

Paralelización a nivel de frame de JEM, el nuevo estándar de vídeo.

Miguel Martínez-Rach,¹ Hector Migallón, Otoniel López-Granado, Vicente Galiano

Resumen— Tras el desarrollo del estándar HEVC (High Efficiency Video Coding), los grupos ITU-T Video Coding Expert Group e ISO/IEC Moving Picture Expert Group están trabajando conjuntamente en el desarrollo de una nueva tecnología de codificación de vídeo con una capacidad de compresión que supere la del actual estándar HEVC. Este actividad está guiada por la evaluación de las tecnologías de compresión propuesta por expertos en este área. Resultados preliminares muestran que el modelo del nuevo estándar JEM reduce el bit rate pero a costa de un incremento extremadamente alto de la complejidad computacional con respecto al HEVC, haciendo más necesario que nunca el uso de técnicas de optimización para rebajar las demandas computacionales en el codificador. Por ello, en este trabajo presentamos un algoritmo paralelo de procesamiento de frames para el modo de codificación All Intra, obteniendo eficiencias cercanas a la ideal usando hasta 30 procesadores.

Palabras clave— JEM, MPI, Video Coding, HEVC.

I. INTRODUCCIÓN

El estándar High Efficiency Video Coding (HEVC) [1] fue desarrollado en 2013 por el Joint Collaborative Team on Video Coding (JCT-VC) para reemplazar al estándar H.264/Advanced Video Coding (AVC) [2]. El estándar HEVC alcanza un ahorro en rate cercano al 50% respecto al estándar anterior, pero a costa de una complejidad computacional muy elevada en la etapa de codificación [3].

Conforme al informe de Cisco titulado ‘The Zettabyte Era: Trends and Analysis’ [4], en 2016 el tráfico de vídeo IP era el 73% del total de tráfico IP, pero se espera que pase al 82% para el 2021, lo que significa que cada segundo un millón de minutos de contenido de vídeo cruzará la red. El informe también predice el continuo incremento de nuevos servicios como Video-on-Demand (VoD), streaming de vídeo en directo, realidad virtual y realidad aumentada. El tráfico VoD se duplicará en el 2021 principalmente por el incremento del número consumidores y por el incremento en la resolución del vídeo (4K y 8K). También está habiendo un creciente interés por la retransmisión en streaming de eventos deportivos en directo como el fútbol o el fútbol americano, donde los equipos más grandes están instalando múltiples cámaras (más de 20) de alta resolución (4K y 5K) alrededor de los estadios para retransmitir una visión 360° del partido.

Este incremento del tráfico IP requiere una nueva generación de técnicas de codificación capaces de conseguir mayores ratios de compresión que los obtenidos con los estándares anteriores. Desde la misma aparición del HEVC, ITU-T Video Coding

Expert Group (VCEG) y ISO/IEC Moving Picture Expert Group (MPEG) han estado estudiando la necesidad potencial de la estala estandarización de nuevas tecnologías con capacidades de compresión que superen a las del HEVC. Con ese fin se ha creado un equipo de colaboración llamado Joint Video Exploration Team (JVET).

Estas mejoras en la compresión están siendo implantadas por el JVET en un modelo software para pruebas llamado Joint Exploration Test Model (JEM) [5]. Los resultados experimentales en la configuración All Intra (AI) [6] muestran que el nuevo modelo (JEM 3.0) alcanza una reducción de rate del 18%, pero a costa de una complejidad computacional muy alta (60x) con respecto al HEVC.

En este artículo presentamos un codificador paralelo del modelo JEM para el modo AI, especialmente adecuado para plataformas de memoria distribuida. Realizamos varios test para demostrar su buen comportamiento en términos de eficiencia y escalabilidad.

El resto del artículo está organizado como sigue: En la sección II, se presenta una breve descripción de la herramientas de codificación introducidas en el nuevo estándar de vídeo (JEM). En la sección III se muestra el algoritmo paralelo propuesto para el modo de codificación AI, mientras que en la sección IV presentamos algunos resultados de los test del algoritmo propuesto. Finalmente en la sección V se presentan las conclusiones.

