UNIVERSIDAD POLITECNICA DE VALENCIA

DEPARTAMENTO DE COMUNICACIONES

Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones

Evaluación del nuevo codificador VVC/H.266 para sistemas de streaming

Autor: Antonio Diyanov Nikolov

Director 1: Juan Carlos Guerri Cebollada

Director 2: Pau Arce Vila

Fecha de comienzo: 30/03/2021

Lugar de trabajo: Grupo de Comunicaciones Multimedia del iTEAM

Objetivos — El objetivo del presente trabajo es la evaluación de las prestaciones del nuevo codificador VVC/H.266 en los sistemas de streaming utilizando la implementación del codificador VVC de Fraunhofer (*Versatile Vídeo Encoder*; VVenC). Para ello se ha realizado una comparativa de las métricas objetivas más importantes obtenidas codificando con VVenC y comparándolas con las mismas métricas que ofrecen los codificadores de generaciones anteriores, HEVC/H265 y AVC/H.264 codificando con *ffmpeg*.

Motivaciones —Debido a que las transmisiones de vídeo consumen gran parte del ancho de banda disponible en Internet, especialmente si son de alta calidad, cada vez más surge la necesidad de optimizar el ancho de banda manteniendo una calidad visual alta. Además, con la aparición de nuevos codificadores resulta interesante evaluar la mejora en calidad visual o *bitrate* del vídeo y el coste en términos de procesamiento y tiempo empleado que estos ofrecen con respecto a los codificadores de generaciones anteriores. En concreto, el estudio realizado forma parte de una tarea del proyecto "DESARROLLO DE UN SISTEMA DE STREAMING MULTIMEDIA AVANZADO PARA UNA NUEVA PLATAFORMA A BORDO BASADA EN REDES 5G Y WIFI 6 PARA MEDIOS DE TRANSPORTE TERRESTRE" realizado por el grupo de Comunicaciones Multimedia del iTEAM.

Metodología — En cuanto la metodología se han elegido un conjunto de vídeos de prueba con diferente complejidad espacial y temporal con el objetivo de cubrir una amplia gama de escenarios reales. Se han elegido cuatro tasas de bit o *bitrates* comprobando que las calidades obtenides en cada *bitrate* son diferentes. A continuación, se ha utilizado la herramienta de *ffmpeg* para codificar los vídeos en H.264 y H.265 y la implementación del codificador VVC de Fraunhofer (*Versatile Video Encoder*; VVenC) para codificar los vídeos en H.266. En cuanto a la comparativa se han analizado parámetros como PSNR, SSIM, VMAF y tiempo de codificación y otras métricas, para poder obtener los resultados y conclusiones.

Desarrollos teóricos realizados — El desarrollo teórico comienza describiendo las principales características de un codificador de vídeo híbrido, siendo este el modelo genérico de codificador utilizado en la mayoría de las implementaciones de hoy en día. A continuación, se realiza una descripción de los diferentes codificadores de vídeo clasificados por familias: VP9, AV1, H.264, SVC, H.265, H.266 y LCEVC. También se realiza una descripción amena y estructurada del funcionamiento del codificador VVC/H.266 haciendo hincapié en las mejoras que aporta respecto a los codificadores anteriores. En cuanto a la metodología de la evaluación de prestaciones se comentan las características de los vídeos elegidos, así como una descripción teórica de las métricas utilizadas en la comparativa.

Resultados — En cuanto a los resultados se han obtenido las gráficas de las métricas de PSNR, SSIM y VMAF, además de resultados de interés como el tiempo de codificación, tamaño resultante de los ficheros, entre otros. Con todo ello se podido comprobar que el nuevo codificado VVC/H.266 cumple con todas las especificaciones requeridas para sustituir a los codificadores anteriores en cuanto a calidad visual puesto que para el mismo *bitrate* ofrece una calidad visual notablemente superior. Como contrapartida, cabe señalar que hoy en día los tiempos de codificación de VVC siguen siendo muy altos comparados con los codificadores existentes.

Líneas futuras — Respecto al trabajo futuro, se propone la realización del mismo estudio cuando exista soporte del codificador VVC/H.266 en la herramienta de *ffmpeg*. De esta manera se podrá evaluar las prestaciones del nuevo codificador con respecto a AVC y HEVC bajo la misma herramienta y las mismas condiciones.

Abstract — The main goal of this work is the evaluation of the performance of the new VVC / H.266 encoder in streaming vídeo systems. To achieve that, a comparison of the most important objective metrics obtained by encoding with VVC/H.266 has been made and compared to the same metrics offered by the encoders of previous generations, HEVC/H265 and AVC/H.264. Regarding the methodology used, a set of test vídeos with different spatial and temporal complexity has been chosen in order to cover a wide range of real scenarios. Four bitrates have been chosen, checking that the qualities obtained in each bitrate are different. The *ffmpeg* tool was then used to encode the vídeos in H.264 and H.265 and the implementation of the Fraunhofer VVC encoder (Versatile Vídeo Encoder; VVenC) to encode the vídeos in H.266. Regarding the comparison, parameters such as PSNR, SSIM, VMAF and coding time have been analyzed in order to obtain the results and conclusions. The theoretical development begins by describing the main characteristics of a hybrid vídeo encoder, this being the generic encoder model used in most implementations today. Next, there is a description of the different vídeo encoders classified by families: VP9, AV1, H.264, SVC, H.265, H.266 and LCEVC. A pleasant and structured description of the operation of the VVC / H.266 encoder is also made, emphasizing the improvements it brings with respect to previous encoders. Regarding the performance evaluation methodology, the characteristics of the chosen vídeos are discussed, as well as a theoretical description of the metrics used in the comparison. Regarding the results, the graphs of the PSNR, SSIM and VMAF metrics have been obtained, as well as results of interest such as encoding time, file size, among others. With all this, it was possible to verify that the new VVC/H.266 encoder meets all the specifications required to replace the previous encoders in terms of visual quality since for the same bitrate it offers a significantly higher visual quality. As a drawback, it should be noted that VVC encoding times are still very high today compared to existing encoders.

Autor: Antonio Diyanov Nikolov, email: andini@teleco.upv.es
Director 1: Juan Carlos Guerri Cebollada, email: jcguerri@dcom.upv.es
Director 2: Pau Arce Vila, email: paarvi@iteam.upv.es
Fecha de entrega: 12-09-21

Índice

1. Introducción

Debido al crecimiento exponencial de las aplicaciones de vídeo y a la demanda de contenido de alta calidad y resolución, en los últimos años se ha incrementado considerablemente el tráfico de vídeo en Internet [1]. El objetivo de los nuevos codificadores es la mejora continua de la Calidad de Experiencia (QoE) al tiempo que se reduce la carga de la red de los *Network Service Provider* (NSP). Además, la implementación de codificadores de vídeo más eficientes será el habilitador que permita la lenta pero imparable incorporación de servicios más exigentes como los vídeos en alta definición (HD), 4K (UHD), 360 grados, realidad aumentada (AR), realidad virtual (VR), etc…

El estándar de vídeo H.264 *Advanced Vídeo Coding* (AVC/H.264) se utiliza hoy en día en la mayoría de las aplicaciones de transmisión y almacenamiento de vídeo. A pesar de ello, este codificador no está a la altura de las demandas del mercado actual. Por ejemplo, un vídeo en resolución 1080p genera 4 veces más tráfico de datos que en una resolución de 480p. Un vídeo de YouTube de 360 grados genera de 4 a 5 veces el tráfico que generaría el mismo vídeo en resolución estándar. La máxima resolución que soporta H.264 es de 4K mientras que los nuevos sistemas de realidad virtual prometen la visualización de vídeos de 6K y 8K. Por tanto, existe la necesidad de desarrollar estándares de codificación de vídeo que permitan una alta compresión con una baja complejidad. El desarrollo de nuevos codificadores como *High Efficiency Vídeo Coding* (HEVC/H.265) o *Versatile Vídeo Coding* (VVC/H.266) se ofrecen como una solución al problema puesto que permiten duplicar la eficiencia de codificación, además de suportar vídeos de resoluciones 8K en el caso de H.265 y hasta 16K en H.266.

2. Conceptos sobre los codificadores de vídeo

Un codificador de vídeo o *encoder* es un equipo o aplicación cuyo propósito es comprimir un vídeo para su posterior transporte en una red. Se puede utilizar tanto para vídeo en directo como vídeo bajo demanda [2]. El algoritmo de compresión utilizado es lo que se denomina códec. En la recepción del vídeo se utiliza un decodificador cuyo propósito es recibir el flujo de bits codificados y decodificarlos para su posterior reproducción.

