los scripts a través de líneas que comienzan (sin espacios) con **#PBS**. También es posible indicar estas opciones directamente como argumentos de **qsub** en la línea de comandos.

- **Colas del Sistema:**

Existen 4 colas del sistema limitadas por el tiempo, cantidad de nodos y núcleos a emplear. Existe una cola de usuario (*Default*) que se encarga de encaminar los trabajos dentro de las colas del sistema de acuerdo a las necesidades de los mismos. Nunca el usuario puede acceder directamente a las colas del sistema. El procedimiento ocurre de la siguiente forma:

1. Los trabajos son asignados a la cola de nombre *Default* y automáticamente son ruteados a la cola del sistema adecuada, teniendo en cuenta los parámetros: Tiempo (*walltime*), número de nodos (*nodes*) y número de *cores* (*npp*).
2. La cola test es de acceso directo, esto quiere decir que los usuarios pueden acceder a ella de forma directa utilizando el parámetro **#PBS -q test** dentro del script de ejecución del trabajo. Es una cola con tiempo máximo de ejecución de 2 horas, que utiliza nodos de bajos recursos, pensada para que los usuarios puedan probar los *scripts* antes de lanzarlos.

También existen algunas limitaciones en cuanto al funcionamiento de las colas:

1. No se deben ejecutar los trabajos directamente en las colas de ejecución, siempre debe utilizarse la cola *Default*.
2. Hay un límite sobre la cantidad de trabajos por usuario en cada cola.

2.3 FFMPEG:

FFmpeg es una herramienta de alta rapidez en línea de comandos que facilita la conversión de archivos de audio y/o video de un formato a otro, capturar y codificar en tiempo real a partir de distintas fuentes de captura como por ejemplo una webcam o una tarjeta de TV, o realiza un *screencast*. Esta herramienta ofrece un manejo eficiente de los parámetros característicos de audio y video como imágenes por segundo (fps), resolución,





relación de aspecto, tasa de bits, compresión, número de canales de audio, entre otros. Esta plataforma fue la primera que contó con las librerías correspondientes a HEVC/H.265, lo que la califica como una de las pioneras en el uso de este estándar de codificación. FFmpeg es una solución completa de código libre y de plataforma cruzada. Es un software libre licenciado bajo LGPL o GPL (*General Public License*) en dependencia de la elección en las opciones de configuración [15], [16]. Es desarrollado bajo Linux y puede ser compilado para plataformas como *Redhat*, *RPMs*, *DEBs*, *Windows* y *MacOS*. Los principales elementos que conforman la plataforma FFmpeg son:

- **FFmpeg**: herramienta que permite el procesamiento de archivos multimedia y su conversión hacia diferentes formatos compatibles. Además, puede capturar y codificar en tiempo real desde *DirectShow*, una tarjeta de televisión u otro dispositivo compatible.
- **FFplay**: reproductor de archivos multimedia, con características simples y de carácter portable.
- **FFserver**: trabaja en conjunto con FFmpeg puesto que funciona como un servidor streaming de audio y video de todos los ficheros que FFmpeg pueda usar y procesar. Soporta HTTP (*Hypertext Transfer Protocol*).
- **FFprobe**: herramienta que obtiene e imprime en pantalla la información de los recursos multimedia previamente procesados.

Las bibliotecas que se utilizan son las siguientes:

- **Libavformat**: biblioteca que contiene los multiplexores y demultiplexores para determinar el tipo de archivo, e identificar sus códecs asociados.
- **Libavcodec**: biblioteca que contiene todos los códecs de FFmpeg, para consultar la información acerca del streaming, y para codificar/decodificar el archivo multimedia. Aunque el objetivo principal de esta biblioteca son los archivos de audio/video, también provee códecs para formatos de imágenes como GIF y PNG. Además, muchos códecs fueron desarrollados desde cero para asegurar una mayor eficiencia y la reutilización de los mismos.
- **Libavutil**: biblioteca de apoyo que contiene todas las rutinas comunes en las diferentes partes de FFmpeg.
- **Libpostproc**: biblioteca de funciones de post-procesado de video.
- **Libswscale**: biblioteca de escalado de video.