II. DESCRIPCIÓN DEL ALGORITMO DEL JOINT EXPLORATION TEST MODEL (JEM)

Esta sección describe algunas de las nuevas características de codificación que están bajo estudio por el JVET, y que se han incluido en el software de referencia del JEM como potenciales mejoras tecnológicas que superan las capacidades del HEVC. JEM se ha desarrollado sobre el modelo de test (HM 16.6) del HEVC [7]. Aunque el flujo de codificación y decodificación del HEVC se ha mantenido intacto, el JEM incluye nuevas herramientas en los módulos más importantes, entre los que se encuentran los módulos de predicción intra e inter [5].

A. Estructura de bloques

De manera similar a como se hace en HEVC, cada cuadro de vídeo o frame, puede dividirse en varios bloques llamados slices y cada slice puede dividirse en una secuencia de árboles de codificación llamados Coding Tree Units (CTUs). Un CTU puede ser demasiado grande como para decidir si se realiza una predicción intra o inter (utilizando el mismo frame

¹Departamento de Física y Arquitectura de Computadores - Universidad Miguel Hernández, e-mail: mmrach@umh.es

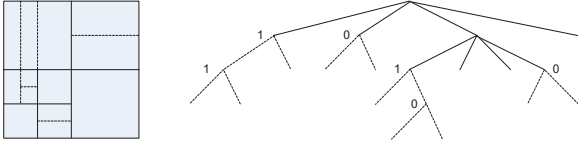


Fig. 1. Ejemplo de estructura QTBT.

o respecto distintos frames), por lo que puede ser dividido recursivamente en 4 sub-regiones con una estructura de quad-tree.

Sin embargo, JEM incorpora una estructura de quad-tree más binary-tree (QTBT) para codificar los bloques. Esta estructura aporta mayor flexibilidad para que las particiones se ajusten mejor a las características locales de la secuencia de video, de tal forma que la organización en Coding Units (CU), Prediction Units (PU) y Transform Units (TU) del HEVC ya no son necesarias. Con QTBT los CUs pueden tener tanto forma cuadrada como rectangular, como los PUs en el HEVC. Cada CTU, que puede llegar hasta 256×256 pixels, primero se divide en CUs cuadrados en una estructura quad-tree como en HEVC. Entonces, los nodos hoja pueden ser divididos adicionalmente por una estructura binary-tree. En este binary-tree, cada CU puede ser dividido en dos mitades horizontales o verticales. Un ejemplo de una estructura CTU en JEM se muestra en la figura 1, donde el quad-tree alcanza dos niveles (línea continua) en una de sus ramas y de sus hojas comienzan los binary-trees (líneas discontinuas).

Los siguientes parámetros definen un QTBT:

- CTUSize: Tamaños del nodo raíz de un quad-tree, mismo concepto que en HEVC.
- MinQTSize: Mínimo tamaño permitido de un nodo hoja del quad-tree.
- MaxBTSize: Máximo tamaño permitido del nodo raíz de un binary-tree.
- MaxBTDepth: Profundidad máxima para un binary-tree.
- MinBTSize: Mínimo tamaño permitido de un nodo hoja del binary-tree.

B. Mejoras en la predicción Intra e Inter

En la predicción Intra, con el fin de capturar más direcciones en los bordes presentes en imágenes naturales, los modos direccionales (intra-modes) se amplían de los 33 definidos en el HEVC hasta 65. Los modos Planar y DC permanecen igual. Todos los modos direccionales se aplican a todos los tamaños de bloque en la predicción intra, tanto para luminancia como para crominancia. Para ajustarse al incremento de modos direccionales, el modo de codificación intra usará 6 modos más probables (Most Probable Modes) para seleccionar el modo final.

Se consideran dos métodos para la predicción de los vectores de movimiento a nivel sub-CU en el codificador mediante la división de un CU en sub-CUs y derivando la información de movimiento a todos los sub-CUs.