En cuanto a la compresión, esta puede ser sin pérdidas o con pérdidas. La compresión sin pérdidas consiste en comprimir el archivo sin perder calidad en el mismo mientras que en la compresión con pérdidas se reduce la calidad a cambio de reducir el tamaño del archivo. Respecto al método de codificación este puede ser de *bitrate* constante o *Constant Bit Rate* (CBR) en el que el decodificador recibe la misma cantidad de bits por segundo a lo largo del tiempo o *birate* variable o *Variable Bit Rate* (VBR) en el que el decodificador recibe una cantidad de bits variable en función de la escena. Algunos de los parámetros a configurar en el codificador son el *bitrate* o cantidad de bits por segundo, resolución, *frames* por segundo, etc…

2.1. Digitalización

Cualquier vídeo almacenado está formado por una secuencia de fotogramas o *frames* muestreados en espacio y en tiempo [3]. En el dominio espacial cada *frame* está formado por píxeles. El número de píxeles es el número de muestras de cada *frame* siendo estos representados en formato anchura por altura, por ejemplo, 1280x720 o 720p. En el dominio temporal el vídeo está formado por un número de *frames* por unidad de tiempo siendo valores típicos: 24 fps, 30fps o 60 fps. Los codificadores de vídeo aprovechan tanto el dominio espacial como el dominio temporal para comprimir el vídeo, siendo este último el que más aporta a la compresión.

2.2. Representación de imagen

Existen diferentes formatos para representar las imágenes de un vídeo. Una de ellas es el formato RGB en el cuál cada píxel se codifica con 8, 10 o 12 bits por color (24, 30 o 36 bits en total). Para realizar la compresión, el formato RGB se convierte a un formato de luminancia y crominancias puesto que el ojo humano es más sensible al brillo (luminancia) que al color (crominancia). La conversión de luminancia a RGB y viceversa se realiza mediante la multiplicación por una matriz de conversión.

Otro concepto relacionado directamente con la representación de una imagen es el de los formatos de muestreo. Los formatos de muestreo pueden ser de diferentes tipos. En el formato 4:4:4 por cada cuatro píxeles en horizontal de luminancia se cogen cuatro de crominancia de azul y cuatro de crominancia de rojo. En el 4:2:2 por cada cuatro de luminancia se cogen dos de crominancia de azul y dos de crominancia en rojo. Existen otros formatos de muestreo como el 4:1:1 o 4:2:0 donde el razonamiento es el mismo. Se trata de eliminar información de color que el ojo humano no es capaz de percibir.

2.3. Bloques del codificador

En la Figura 1 podemos observar la estructura básica de un codificador de vídeo. Este está formado por tres bloques: el modelo temporal, el modelo espacial y la codificación entrópica.

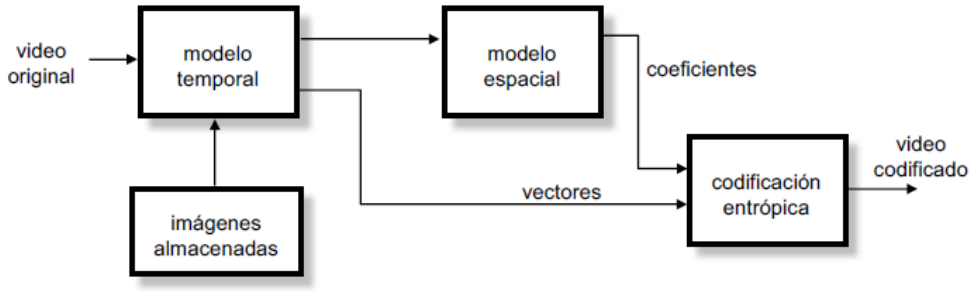

Figura 1. Estructura básica codificador de vídeo.

En los siguientes subapartados se procede a describir brevemente el funcionamiento de cada uno de estos bloques, comenzando por la codificación entrópica, siguiendo con el modelo espacial y finalmente el modelo temporal.

2.3.1. Codificación entrópica

Se trata de un sistema general utilizado en imágenes para eliminar redundancia estadística de los datos sin pérdida de información. Algunos de los tipos de codificación entrópica más conocidos son: *Run-Level Encoding* (RLE), *Variable-Length Coding* (VLC) y Context *Adaptative Binary Arithmetic Coding* (CABAC).

- **RLE (Run-Level Encoding)**: Es una forma de compresión de los datos en los que se pueden comprimir largas secuencias de valores consecutivos. La representación de símbolos codificados viene dada por la terna de valores (**last, run, level**) donde **last** vale 0 o 1 en función de si el coeficiente representado es el último, **run** es el número de ceros delante del coeficiente representado y **level** es el valor del coeficiente. Por ejemplo: para la serie de datos: [16, 0, 0, -3, 5, 6, 0, 0, 0, 0, -7] los datos codificados serían: [(0,0,16), (0,2,-3), (0,0,5), (0,0,6), (1,4,-7)]. Podemos observar que para una alta compresión en la codificación entrópica interesa que los coeficientes procedentes del modelo temporal tengan muchos ceros consecutivos.
- **VLC (Variable-Length Coding):** Consiste en codificar una serie de símbolos o coeficientes en función de la probabilidad de aparición de estos. Para ello los símbolos más frecuentes se representan con menor número de bits y símbolos menos frecuentes con mayor número de bits. Uno de los métodos más utilizados es la codificación Huffman. Por ejemplo: para los símbolos [-2, -1, 0, 1, 2] se calcula su probabilidad de aparición [0,1 0,2 0,4 0,2 0,1] y se asignan palabras código [000 010 1 011 001] de manera que se puede observar que para el símbolo 0, que tiene mayor probabilidad de aparición se utiliza un único bit mientras que para los demás se utilizan tres. La ventaja de utilizar palabras código generadas por el método de Huffman es que ninguna palabra código es prefijo de otra con lo que se pueden decodificar todas correctamente conforme se van recib iendo las palabras código. Los estándares de compresión utilizan tablas genéricas en los que los símbolos definidos coinciden con las ternas definidas proporcionadas por la codificación RLE.
- **CABAC (Context *Adaptative Binary Arithmetic Coding*):** Modelo de codificación entrópica que se basa en la codificación aritmética [4]. La codificación aritmética consiste en mapear los símbolos obtenidos del modelo temporal (coeficientes o vectores de movimiento) a palabras código de bits variables. Existen tres procesos en CABAC: binarización, modelo de contexto y codificación aritmética. La binarización consiste en transformar los símbolos en palabras código o *bins*. Se elige un modelo de contexto mediante el cual se calcula la probabilidad de aparición de cada *bin* en función de las estadísticas de los símbolos ya transmitidos. El codificador aritmético codifica cada *bin* en función del modelo de contexto seleccionado. Por último, se realiza una actualización de las probabilidades de aparición basado en el valor anterior del *bin* elegido.

2.3.2. Modelo espacial

El modelo espacial es el encargado de generar los coeficientes que recibe la codificación entrópica. Se ha visto en el apartado anterior que para que la codificación entrópica sea lo más eficiente posible es necesario que el modelo espacial genere coeficientes en los que existan muchos ceros consecutivos. El modelo espacial aprovecha la alta correlación entre píxeles adyacentes en las zonas de la imagen en la que los valores de luminancia y crominancia son parecidos. Además, se aprovecha las zonas de la imagen en la que hay mucho detalle para comprimir la misma. Para ello el modelo espacial se divide en tres fases: Transformación, Cuantificación y Reordenamiento.

- **Transformación**: En la gran mayoría de codificadores se utiliza la transformada discreta del coseno DCT o una variante de esta. Consiste en realizar una transformación multiplicando la matriz de los valores de las luminancias o crominancias por una matriz de transformación. Con ello, se obtienen una serie de coeficientes mediante los cuales se puede regenerar la imagen original. La matriz o bloques utilizados en la transformación pueden ser de tamaño 4x4, 8x8, 16x16, etc...
- **Cuantificación**: Es un proceso con pérdidas que consiste en establecer una cuantificación de los coeficientes. Para ello se elige un valor de QP o *quantizer* por el cual se dividen todos los coeficientes de la imagen. A menor valor de QP la compresión será muy baja mientras que a mayor valor de QP la compresión será muy alta. En función de la cantidad de coeficientes seleccionados se obtendrá una compresión menor o mayor. Para pocos coeficientes la compresión será muy alta mientras que para muchos coeficientes la compresión será muy baja.
- **Reordenamiento**: Consiste en generar un vector de coeficientes en los que existen muchos ceros consecutivos a partir de la matriz de coeficientes generada con la transformación y la aplicación del QP. Uno de los reordenamientos más eficientes para la mayoría de los codificadores es en zigzag.

2.3.3. Modelo temporal

Es en el modelo temporal donde se produce la mayor compresión del vídeo puesto que se explota que el vídeo está compuesto por imágenes consecutivas que se parecen entre sí. El modelo temporal se encarga de proporcionar al modelo espacial las diferencias de imágenes consecutivas y no la imagen completa. Estas diferencias se denominan imagen residual. La imagen residual se obtiene restando al *frame* N el *frame* N-1 reconstruido siendo la estimación y compensación de movimiento el mecanismo que utilizan los codificadores para poder disminuir las diferencias entre dos imágenes consecutivas.

- **Estimación de movimiento**: Para la estimación de movimiento se divide cada *frame* en macrobloques (bloques de 4x4, 8,8, 16x16 píxeles) y se compara el bloque del *frame* N con todos los posibles bloques del *frame* N-1. Para ello se utiliza un algoritmo de búsqueda y comparación de energía residual. Se escogerá el macrobloque del *frame* anterior que tenga una energía residual más pequeña. Una vez encontrado dicho bloque se calcula el vector de movimiento como el offset entre la posición del macrobloque del *frame* actual respecto al macrobloque del *frame* anterior. Para incrementar la probabilidad de encontrar un macrobloque parecido al del *frame* actual además de buscar en el *frame* anterior los codificadores realizan la búsqueda en *frames* anteriores y posteriores. Los vectores de movimiento se envían directamente a la codificación entrópica, sin pasar por el modelo espacial.
- **Compensación de movimiento:** Consiste en desplazar los macrobloques del *frame* N-1 en la posición en la que se encuentran esos mismos bloques en el *frame* N. El objetivo es obtener una imagen residual que contiene menos información (zonas más homogéneas) que la que tendría sin utilizar compensación del movimiento.