3. Resultados y discusión

En esta sección se trae a manera de ejemplo los resultados obtenidos al utilizar la plataforma antes descrita en el artículo. Se utilizaron dos muestras sin compresión en formato y4m. El uso de FFMPEG se facilita, pues esta plataforma cuenta, para la codificación de video con funciones o perfiles de uso bien definidas. En la codificación se usaron dos de estos perfiles. El perfil Placebo, al cual se corresponde con una mayor intensidad de codificación pues hace un uso extensivo de los recursos computacionales y explota a fondo todas las posibilidades óptimas que brinda el códec. Este perfil garantiza máxima eficiencia en el proceso de codificación en demerito del tiempo de duración del proceso. El perfil *medium*, el cual garantiza un balance entre un proceso de codificación eficiente e intensivo y un tiempo prudente para realizar el proceso. Es meritorio aclarar que existen varios perfiles más para ejecutar el proceso de codificación. Los perfiles escogidos evidencian una marcada diferencia entre sí en cuanto a los parámetros de codificación. La variación de parámetros entre cada uno de los perfiles no es significativa comparada con la variación entre estos dos perfiles mencionados. Este análisis pudiera quedar pendiente para trabajos futuros.

Las características de las muestras y los resultados obtenidos en la codificación utilizando H.264 y H.265 se muestran a continuación en las siguientes tablas:

Tabla 1. Parámetros de las muestras sin compresión.

Parámetros	Muestra Definición Estándar (SD)	Muestra en FullHD
Resolución (píxeles)	704x576	1920x1080
Profundidad de bits	8	8
Bitrate (kbps)	291 964	622 080
Tamaño de archivo (MB)	348	643
Formato	y4m	y4m
Relación de aspecto	4:3	16:9
Número de Imágenes	600	217

Fuente: (Elaboración Propia)



Tabla 2. Resultados obtenidos en la codificación para resolución estándar.

Parámetros	H.264 (SD) Medium	H.265 (SD) Medium	H.264 (SD) PB	H.265 (SD) PB
Bitrate (Kbps)	2420,07	854,86	2301,06	789,63
Tamaño de archivo (MB)	3,033	1,1	2,9	0,997 8
Duración de la codificación	9,96	17,85	10	515,52

Fuente: (Elaboración Propia)

Tabla 3. Resultados obtenidos en la codificación para resolución *FullHD*.

Parámetros	H.264 (FHD) Medium	H.265 (FHD) Medium	H.264 (FHD) PB	H.265 (FHD) PB
Bitrate (Kbps)	7275,94	2095,93	6425,77	2116,58
Tamaño de archivo (MB)	7,9	2,3	7,0	2,3
Duración de la codificación	8,60	19,05	8,60	328,23

Fuente: (Elaboración Propia)

Como se puede apreciar los resultados demuestran la marcada superioridad de H.265 en parámetros elementales para el manejo y procesamiento de información audiovisual. Para el caso de la resolución estándar en H.265, el bitrate se reduce 2,83 veces para el caso del perfil *medium*. En el caso del perfil placebo el bitrate se reduce 2,91 veces. Si se analiza lo que ocurre con el tamaño del archivo codificado es evidente la superioridad pues en ambos casos es aproximadamente 3 veces menor que el tamaño obtenido para H.264. Para el caso de la muestra de *FullHD* el bitrate, con el códec H.265, es 3,47 veces menor para el perfil *medium* y aproximadamente 3 veces menor para el perfil placebo. El fenómeno se repite para el caso del tamaño de los archivos donde es evidente una sustancial disminución de los tamaños con respecto a los archivos en H.264 y al archivo original sin compresión. El análisis en los tiempos de codificación es diferente pues es evidente que para codificar en H.264 requiere de menor tiempo ya que no utiliza las herramientas que si están presentes en H.265, tales como el WPP y el SAO.

4. Conclusiones

El estándar HEVC/H.265 presenta nuevas características que proponen un grado superior de complejidad en el procesamiento de la información audiovisual. A su vez, estas características posibilitan que tenga un desempeño superior a su antecesor H.264/AVC.



Para realizar un proceso de codificación de información audiovisual utilizando HEVC/H.265, la manera más eficiente de realizar esta tarea es mediante un clúster de computadoras. Ello garantiza los recursos de hardware para realizar el proceso en el menor tiempo posible. La herramienta FFMEPG es una solución para procesar materiales audiovisuales que no se ha explorado mucho. Ofrece varias herramientas y posibilidades muy útiles en el manejo de la información audiovisual. Además, es una plataforma de código abierto lo que incrementa su flexibilidad. Se demostró a partir de los experimentos realizados la superioridad del estándar HEVC/H.265 con respecto a H.264/AVC, algo que no es una novedad científica pues está ampliamente comprobado. Lo que si resulta novedoso son las potencialidades de H.265 para ser el sustituto del H.264/AVC, el cual es el códec que se utiliza actualmente en la norma DTMB China que se utiliza en Cuba para la TDT.