El método avanzado de predicción de vectores de movimiento (Advanced Temporal Motion Vector Prediction), permite a cada CU extraer múltiples conjuntos de información de movimiento desde múltiples bloques más pequeños que el CU procesado en el frame de referencia correspondiente. En el método de predicción espacio-temporal de los vectores de movimiento (Spatial-Temporal Motion Vector Prediction), los vectores de movimiento de los sub-CUs se obtienen recursivamente usando la predicción temporal y una vecindad espacial de vectores de movimiento. En JEM, la precisión de los vectores de movimiento y la fusión de candidatos se incrementa hasta un dieciseisavo de pixel, mientras que para el HEVC era de un cuarto de pixel.

C. Otras mejoras

En JEM, además de la transformada discreta del coseno (DCT-II) y la transformada discreta del Seno (DST-VII) para bloques de 4×4 usadas hasta ahora en la etapa de transformación del HEVC, se añade una transformada adaptativa múltiple (Adaptive Multiple Transform - AMT) que se ha elegido para codificar el residuo intra e inter, que utiliza diferentes familias DCT y DST que las usadas en el HEVC.

En JEM se aplica un filtro adaptativo a nivel de bloque (Adaptive Loop Filter - ALF). Para la componente de luminancia, se selecciona uno de entre 25 filtros en función de la dirección y la cantidad de textura local para bloques de 2×2 . Este filtro adaptativo intenta reducir las distorsiones visibles como ringing y blurring reduciendo el error absoluto medio entre la imagen original y la reconstruida.

En HEVC, el codificador entrópico utilizado es el Context-based Adaptive Binary Arithmetic Coding (CABAC). JEM utiliza una versión mejorada de CABAC con un modelo diferente de de contextos para los coeficientes transformados, una estimación de probabilidad multi hipótesis con una inicialización de los modelos adaptativa.

III. ALGORITMO PARALELO A NIVEL DE FRAME

La propuesta paralela a nivel de frame, específicamente diseñada para la codificación AI, se muestra en la figura 2.

Para este artículo, el algoritmo se ha probado en un sistema de memoria heterogénea, gestionado por Message Passing Interface (MPI) y compuesto por N nodos de cálculo en una arquitectura de memoria distribuida. Cada nodo tiene una arquitectura de memoria compartida, donde en cada uno de los N nodos se ejecutan R_i procesos MPI ($P_{i,j}; (i = 1 \dots N, j = 1 \dots R_i)$).

Cada uno de los $P_{i,j}$ solicita al proceso coordinador (P_{coord}) un frame para codificarlo. En el modo de codificación AI, todos los frames son codificados como I frames, es decir, sin usar información de frames previamente codificados. El proceso P_{coord} se encarga de servir frames a los procesos solicitantes y almacenar estadísticas y los datos codifi-