En cuanto a la búsqueda de macrobloques en *frames* anteriores o posteriores existen diferentes aproximaciones o algoritmos. Además, la unidad básica de búsqueda de macrobloques puede ser mitad de un píxel, ¼ de píxel, etc, y de esta manera el decodificador tiene más granularidad a la hora de encontrar el macrobloque con energía residual mínima.

2.3.4. Estructura general codificador de vídeo híbrido

En la Figura 2 podemos ver en más detalle el diagrama de bloques de un codificador de vídeo híbrido.

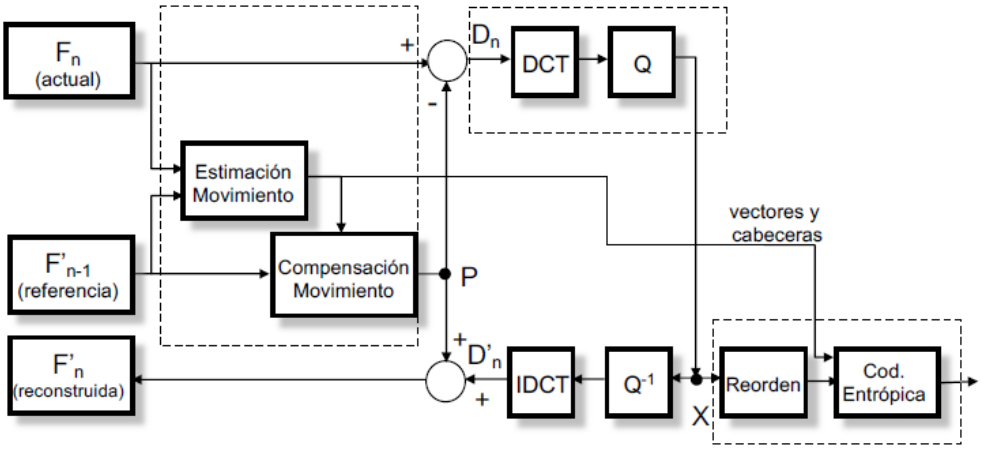

Figura 2. Diagrama de bloques de un codificador de vídeo híbrido.

Se parte del vídeo en formato de luminancias y crominancias y el codificador comienza a analizarlo *frame* a *frame*. El primer *frame* no tiene un *frame* anterior con lo que no podrá obtener un macrobloque parecido y por tanto pasa directamente al modelo espacial, donde se le aplica la transformación, cuantificación, reordenación y codificación entrópica. Este tipo de *frame* se

denomina de tipo Intra (I) y es el tipo que menos se comprime. Con los siguientes *frames* se aplica la estimación y compensación de movimiento puesto que existen *frames* anteriores con los que comparar. Una vez obtenida la imagen de referencia N-1 se le resta a la imagen actual y se aplica la estimación y compensación de movimiento, obteniendo la imagen residual. A estos *frames* se les denomina de tipo P o B. A continuación, esta imagen pasa por el modelo espacial donde se le aplica transformación, cuantificación, reordenación y codificación entrópica. La cantidad de *frames* entre Intra a Intra se denomina *Group of Pictures* (GOP). Cabe destacar también que las predicciones de la estimación y compensación de movimiento son *inter-frame*, es decir de una imagen a otras anteriores o posteriores, aunque también se pueden realizar predicciones *intra-frame* que consisten en comparar macrobloques de una posición con macrobloques de otra posición en el mismo *frame*.

2.4. Familias de codificadores

En el siguiente apartado se procede a desarrollar una breve descripción de los diferentes codificadores ordenados por familias. En concreto se han seleccionado en los codificadores: H.264/AVC, H.264/SVC, H.265/HEVC, VP9, AV1, LCEVC y H.266/VVC por ser estos los más conocidos e implementados.

- **H.264/AVC:** Codificador conocido también como MPEG-4 parte 10 o *Advanced Vídeo Coding* (AVC) sustituyó en 2003 a los estándares anteriores MPEG-2 y H.263 y MPEG-4 parte 2. Incluye funcionalidades como el uso de transformadas DCT, *quantizer* (QP), estimación y compensación de movimiento con predicciones inter e *intra-frame* y codificación entrópica [5]. Algunas de las mejoras que aporta respecto a codificadores anteriores es la predicción de tramas de tipo Intra, DCT de enteros de tamaño 4x4, uso de múltiples *frames* de referencia, macrobloques de tamaño variable, precisión de un cuarto de píxel para la compensación de movimiento y el uso de un filtro de *deblocking* con el objetivo de suavizar los bordes de los macrobloques. Todas estas mejoras hacen que el codificador aporte una mejora del 50% en la reducción del *bitrate* para un vídeo de la misma calidad. La resolución suportada por H.264 llega a 4K (4096x2160) con 60 fps.
- **H.264/SVC:** También conocido como codificador de vídeo escalable (SCV) fue creado como extensión del codificador H.264. Consiste en dividir el flujo de vídeo en una capa base y deferentes capas de mejora [1]. La capa base proporciona la información esencial mientras que las capas de mejora añaden los detalles necesarios a la capa base. Estas capas de mejora hacen que la calidad del vídeo aumente. Existen tres tipos de escalabilidad: temporal, espacial y de calidad. La escalabilidad espacial se refiere a tener una capa base de baja resolución y capas de mejora que hacen que la resolución aumente. La escalabilidad temporal se refiere a aumentar el número de *frames* por segundo con lo que aumenta la calidad. Finalmente, la escalabilidad de calidad consiste en que las capas de mejora aumentan la calidad (se añaden más coeficientes) de la capa base.

- **H.265/HEVC**: Desarrollado en 2013 para su primera versión por miembros del VCEG de la ITU-T y del MPEG de la ISO/IEC emplea la misma estructura de predicción *inter* e *intra-frame* que H.264. La primera imagen utiliza únicamente predicción *intra-frame* mientras que todas las demás utilizan predicción *intra* e *inter-frame* [6]. El concepto de macrobloque utilizado en H.264 se convierte en un *Coding Tree Unit* (CTU). Cada CTU puede ser de tamaño 16x16, 32x32 o 64x64. A su vez los CTU se dividen en un árbol cuaternario *Quad-Tree* (QT) que permite particiones más pequeñas llegando a *Coding Units* (CU). Cada CU se puede predecir tanto *inter-frame* como *intra-frame*. La imagen residual se codifica utilizando transformaciones de bloque. En cuanto a la codificación entrópica se utiliza el sistema de *Context Adaptative Binary Arithmetic Coding* (CABAC). Además, H.265 permite procesado en paralelo con lo que se acelera el proceso de codificación. H.265 soporta resoluciones hasta 8K UHDTV (8192x4320) de hasta 300 fps.
- **VP9**: Desarrollado por Google en 2012 proporciona una reducción de 50% del *bitrate* respecto a VP8 y con el objetivo de igualar o superar el rendimiento de H.265/HEVC [7]. Incorpora un tamaño de macrobloque que llega hasta 64x64 con posibilidad de un granulado específico de bloques de 4x4 para modelo temporal. Soporta 10 modos de predicción *intra-frame* y cuatro modos de predicción *inter-frame*. Utiliza la Transformada Discreta del Coseno (DCT), la Transformada Asimétrica Discreta del Seno (ADST) y la Transformada de Walsh-Hadamard para el modelo espacial. Se ha diseñado un filtro para eliminar los defectos de los bordes de los macrobloques. Soporta *High-dinamic-range* (HDR) y permite la codificación sin pérdidas. Al igual que H.265 permite el procesamiento en paralelo además de escalabilidad temporal y espacial. Tres años más tarde Google lanzó VP10 alcanzando mejoras de hasta 40% en cuanto a compresión respecto a VP9 con el coste de incrementar el tiempo de codificación.
- **AV1**: Codificador desarrollado por *Alliance for Open Media* (AOMedia) es una solución basada en VP10 con algunas mejoras adicionales. Se termino en 2018 y su principal propósito fue obtener una compresión mayor respecto a sus antecesores además de proporcionar escalabilidad con dispositivos modernos y diferentes enlaces de datos con una complejidad de decodificación muy baja. AV1 ofrece una reducción del 30% respeto al *bitrate* medio obtenido con VP9. Algunas de las ventajas que ofrece comparado con H.265/HEVC es que es libre de derechos de autor, ofrece más compresión y soporte para servicios compatibles con Apple, Google, Microsoft, Mozilla con lo que la mayoría de los navegadores soportan este codificador.