Trabajos Futuros:

1. Implementar un codificador HEVC/H.265 basado en la familia de codificadores HM ya que en los principales trabajos relacionados al tema se utilizan estos codificadores y se aprovecharía al máximo las potencialidades del procesamiento paralelo a nivel de nodos.
2. Realizar experimentos con materiales en UHD (4K y 8K).
3. Análisis de las métricas de calidad perceptual de video utilizando HEVC/H.265 y los diferentes perfiles que brinda FFMPEG.

5. Referencias bibliográficas

- [1] G. J. Sullivan, J. Ohm, W.-J. Han, y T. Wiegand, «Overview of the high efficiency video coding (HEVC) standard», *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, n.º 12, pp. 1649–1668, 2012.
- [2] R. H. ITU-T, «265 (04/13)», *Ser. H Audiov. Multimed. Syst. Infrastruct. Audiov. Serv. Mov. Video High Effic. Video Coding Online* [Httpwww Itu IntrecT-REC-H](http://www.itu.int/rec/T-REC-H), pp. 265-201304.
- [3] V. Sze, M. Budagavi, y G. J. Sullivan, «High efficiency video coding (HEVC)», *Integr. Circuit Syst. Algorithms Archit. Springer*, pp. 1–375, 2014.
- [4] K. Chen, Y. Duan, L. Yan, J. Sun, y Z. Guo, «Efficient SIMD optimization of HEVC encoder over X86 processors», en *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, 2012, pp. 1–4.
- [5] Y. Zhao, L. Song, X. Wang, M. Chen, y J. Wang, «Efficient realization of parallel HEVC intra encoding», en *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, 2013, pp. 1–6.
- [6] S. Zhang, X. Zhang, y Z. Gao, «Implementation and improvement of wavefront parallel processing for HEVC encoding on many-core platform», en *Multimedia*



- and Expo Workshops (ICMEW), 2014 IEEE International Conference on, 2014, pp. 1–6.
- [7] X. Wang, L. Song, M. Chen, y J. Yang, «Paralleling variable block size motion estimation of HEVC on multi-core CPU plus GPU platform», en *Image Processing (ICIP), 2013 20th IEEE International Conference on*, 2013, pp. 1836–1839.
- [8] T. K. Heng *et al.*, «A highly parallelized H.265/HEVC real-time UHD software encoder», *2014 IEEE Int. Conf. Image Process. ICIP*, pp. 1213–1217, oct. 2014.
- [9] P. U. Molina, «Técnicas de mejora de la eficiencia de codificación de video», 2016.
- [10] M. T. Pourazad, C. Doutre, M. Azimi, y P. Nasiopoulos, «HEVC: The new gold standard for video compression: How does HEVC compare with H. 264/AVC?», *IEEE Consum. Electron. Mag.*, vol. 1, n.º 3, pp. 36–46, 2012.
- [11] D. Grois, D. Marpe, A. Mulayoff, B. Itzhaky, y O. Hadar, «Performance comparison of h. 265/mpeg-hevc, vp9, and h. 264/mpeg-avc encoders», en *Picture Coding Symposium (PCS), 2013*, 2013, pp. 394–397.
- [12] G. Pastuszak y M. Trochimiuk, «Algorithm and architecture design of the motion estimation for the H. 265/HEVC 4K-UHD encoder», *J. Real-Time Image Process.*, vol. 12, n.º 2, pp. 517–529, 2016.
- [13] J. P. C. Velasco, «Caracterización de un decodificador HEVC ejecutándose en un DSP», *Univ. Politécnica Madr. Esp.*, 2014.
- [14] D. Jiménez y A. Medina, «Cluster de Alto Rendimiento», *J. Innov. Tecnol.*, p. 16, 2014.
- [15] «FFmpeg». [En línea]. Disponible en: <https://ffmpeg.org/>. [Accedido: 15-may-2017].
- [16] P. E. Martínez Martínez, «Compresión eficiente de materiales audiovisuales utilizando H. 265-HEVC», Universidad Central "Marta Abreu" de Las Villas. Facultad de Ingeniería Eléctrica. Departamento de Electrónica y Telecomunicaciones, 2016.