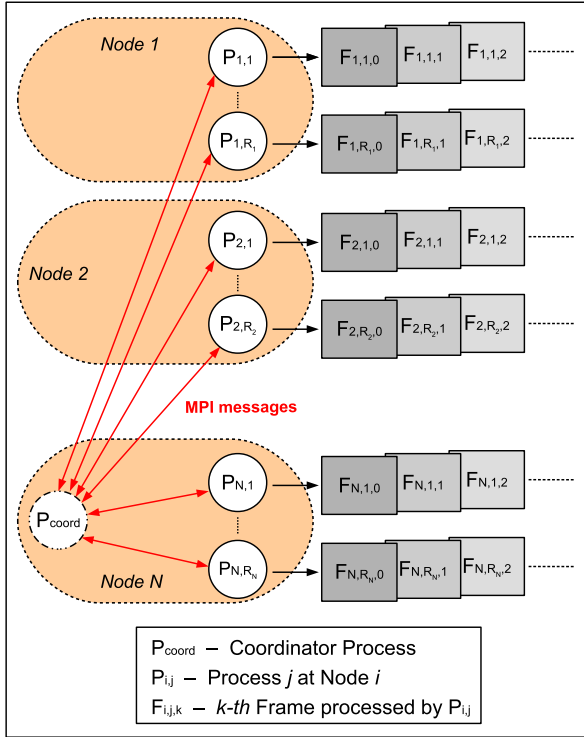


Fig. 2. Algoritmo paralelo a nivel de frame

cados. Cuando un proceso $P_{i,j}$ termina la codificación del primer frame recibido ($F_{i,j,0}$), envía el bitstream codificado al coordinador P_{coord} , recibiendo de éste un nuevo frame a codificar, hasta que todos los frames se hayan codificado. Así, cuando un proceso de cálculo $P_{i,j}$ termina de procesar un frame y queda ocioso se le asigna inmediatamente un nuevo frame sin esperar a que el resto de procesos terminen su trabajo. Este hecho proporciona una buena distribución de la carga y excelentes valores de speed-up.

Como el algoritmo es asíncrono, el orden en el que cada frame es codificado no es necesariamente el orden de renderizado, así que el proceso coordinador tiene que formar el bitstream codificado en el orden apropiado. El proceso coordinador está asignado a uno de los procesadores que corren procesos de cálculo debido a que la carga computacional de sus tareas es despreciable.

El algoritmo paralelo propuesto genera un bitstream que es idéntico al producido por el algoritmo secuencial, por lo que no hay degradación en términos de rate/distortion (R/D).

IV. RESULTADOS EXPERIMENTALES

El algoritmo paralelo propuesto se ha probado en una plataforma paralela compuesta por 10 nodos HP Proliant SL390 G7, donde cada nodo está equipado con dos procesadores Intel Xeon X5660. Cada X5660 incluye seis núcleos de procesamiento a 2.8 GHz y se ha utilizado una red de comunicaciones QDR Infiniband. El sistema paralelo se ha gestionado con MPI v2.2 [8]. El compilador utilizado ha sido *g++ v.4.1.2*. El software de referencia para el codificador HEVC utilizado es el HM 16.3 [9] y el JEM 7.0 para el codificador

JEM.

Las secuencias de video de test utilizadas en nuestros experimentos son People on Street (2560x1600), Park Scene (1920x1080), y Four People (1280x720). Presentamos resultados para el modo de codificación All Intra (AI), codificando 150 frames para las secuencias People on Street y Four People y 240 frames para la secuencia Park Scene, utilizando diferentes parámetros de cuantización (QPs) (22, 27, 32, 37). La codificación se ha realizado usando la configuración Main profile a 8-bits de profundidad.

Primero presentamos una comparación entre HEVC y JEM en términos de Rate/Distortion y tiempo de codificación utilizando el modo de codificación AI. En la Tabla I mostramos el índice Bjøntegaard delta rate (BD-rate) que representa el porcentaje de variación de bit rate entre dos secuencias para la misma calidad objetiva [10]. En este índice, un valor negativo implica que la propuesta mejora la eficiencia, es decir, reduce el rate, respecto al codificador de referencia. Como puede verse, el codificador JEM supera al codificador HEVC. La ganancia máxima (18.88%) se ha obtenido para la secuencia de mayor resolución.

En la Tabla I mostramos el incremento medio del tiempo de codificación requerido por el codificador JEM respecto al HEVC. Como se puede apreciar, el codificador JEM requiere, en promedio, 41 veces el tiempo total requerido por el HEVC para codificar las secuencias de test. El enorme incremento en la complejidad del codificador JEM se ve en el hecho de que para codificar 150 frames de la secuencia People on Street (2560x1600) con un valor de QP de 22, el codificador JEM requiere 95.5 horas mientras que el codificador HEVC requiere sólo 1.71horas.

TABLA I
COMPARACIÓN DE LA COMPLEJIDAD DE JEM RESPECTO
HEVC EN TÉRMINOS DE R/D

Resolución	BD-Rate (%)	Δ tiempo promedio
2560x1600	-18.88	44.53x
1920x1080	-13.31	44.99x
1280x720	-17.60	34.58x

Al ver los resultados de la Tabla I son necesarios grandes esfuerzos para acelerar el tiempo total de codificación del nuevo estándar de codificación JEM. Este trabajo presenta una versión paralela de la versión del codificador JEM específicamente adecuada para cine digital (digital cinema) donde todos los frames son codificados en modo intra. Como se especifica en la sección III, el algoritmo paralelo propuesto no introduce ninguna degradación en la calidad del video reconstruido con respecto a la versión secuencial.