- **LCEVC**: MPEG-5 Parte 2 *Low Complexity Enhancement Vídeo Coding* (LCEVC) es un nuevo estándar de vídeo de MPEG que especifica capas de mejora que, combinadas con un vídeo base codificado, produce un flujo de vídeo mejorado. LCEVC mejora el rendimiento de compresión de cualquier códec básico (H.264, H.265, AV1, etc.) ofreciendo un *bitrate* hasta 40% menor tanto en Vídeo Bajo Demanda (VoD) como en transmisiones en directo [8]. Además, ofrece una mejora de 2 a 4 veces en la eficiencia de codificación lo que permite mejorar la calidad de experiencia de los usuarios sin tener que aumentar el *bitrate* o esperar a la mejora tecnológica de los dispositivos. Algunas de las ventajas de utilizar capas de mejora son: ampliar la capacidad de compresión del códec base, reducir la complejidad de la codificación y decodificación y proporcionar una plataforma para mejoras futuras. Todas estas ventajas permiten reducir los costes de almacenamiento en CDN hasta un 70% y proporcionar acceso a reproducción de vídeos a usuarios con redes de muy baja velocidad o congestionadas. Además, LCEVC permite el despliegue inmediato en servicios OTT que utilizan formatos estándar como HLS o MPEG-DASH. Los dispositivos que utilizan otros códecs podrán fácilmente implementar las capas de mejora de LCEVC mediante una simple actualización de software en su *player*.
- **H.266/VVC**: *Versatile Vídeo Coding* (VVC/H.266) es el estándar de codificación más reciente ofreciendo un 50% más de compresión que H.265/HEVC. Aumenta el tamaño de bloque en el modelo temporal a 128x128 con bloques con particionados binarios o ternarios respecto al particionado cuaternario de HEVC. Permite particionados diferentes para los planos de luminancia y crominancia y habilita la aceleración hardware mediante el procesamiento en paralelo. En cuanto a la predicción *intra-frame* se utilizan 67 modos de predicción en lugar de los 33 utilizados en HEVC, además de habilitar el uso de bloques rectangulares. En cuanto a la predicción *inter-frame* permite la predicción a partir de dos imágenes de referencia, además de incrementar de 2 a 3 dimensiones los grados de libertad de los vectores de movimiento. Para la transformación se utilizan bloques no rectangulares realizando transformadas de diferentes tipos en función del modo de predicción. Se incrementa el valor de QP o *quantizer* máximo de 51 a 63 para permitir tasas de bit menores. En la codificación entrópica se sigue utilizando Context *Adaptative Binary Arithmetic Coding* (CABAC). En el apartado siguiente se analizará cada uno de los cambios mencionados con más detalle.

3. Codificador H.266/VVC

Versatile Vídeo Coding (VVC/H.266) fue lanzado en julio de 2020 por Joint Vídeo Experts Team (JVET) y ISO/IEC Moving Picture Expers Group (MPEG) como el estándar de codificación de vídeo más reciente [9]. Su objetivo es ofrecer una compresión 50% mayor respecto a H.265/HEVC, así como ser utilizado en aplicaciones de vídeo de alta definición, HDR, *streaming* adaptativo con cambios de resolución, *streaming* de baja latencia, vídeo 360 grados inmersivo y codificación en capas.

En cuanto al diseño de VVC este sigue siendo el codificador híbrido basado en bloques utilizado en los codificadores anteriores. Para la representación de las imágenes, VVC utiliza al igual que sus antecesores tres planos, uno de luminancia y dos de crominancia con una profundidad de bit que puede ser de 8 o 10 bits. En cuanto al formato de muestreo este suele ser 4:2:0 para aplicaciones de vídeo típicas, aunque también soporta el formato 4:4:4 o 4:2:2 siendo este último el menos utilizado en aplicaciones reales. En la Figura 3 podemos ver el diagrama de bloques híbrido utilizado en el codificador VVC. Este incluye un particionado inicial en *Coding Tree Unit* (CTU), bloque de predicción *inter e intra-frame*, transformación y cuantización de la imagen residual, filtrado en bucle de la imagen tras el reescalado (cuantización inversa), transformación inversa y uso de *context adaptative binary arithmetic coding* (CABAC) en la codificación entrópica.

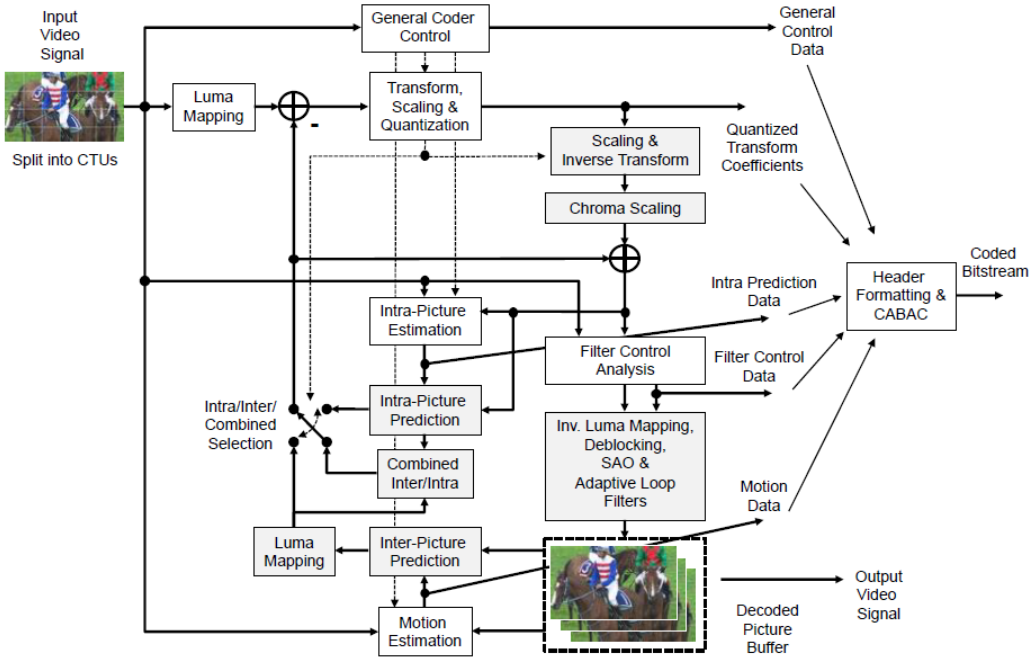

Figura 3. Diagrama de bloques codificador VVC

Comparado con H265/HEVC, VVC introduce elementos nuevos como: predicción *inter* e *intra-frame* combinada, mapeo de luminancia con reescalado, nuevos filtros de bucle y soporte de bloques no rectangulares. A continuación, se analizará en detalle todas las mejoras que VVC introduce.

3.1. Particionado de bloques

VVC mantiene el particionado cuaternario de HEVC además de permitir un particionado más flexible y de incrementar los tamaños de bloque. El particionado que introduce VVC consiste en dividir los bloques en trozos no cuadrados tanto para el plano de luminancia como para los de crominancia. De la misma forma que en HEVC, VVC utiliza CTUs formados por CUs como unidades de procesamiento básico. Las técnicas de particionado de bloques más importantes utilizadas en VVC se presentan a continuación:

- **Quadtree plus multi-type tree (QT+MTT):** Consiste en extender el particionado cuaternario de HEVC a un árbol multi-tipo en el que se utilizan particionados binarios o *Binary-Tree* (BT) y ternarios o *Ternary-Tree* (TT) [10]. Además, VVC incrementa el tamaño de bloque máximo de HEVC de 64x64 a 128x128. En la Figura 4 podemos observar los diferentes particionados que ofrece VVC para un *Coding Unit* (CU) de 4Nx4N donde N indica el número de muestras o píxeles de luminancia o crominancia.

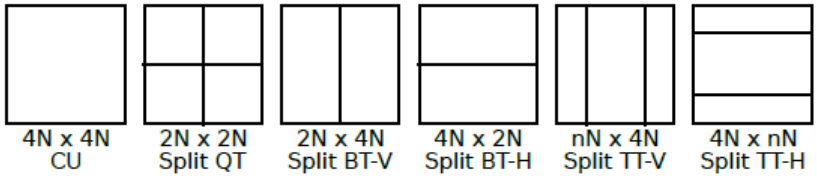

Figura 4. Particionados VVC de *Coding Unit* (CU)

Las particiones BT pueden ser simétricas horizontales o verticales mientras que las particiones TT están formadas por un rectángulo en el medio del CU cuyo tamaño es la mitad del CU. En la Figura 5 podemos ver un ejemplo de particionado QT+MTT de una imagen.

Figura 5. Ejemplo de particionado QT+MTT en VVC

Mediante el particionado QT+MTT la imagen residual obtenida como la resta entre la imagen actual e imágenes anteriores o posteriores y la aplicación de la estimación y compensación de movimiento, tiene una energía residual mínima y por tanto se consigue una alta compresión del vídeo.

- **Chroma separate tree (CST)**: VVC permite el uso de particionado diferente para los planos de luminancia y crominancia. Puesto que el plano de luminancia tiene una textura más fina y bordes más definidos que el de crominancia requiere de la utilización de CUs de tamaño pequeño. En los planos de crominancia donde los detalles de las texturas son menores se pueden utilizas CUs de tamaño mayor mejorando la velocidad de codificación.
- **Virtual pipeline data units (VPDUs)**: En los decodificadores VVC implementados en hardware existen regiones de bloques CTU conocidos como VPDUs. Estos bloques pueden ser decodificados en paralelo con el objetivo de incrementar la velocidad de decodificación.

3.2. Predicción imágenes

En cuanto a la predicción *intra-frame* e *inter-frame* VVC utiliza nuevas técnicas que permiten aumentar la compresión del vídeo.

- **Predicción intra-frame**: La predicción *intra-frame* se refiere a la capacidad de predecir los píxeles del CU actual a partir de los píxeles frontera de CUs adyacentes del mismo *frame*. Se mantienen los modos no angulares *Planar* y *DC* y se añaden 65 modos angulares en lugar de los 33 que tenía HEVC obteniendo un total de 67 modos de predicción *inter-frame* [9]. En la Figura 6 podemos observar los 67 modos de predicción *intra-frame* de VVC.

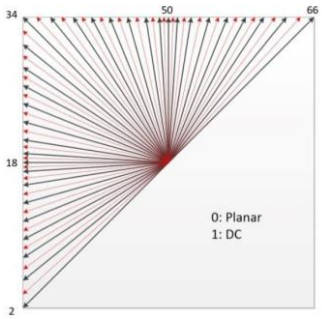

Figura 6. Modos de predicción *intra-frame* VVC

Existe una lista de *Most Probable Modes* (MPM) en la que se guardan 6 candidatos para poder seleccionar eficientemente entre las 67 opciones. Por otro lado, en VVC se habilita el uso de bloques rectangulares en la predicción *inter-frame* en lugar de únicamente bloques cuadrados. El uso de bloques no cuadrados obliga a extender el ángulo de predicción *Wide Angle Intra Prediction* (WAIP). Además, debido a que normalmente la información del plano de luminancia y crominancias está correlada se añade un nuevo predictor (*Cross-Component Prediction*) que permite obtener los píxeles del plano de crominancias a partir de una combinación lineal del plano de luminancia reconstruido. Por último, se puede realizar una predicción con modos no adyacentes al CU actual (*Multi Reference Line Prediction).*

- **Predicción inter-frame**: La predicción *inter-frame* se refiere a la capacidad de predecir los píxeles del CU actual a partir de CU de *frames* anteriores y posteriores. En concreto, VVC, al igual que sucedía en HEVC permite la predicción de una única imagen de referencia o bien de dos al mismo tiempo, haciendo una media de las dos predicciones [11]. Debido a que en los vídeos de escenarios reales los objetos se mueven en cualquier dirección se introduce el mecanismo de *Affine Motion Estimation Model* que consiste en utilizar tres en lugar de dos vectores de movimiento con lo que se aumentan los grados de libertad de predicción de 2 a 6. Otro mecanismo utilizado es el de *Overlapped Block Motion Compensation* que se basa en la superposición de los bordes de los bloques con el objetivo de evitar transiciones bruscas entre un bloque y otro. Por último, VVC añade la posibilidad de un particionado geométrico de los bloques, es decir, se puede realizar una partición no horizontal de los bloques CU utilizados en la predicción. En la Figura 7 podemos observar diferentes particiones no horizontales de los bloques CU.

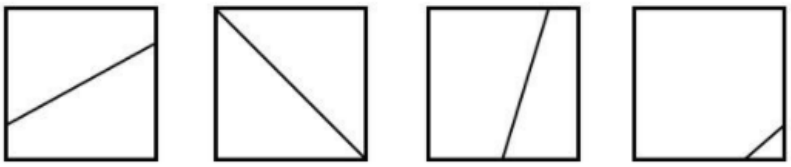

Figura 7. Ejemplos de particionado geométrico no horizontal

El particionado geométrico ayuda a encontrar un bloque parecido en la imagen de referencia y por tanto aumentar considerablemente la compresión.

3.3. Transformación y cuantización

Respecto a la transformación, VVC soporta bloques no rectangulares e incrementa el tamaño máximo de bloque a 64x64. HEVC emplea una única transformada discreta del coseno DCT (DCT-II) mientras que VVC utiliza tres transformadas diferentes DCT-II, DST-VII, and DCT-VIII siendo el codificador el que selecciona la combinación más adecuada en función del modo de predicción [12]. Por otro lado, se mantiene el parámetro de *quantizer* o QP de HEVC para controlar la cantidad de compresión introducida en el modelo espacial. Con el objetivo de alcanzar tasas de bit menores, se incrementa el valor máximo de QP de 51 a 63. Por último, se introduce el concepto de *Dependent quantization* (DQ) que permite el cambio o *switch* entre dos parámetros de QP inverso a la hora de decodificar. El parámetro QP elegido está en función del valor de QP del coeficiente anterior y para su elección se utiliza una máquina de estados de 4 estados.

3.4. Codificación entrópica

VVC utiliza, al igual que en HEVC, context *Adaptative Binary Arithmetic Coding* (CABAC) como mecanismo de codificación entrópica, aunque se cambia el modelo de cálculo de probabilidades de aparición de símbolos.

4. Metodología de evaluación de prestaciones

En esta sección se describirá la metodología empleada para realizar la comparativa de los codificadores de vídeo H.264/AVC, H.265/HEVC y H.266/VVC [13]. Para ello, en primer lugar, se describirán las características del conjunto de vídeos elegidos. A continuación, se analizarán los conceptos teóricos acerca de las métricas analizadas. Se realizará una breve descripción de la implementación de los codificadores utilizada, y los comandos empleados para la codificación y obtención de resultados.

4.1. Conjunto de vídeos de prueba

Se han seleccionado 10 vídeos, todos con las mismas características: duración 30 segundos, resolución 1280x720, 24 fps, formato de muestreo 4:2:0 (yuv420p) de profundidad de bit de 8 y GOP de 16. En la tabla 1 podemos ver un resumen de los vídeos elegidos.

Nombre secuencia	Resolución	Frecuencia imagen (fps)	Número de *frames*	Profundidad de bit
Agente 327	1280x720	24	720	8
Big Buck Bunny	1280x720	24	721	8
Building traffic	1280x720	24	722	8
Caminandes 3	1280x720	24	722	8
Churring night	1280x720	24	721	8
Close to bird	1280x720	24	722	8
Pedestrians	1280x720	24	721	8
Tearsofsteel	1280x720	24	722	8
Vaccine	1280x720	24	723	8
Venice	1280x720	24	721	8

Tabla 1: Secuencias de vídeo utilizados en las simulaciones

Todos los vídeos se han obtenido de fuentes libres de derechos de autor como Blender o Videvo y se ha utilizado la herramienta de *ffmpeg* para recodificarlos con las mismas características. Es fundamental que los vídeos tengan las mismas características para poder obtener resultados fiables en las simulaciones y comparativa de métricas.

4.2. Métricas

En el presente apartado se procederá a definir todas las métricas utilizadas en la comparativa, así como explicar la necesidad de su uso. En primer lugar, se ha calculado la complejidad espacial y temporal de las secuencias de vídeo con el objetivo de cubrir una amplia gama de escenarios reales. Como métricas de evaluación de prestaciones se ha obtenido el valor del PSNR, SSIM y VMAF de todas las secuencias para un conjunto de *bitrates* fijos. Finalmente, se presentan los resultados y conclusiones obtenidas a partir de estos.

4.2.1. Información espacial y temporal

En la recomendación ITU-T P.910 se especifican dos métricas que permiten clasificar contenido de vídeo: Información Espacial (SI) e Información Temporal (TI) [14].

- **Información Espacial (SI):** Métrica que indica la cantidad de detalle espacial de una imagen. Cuanto más alto sea su valor indica que la escena es más compleja. Está basada en un filtro de Sobel. Cada imagen del plano de luminancia en el instante n, (Fn) es filtrada mediante un filtro de Sobel [Sobel (Fn)]. Se calcula la desviación estándar sobre cada uno de los píxeles (std_{space}) para cada imagen filtrada con el filtro de Sobel. Esta operación se repite para cada una de las imágenes que componen el vídeo obteniendo un vector con valores de SI. Se escoge el valor máximo max_{time} como representativo del SI del vídeo. La fórmula que representa dicho proceso viene dada por la ecuación (1)

$$\mathrm{SI} = \max_{time}\{\mathrm{std}_{space}[\mathrm{Sobel}\,(\mathrm{Fn})]\} \quad (1)$$

- **Información Temporal (TI):** Métrica que indica la cantidad de cambio en el tiempo en una secuencia de vídeo. Cuanto más alto sea su valor indica que el vídeo tiene escenas de mucho movimiento. Está basado en una función de diferencia de movimiento $M_n(i,j)$ que se define como la diferencia entre los valores de los píxeles del plano de luminancia de una posición concreta de la imagen y la misma posición en imágenes consecutivas del mismo vídeo. La expresión de $M_n(i,j)$ viene dada por (2)

$$\mathrm{M}_n(i,j) = \mathrm{F}_n(i,j) - \mathrm{F}_n(i,j) \quad (2)$$

  donde $F_n(i,j)$ es el píxel de la fila i-ésima y la columna j-ésima de la imagen n.
  La medida de la información temporal (TI) viene dada por el valor máximo max_{time} de la desviación estándar sobre el espacio (std_{space}) de $M_n(i,j)$ para todo i y j. La ecuación que representa el proceso es la siguiente:

$$\mathrm{TI} = \max_{time}\{\mathrm{std}_{space}[\mathrm{Sobel}\,(\mathrm{Fn})]\} \quad (3)$$

Los valores de SI y TI obtenidos se han representado en un plano de dos dimensiones. Los vídeos se han elegido cuidadosamente para tener diferentes valores de SI y TI para de esta manera cubrir la mayoría de los escenarios reales posibles.