En la Figura 3 podemos observar la aceleración (speed-up) obtenida por el algoritmo propuesto para la secuencia de video PeopleOnStreet. Como se puede ver, se obtiene un speed-up de hasta 27.69x

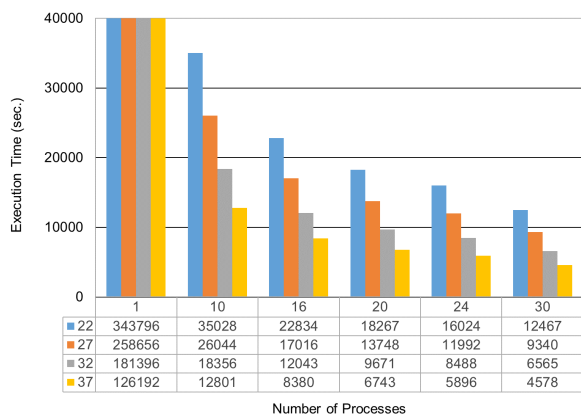


Fig. 3. Speed-up obtenido por el algoritmo paralelo a nivel de frames para la secuencia PeopleOnStreet en el codificador JEM.

cuando se usan 30 procesos para comprimir la secuencia de video. Además, el algoritmo paralelo muestra una gran escalabilidad, obteniendo eficiencias de hasta un 99.77%. Resultados similares se obtienen para todas las secuencias de video procesadas.

V. CONCLUSIONES

En este artículo hemos presentado una versión paralela del software de referencia JEM especialmente indicada para el modo de codificación AI (all intra). Primero se ha presentado una comparación entre el estándar de video actual, HEVC y el futuro estándar JEM en términos de R/D y tiempo de codificación. JEM supera a HEVC en R/D, obteniendo reducciones de rate de hasta un 18.8% en el modo de codificación AI para las secuencias procesadas. Esta reducción en bit rate se obtiene a costa de un enorme incremento en el coste computacional, requiriendo JEM 41 veces, de media, el tiempo usado por el HEVC para comprimir una secuencia de video.

Tras la comparación entre HEVC y JEM, hemos presentado varios resultados del algoritmo paralelo propuesto para JEM. Los resultados muestran que el algoritmo propuesto obtiene buenas aceleraciones de hasta 27.69x al usar 30 procesos y sin degradación en R/D. El algoritmo propuesto tiene una alta escalabilidad, obteniendo eficiencias de hasta un 99.77%.

Como trabajo futuro, proponemos el desarrollo de nuevos algoritmos paralelos para los diferentes modos de codificación disponibles en JEM, como Random Access o Low-delay.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad de España y la Comisión Europea (fondos FEDER) bajo el Proyecto TIN2015-66972-C5-4-R.

REFERENCIAS

[1] G. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1648–1667, December 2012.

[2] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," *ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC) version 16*, 2012, 2012.

[3] J. Ohm, G. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards - including high efficiency video coding (HEVC)," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1669–1684, 2012.

[4] Cisco, "Cisco visual networking index: forecast and methodology, 2016-2021," Tech. Rep., Accessed 31 Jul 2017, <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>.

[5] J. Chen, E. Alshina, G. J. Sullivan, J. R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 3," *Technical Report JVET-C1001*, 2016.

[6] M. Karczewicz and E. Alshina, "JVET AHG report: tool evaluation (AHG1)," Technical Report JVET-D0001, Tech. Rep., 2016.

[7] ISO/IEC, ITU-T (2016) HEVC, "test model (HM) reference software. <https://hevc.hhi.fraunhofer.de/>," Tech. Rep.

[8] MPI Forum, "MPI: A Message-Passing Interface Standard. Version 2.2," September 4th 2009, available at: <http://www.mpi-forum.org> (Dec. 2009).

[9] HEVC Reference Software, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.3/.

[10] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," Technical Report VCEG-M33, ITU-T Video Coding Experts Group (VCEG), Tech. Rep., 2001.