4.2.2. PSNR

Peak Signal-to-Noise Ratio o PSNR es una métrica que indica la medida de similitud entre dos imágenes, una imagen original o de referencia de alta calidad y otra degradada, procedente de una codificación de la imagen original [15]. Aplicado a una secuencia de imágenes se calcula para cada imagen y se promedia. La expresión matemática que permite obtener el PSNR viene dada por:

$$\text{PSNR} = 20\ \log_{10}\left(\frac{MAX_f}{\sqrt{MSE}}\right) \tag{4}$$

donde MSE es el error cuadrático medio dado por la expresión:

$$\text{MSE} = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1}\| f(i,j) - g(i,j)\|^2 \tag{5}$$

donde

- f representa la imagen original o de referencia.
- g representa la imagen degradada.
- m representa el número de filas de píxeles de las imágenes e i el índice de la fila actual.
- N representa el número de columnas de píxeles de la imagen y j el índice de la columna actual.
- MAX_f es el máximo valor de píxel de la imagen original.

El rango de valores que puede tomar el PSNR va desde los 0 dB (mínima similitud) hasta los ∞ dB (máxima similitud). La similitud será máxima cuando el error cuadrático medio es nulo, es decir cuando las dos imágenes, original y distorsionada son idénticas. Cabe destacar que el PSNR se puede utilizar como métrica objetiva para comprar dos vídeos, pero no como indicador subjetivo de la calidad de una imagen puesto que no tiene en cuenta la posición de los cambios entre las dos imágenes. En la Figura 8 podemos observar dos imágenes codificadas en las que el valor del PSNR es idéntico, sin embargo, se puede apreciar claramente como en la imagen (b) toda la degradación de la imagen está concentrada en un cuadrado mientras que en la (a) está repartida por toda la imagen, siendo su evaluación subjetiva claramente mejor.

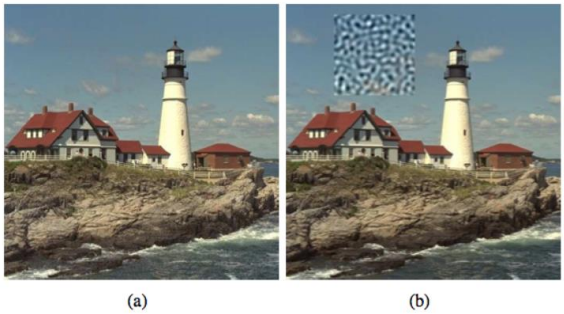

Figura 8. Imágenes con el mismo valor de PSNR

4.2.3. SSIM

Structural Similarity Index Measure (SSIM) es una métrica objetiva de mayor complejidad que el PSNR cuyo objetivo es adaptarse mejor a la calidad subjetiva de la imagen. Para ello, se realiza una ponderación distinta de las distorsiones producidas en luminancia, contraste y estructura de la imagen, dándole mayor importancia a aquellas que más afectan a la calidad subjetiva como por ejemplo los cambios de estructura y menor a las que tienen que ver con la luminancia [16]. Con ello, SSIM es una métrica que no compara los píxeles de la imagen, si no los elementos percibidos por el ser humano. Las expresiones que permiten obtener el SSIM se presentan a continuación.

$$\mathrm{SSIM(x,y)} = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \quad (6)$$

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_2'} \quad (7)$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2'} \quad (8)$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_{xy} + C_3'} \quad (9)$$

$$C_3 = \frac{C_2}{2} \quad (10)$$

En el caso de que $\alpha = \beta = \gamma = 1$ se tiene que

$$\mathrm{SSIM(x,y)} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (11)$$

$$SSIM_{ij} = W_Y \cdot SSIM_{IJ}^{Y} + W_{Cb} \cdot SSIM_{IJ}^{Cb} + W_{Cr} \cdot SSIM_{IJ}^{Cr} \quad (12)$$

donde

- α, β, γ son los coeficientes de importancia
- μ_x es el valor medio de la muestra de luminancia de la imagen x.
- μ_y es el valor medio de la muestra de luminancia de la imagen y.
- σ_x es la desviación estándar de la muestra de luminancia de la imagen x.
- σ_y es la desviación estándar de la muestra de luminancia de la imagen y.
- σ_{xy} es la covarianza de la luminancia de las dos imágenes x e y.
- C_1, C_2 son los coeficientes de estabilización
- W_Y es el peso de la luminancia
- W_{Cb} es el peso de la crominancia de azul.
- W_{Cr} es el peso de la crominancia de rojo.

Los valores de SSIM pueden variar entre 0 (mínima similitud) y 1 (máxima similitud y misma imagen).

4.2.4. VMAF

Vídeo Multi-method Assesment Fusion o VMAF es una métrica de calidad de vídeo desarrollada por Netflix cuyo objetivo es predecir la calidad subjetiva de un vídeo mediante la combinación de métricas elementales de medida de calidad. Se utiliza el algoritmo de *Machine-Learning Support Vector Machine* (SVM) para fusionar todas estas métricas [17]. El algoritmo asigna pesos a cada métrica elemental de manera que la métrica final obtenida mantiene los puntos fuertes de las métricas individuales. El modelo de *machine-learning* es entrenado con MOS (*Mean opinión scores*) obtenidos con el experimento subjetivo NFLX Vídeo Dataset.

En el experimento NFLX Vídeo Dataset, Netflix generó un conjunto de vídeos de referencia, codificados y distorsionados con el objetivo de cubrir escenarios de animación, exteriores e interiores, con objetos, con personas en movimiento, etc. Mediante una evaluación subjetiva con usuarios reales se determinó el MOS medio para cada vídeo. El MOS es una métrica en la que se le pregunta al usuario por la calidad del vídeo y este debe darle una nota del 1 al 5 siendo 1 la calidad más baja y 5 calidad excelente. Para el cálculo del MOS se utilizó la evaluación subjetiva de *Double Stimulus Impairment Scale* (DSIS) en un entorno en el que la iluminación, distancia a la TV y sala de estar entre otras características del entorno estaban totalmente controladas.

La versión actual de VMAF 0.3.1 utiliza las siguientes métricas de evaluación de calidad:

- ***Visual Information Fidelity* (VIF):** Es una métrica de calidad de imagen basada en la calidad percibida por el ser humano que en general ofrece mejor correlación con la evaluación subjetiva que el PSNR.
- **Detail Loss Metric (DLM):** Se trata de una métrica que consiste en la medida de la pérdida de detalle que afecta a la visualización del vídeo y a la redundancia que puede captar la atención del usuario.
- **Movimiento:** Medida de las características temporales del vídeo en la que se obtiene la diferencia temporal entre *frames* adyacentes. Para ello se calcula la diferencia en valor absoluto de la componente de luminancia.

En la Figura 9 podemos ver el esquema del algoritmo de VMAF.

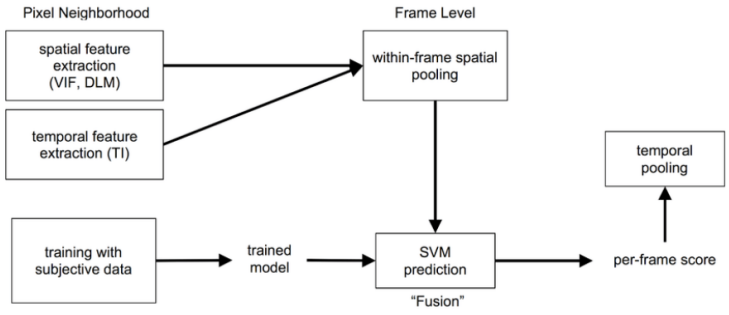

Figura 9. Esquema del algoritmo de VMAF

Se observa que el modelo SVM recibe tanto los vídeos de prueba como las métricas. VMAF varía entre 0 y 100, siendo 0 el valor mínimo (peor calidad) y 100 el valor máximo (mejor calidad).

4.3. Codificación en H.264 y H.265

Respecto a la implementación de los codificadores se ha elegido *ffmpeg* para los codificadores de AVC/H.264 y HEVC/H.265 y la implementación del codificador VVC de Fraunhofer (*Versatile Vídeo Encoder*; VVenC) para codificar los vídeos en H.266/VVC.

4.3.1. Herramienta ffmpeg

La herramienta de *ffmpeg* es una colección de software libre que puede grabar, codificar y hacer *streaming* de audio y vídeo ofreciendo soporte para la mayoría de los codificadores del mercado [18]. Está desarrollado en GNU/Linux, aunque puede ser compilado en la mayoría de los sistemas operativos. En este caso se ha optado por la utilización de Windows.
Incluye tres herramientas para tratamiento de vídeo:

- `ffmpeg`: herramienta principal y más utilizada. Incluye las herramientas de conversión y procesado de vídeo.
- `ffplay`: reproductor de multimedia básico.
- `ffprobe`: analizador de contenido multimedia que proporciona información técnica.

4.3.2. Codificación de vídeos

En primer lugar, se convierte el vídeo original de formado MP4 a YUV (luminancia y crominancias)

- Conversión MP4 a YUV**:**

```
# ffmpeg -i video.mp4 video.yuv
```

A continuación, se procede a codificar el vídeo a cuatro *bitrates* fijos: 128k, 256k, 512k y 1024k utilizando la librería h264.

- *Bitrate* 128k**:**

```
# ffmpeg -f rawvídeo -pix_fmt yuv420p -video_size 1280x720 -framerate 24 -i video.yuv -vcodec libx264 -b:v 128k -s 1280x720 -r 24 -g 16 -bf 2 -sc_threshold 0 -b_strategy 0 -flags -cgop -report video_h264_128k.mp4
```

- *Bitrate* 256k**:**

```
# ffmpeg -f rawvideo -pix_fmt yuv420p -video_size 1280x720 -framerate 24 -i video.yuv -vcodec libx264 -b:v 256k -s 1280x720 -r 24 -g 16 -bf 2 -sc_threshold 0 -b_strategy 0 -flags -cgop -report video_h264_256k.mp4
```

- *Bitrate* 512k**:**

```
# ffmpeg -f rawvideo -pix_fmt yuv420p -video_size 1280x720 -framerate 24 -i video.yuv -vcodec libx264 -b:v 512k -s 1280x720 -
```

```
r 24 -g 16 -bf 2 -sc_threshold 0 -b_strategy 0 -flags -cgop -
report video_h264_512k.mp4
```

- *Bitrate* 1024k**:**

```
# ffmpeg -f rawvideo -pix_fmt yuv420p -video_size 1280x720 -
framerate 24 -i video.yuv -vcodec libx264 -b:v 1024k -s 1280x720
-r 24 -g 16 -bf 2 -sc_threshold 0 -b_strategy 0 -flags -cgop -
report video_h264_1024k.mp4
```

Se codifica también utilizando la librería de h265 para obtener los vídeos de H.265 codificados a los mismos cuatro *bitrates*.

- *Bitrate* 128k**:**

```
# ffmpeg -f rawvideo -pix_fmt yuv420p -video_size 1280x720 -
framerate 24 -i video.yuv -vcodec libx265 -b:v 128k -x265-
params"keyint=16:min-keyint=12:no-opengop=0:scenecut=0:bframes=2
:b-adapt=0" -s 1280x720 -r 24 -report video_h265_128k.mp4
```

- *Bitrate* 256k**:**

```
# ffmpeg -f rawvideo -pix_fmt yuv420p -video_size 1280x720 -
framerate 24 -i video.yuv -vcodec libx265 -b:v 256k -x265-
params"keyint=16:min-keyint=12:no-opengop=0:scenecut=0:bframes=2
:b-adapt=0" -s 1280x720 -r 24 -report video_h265_256k.mp4
```

- *Bitrate* 512k**:**

```
# ffmpeg -f rawvideo -pix_fmt yuv420p -video_size 1280x720 -
framerate 24 -i video.yuv -vcodec libx265 -b:v 512k -x265-
params"keyint=16:min-keyint=12:no-opengop=0:scenecut=0:bframes=2
:b-adapt=0" -s 1280x720 -r 24 -report video_h265_512k.mp4
```

- *Bitrate* 1024k**:**

```
# ffmpeg -f rawvideo -pix_fmt yuv420p -video_size 1280x720 -
framerate 24 -i video.yuv -vcodec libx265 -b:v 1024k -x265-
params"keyint=16:min-keyint=12:no-opengop=0:scenecut=0:bframes=2
:b-adapt=0" -s 1280x720 -r 24 -report vídeo_h265_1024k.mp4
```

Los comandos utilizados tienen el siguiente significado:

- `-s [: stream_specifier] size:` Resolución de salida del vídeo.
- `-r [: stream_specifier] fps:` *frames* por segundo del vídeo resultante.
- `-pix_fmt yuv420p:` Indica que el formado de muestreo es 4:2:0 de 8 bits.
- `-g:` Se utiliza para especificar el tamaño del *Group of Pictures* (GOP).
- `-bf integer:` número de tramas de tipo B entre tramas que no son de tipo B.

- `-sc_threshold` *`(scenecut):`* Marca el límite para la detección de cambio de escena.
- `-b_strategy` *`(b-adapt):`* Algoritmo que coloca tramas B de forma adaptativa en función del *frame*.
- `-flags -cgop:` Establece un GOP cerrado.
- `keyint, min-keyint:` Establece tamaño de GOP y tamaño de GOP mínimo.
- `no-open-gop:` Establece que el GOP debe ser cerrado.
- `bframes:` Establece el número de tramas B.
- `b-adapt:` Algoritmo que coloca tramas B de forma adaptativa en función del *frame*.

4.4. Codificación en H.266

En cuanto a la codificación de los vídeos en VVC/H.266 no se ha podido emplear *ffmpeg* puesto que al ser un codificador nuevo aún no se han implementado las librerías necesarias para su uso. En su lugar se ha optado por la utilización de la implementación del codificador VVC de Fraunhofer (*Versatile Vídeo Encoder*; VVenC).

4.4.1. Herramientas VVenC y VVdeC

El conjunto de las herramientas VVenC y VVdeC se han utilizado para codificar y decodificar los vídeos en VVC/H.266. Las motivación de realizar el estudio utilizando la implementación del codificador VVC de Fraunhofer (*Versatile Video Encoder*; VVenC) en lugar de los modelos de prueba o *VVC test model* (VTM), típicamente utilizados en comparativas de codificadores son las siguientes: VVenC es una implementación rápida y eficiente del codificador VVC que permite obtener una calidad visual aceptable con tiempos de codificación bajos, ofrece cinco velocidades de codificación o *presets*, codificación en paralelo, entre otros.

En concreto, el codificador *Versatile Vídeo Coding* de Fraunhofer (VVenC) ofrece las siguientes características:

- Implementación de codificador fácil de usar con cinco ajustes o *presets* predefinidos de calidad y velocidad de codificación.
- Optimización perceptiva para mejorar la calidad de vídeo subjetiva, basada en el modelo visual XPSNR.
- Admite una o dos pasadas (*single-pass* o *two-pass*) en el proceso de codificación además de ofrecer codificación de velocidad de bits variable (VBR).
- Existe una interfaz en modo experto en el proceso de codificación disponible que permite un control más detallado del proceso de codificación.

En el siguiente apartado se describirán en detalle los comandos utilizados para realizar la codificación para los diferentes *bitrates* elegidos, así como el comando utilizado para la decodificación de estos.

4.4.2. Codificación de vídeos

En cuanto a la codificación de vídeos se han codificado los mismos vídeos que en H.264 y H.265 a los mismos *bitrates.* Los comandos utilizados son los siguientes.

- *Bitrate* 128k**:**

```
# vvencapp -i video.yuv -s 1280x720 -c yuv420 --internal-bitdepth
8 -r 24 -b 128000 --qpa 0 -g 16 -t 8 -o video_h266_256k.266
```

- *Bitrate* 256k**:**

```
# vvencapp -i video.yuv -s 1280x720 -c yuv420 --internal-bitdepth
8 -r 24 -b 256000 --qpa 0 -g 16 -t 8 -o video_h266_256k.266
```

- *Bitrate* 512k**:**

```
# vvencapp -i video.yuv -s 1280x720 -c yuv420 --internal-bitdepth
8 -r 24 -b 512000 --qpa 0 -g 16 -t 8 -o video_h266_256k.266
```

- *Bitrate* 1024k**:**

```
# vvencapp -i video.yuv -s 1280x720 -c yuv420 --internal-bitdepth
8 -r 24 -b 1024000 --qpa 0 -g 16 -t 8 -o video_h266_256k.266
```

donde las opciones utilizadas tienen el siguiente significado:

- `--size, -s:` Establece la resolución del vídeo de entrada.
- `--format, -c:` Especifica el formato de muestreo, yuv420 para 4:2:0 de 8 bits.
- `--internal-bitdepth:` Indica que la profundidad de color del vídeo de salida es de 8 bits.
- `--framerate, -r:` Indica el número de *frames* por segundo del vídeo de entrada.
- `--bitrate, -b:` Especifica un *bitrate* fijo que debe seguir el codificador.
- `--qpa:` Adaptación de QP perceptual (QPA) para mejorar la calidad subjetiva del vídeo.
- `--gopsize, -g:` Se utiliza para especificar el tamaño del *Group of Pictures* (GOP).
- `--threads, -t:` Número de hilos. Si la resolución es >= 1280x720 toma el valor de 8, en caso contrario toma el valor de 4.

Existen 5 ajustes o *presets: faster, fast, medium, slow* y *slower*. En función del ajuste elegido el codificador prioriza más el tiempo de codificación o la calidad visual. Se ha mantenido el *preset* por defecto de *medium* puesto que no tarda demasiado tiempo en realizar la codificación y aumenta ligeramente la calidad visual.

Por último, respecto a la decodificación, el decodificador se proporciona como herramienta por separado, Fraunhofer *Versatile Vídeo Decoder* (VVdeC) [20]. Se ha utilizado para la obtención del archivo en formato YUV a partir del vídeo codificado en H.266. El comando empleado para todos los vídeos y *bitrates* es el siguiente.

- Conversión H266 a YUV**:**

```
# vvdecapp -b video.266 -o video_out.yuv
```

4.5. Herramientas para obtención de métricas

Respecto a las herramientas utilizadas para la obtención de métricas, estas han sido *Spatial Information and Temporal Information* (SITI) para la obtención de la información temporal y espacial, *Vídeo Quality Measurement Tool* (VQMT) para la obtención del PSNR y SSIM y *Vídeo Multi-Method Assessment Fusion* (VMAF) para la obtención del VMAF, todas gratuitas con el código fuente proporcionado en la página web de cada una.

4.5.1. SITI - Spatial Information and Temporal Information

Herramienta de comandos creada en Python cuyo propósito es el cálculo de la información espacial (SI) y temporal (TI) de una secuencia de vídeo [21]. SITI tiene como dependencias las librerías de *ffmpeg* y Python versión 3.7 como mínimo. Para la instalación se requiere de la ejecución del comando `pip3 install --user siti` y para lanzar la aplicación se utiliza el comando `siti`. El comando utilizado para el cálculo de SITI de un vídeo ha sido el siguiente:

- Obtención de información espacial y temporal (SI TI):

```
# siti -of json -n 720 --width 1280 --height 720 vídeo.yuv
```

Las opciones recibidas tienen el significado siguiente.

- `--output-format, -of {json,csv}`: Formado del resultado obtenido.
- `--num-frames, -n NUM_FRAMES`: Número de *frames* del vídeo proporcionado.
- `--height HEIGHT`: Número de píxeles en vertical.
- `--width WIDTH:` Número de píxeles en horizontal.

Se ha guardado el valor SI TI de cada uno de los 10 vídeos de prueba para posteriormente dibujarlos todos conjuntamente en una gráfica.

4.5.2. VQMT - Vídeo Quality Measurement Tool

Software de implementación rápida que permite la obtención de métricas objetivas como PSNR, SSIM, MS-SSIM, VIFp entre otras [22]. Está implementado en C++ junto con las librerías de OpenCV basado en las implementaciones de Matlab de los creadores. Para realizar la compilación de la herramienta, VQMT tiene como dependencias las librerías de OpenCV (*core* y módulos *imgproc*) además de la herramienta de Linux *make*. Para la instalación se requiere de ejecutar el comando *make* en el directorio de la herramienta. Una vez ejecutado se genera el ejecutable en la ruta *build/bin/Release* del mismo directorio.

Los comandos utilizados para la obtención del PSNR y SSIM se presentan a continuación.

- Obtención de PSNR:

```
# vqmt video_original.yuv video_distorsionado.yuv 720 1280 720 1
ssim_video PSNR
```

- Obtención de SSIM:

```
# vqmt video_original.yuv video_distorsionado.yuv 720 1280 720 1
ssim_video SSIM
```

Las opciones recibidas son los siguientes y en el orden dado.

- `OriginalVídeo`: Vídeo original en formato YUV con 8 bits por píxel.
- `ProcessedVídeo`: Vídeo distorsionado en formato YUV con 8 bits por píxel.
- `Height`: Número de píxeles en vertical.
- `Width`: Número de píxeles en horizontal.
- `NumberOfFrames`: Número de *frames* a procesar.
- `ChromaFormat`: Formato de muestreo. 0: yuv400, 1: yuv420, 2: yuv422, 3: yuv444.
- `Output`: Nombre del archivo que contiene el resultado de la métrica.
- `Metrics`: Métrica a calcular. Lista de métricas disponibles: PSNR, SSIM, MSSSIM, VIFP, PSNRHVS.

Tras ejecutar el comando se crea un archivo Excel con las métricas de PSNR o SSIM calculadas para cada *frame* y finalmente se proporciona el valor medio calculado como la suma de métricas dividido entre el número de *frames*.

4.5.3. VMAF - Vídeo Multi-Method Assessment Fusion

Librería de cálculo de VMAF desarrollado por Netflix. Este paquete de software incluye una biblioteca C independiente *libvmaf* que habilita el uso de la herramienta de comandos *vmaf* [23]. Dicha herramienta utiliza los modelos de *machine-learning* de Netflix para proporcionar un valor de VMAF como medida de la calidad subjetiva de la imagen. Para la instalación se requiere de la librería de *libvmaf* con todas las dependencias que esta requiere.
El comando para el cálculo de VMAF empelado es el siguiente.

- Obtención de VMAF:

```
# vmaf -r video.yuv -d video_h266_256k.yuv -w 1280 -h 720 -p 420
-b 8
```

Las opciones utilizadas se definen como:

- `--reference, -r`: Vídeo original en formato YUV.
- `--distorted, -d:` Vídeo distorsionado en formato YUV.
- `--width, -w`: Número de píxeles en horizontal.
- `--height, -h`: Número de píxeles en vertical.
- `--pixel_format/-p`: Formato de muestreo. 420: yuv420, 422: yuv422, 444: yuv444
- `--bitdepth/-b:` Profundidad de color (8/10/12)

5. Pruebas y resultados

Respecto a los resultados obtenidos, estos se presentarán en el siguiente orden. En primer lugar, se presentará la información espacial y temporal de todos los vídeos. A continuación, se verán las métricas de PSNR, SSIM y VMAF para cada uno de los vídeos. Por último, se mostrarán los resultados obtenidos de tiempo de codificación, tamaño final de los ficheros, valor medio de QP de las tramas Intra entre otros.

5.1. Información espacial y temporal

En la Figura 10 podemos observar los valores de información espacial (SI) y temporal (TI) de los 10 vídeos utilizados para la comparativa de codificadores.

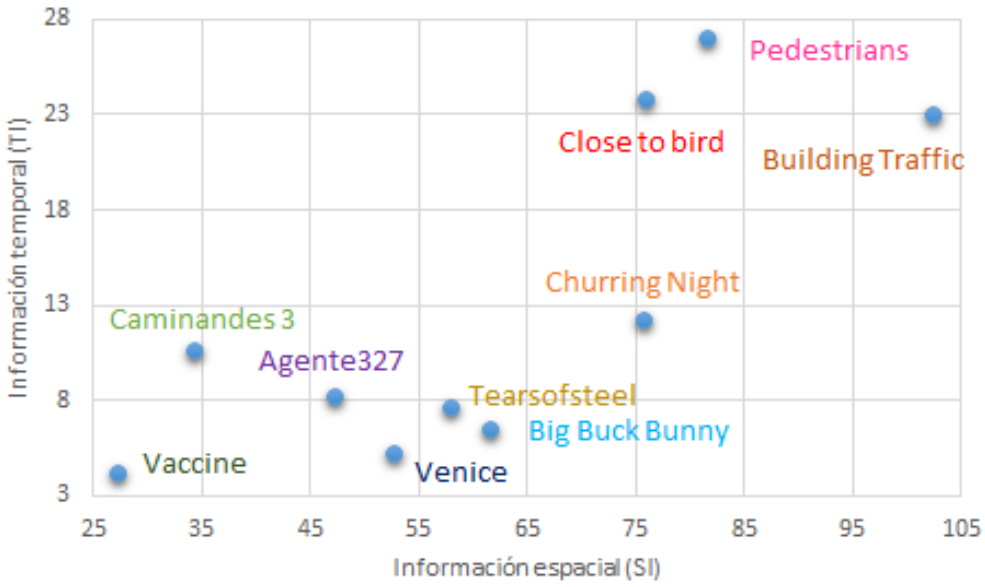

Figura 10. Valores de información espacial (SI) y temporal (TI)

Los valores de información espacial varían entre 27 y 103 mientras que los de información temporal están en el rango de 4 a 27. Estos valores indican que los vídeos tienen una complejidad temporal y espacial diferente. Esto es debido a que se han seleccionado una amplia variedad de vídeos que tienen diferentes escenas, detalles y movimiento. Las escenas en las que hay mucho detalle tienen una complejidad o información espacial mayor que aquellas que tienen menos detalle. Las escenas en las que hay mucho movimiento tienen una complejidad o información temporal mayor que aquellas en las que el movimiento es menor.

Por ejemplo, en vídeos como *Vaccine* o *Agente327* existen escenas con menos detalle en los que el movimiento es menor comparado con secuencias como *Pedestrians* o *Close to bird* en los que las escenas son más complejas puesto que existen más detalles y movimiento. Analizaremos el comportamiento del codificador VVC con cada uno de estos escenarios y como estos afectan a la calidad visual de los vídeos codificados.

5.2. Resultados PSNR, SSIM y VMAF

Las codificaciones se han realizado en los tres codificadores AVC/H264, HEVC/H.265 y VVC/H.266 para los *bitrates* de 128.000 bps (128k), 256.000 bps (256k), 512.000 bps (512k) y 1.024.000 bps (1024k). Se han seleccionado dichos *bitrates* puesto que cubren una amplia gama de calidades de vídeo, desde una calidad muy baja (128k) a una calidad alta (1024k).

Los resultados de métricas obtenidas se presentarán en el siguiente orden: PSNR, SSIM y VMAF. Para cada una de las métricas se analizará el resultado para la secuencia de Agente327 y a continuación se adjuntarán las gráficas de todas las demás secuencias.

5.2.1. Resultados PSNR

En la Figura 11 podemos observar los valores de PSNR obtenidos para la secuencia Agente327, con los *bitrates* de 128k, 256k, 512k y 1024k y los codificadores AVC/H264, HEVC/H.265 y VVC/H.266.

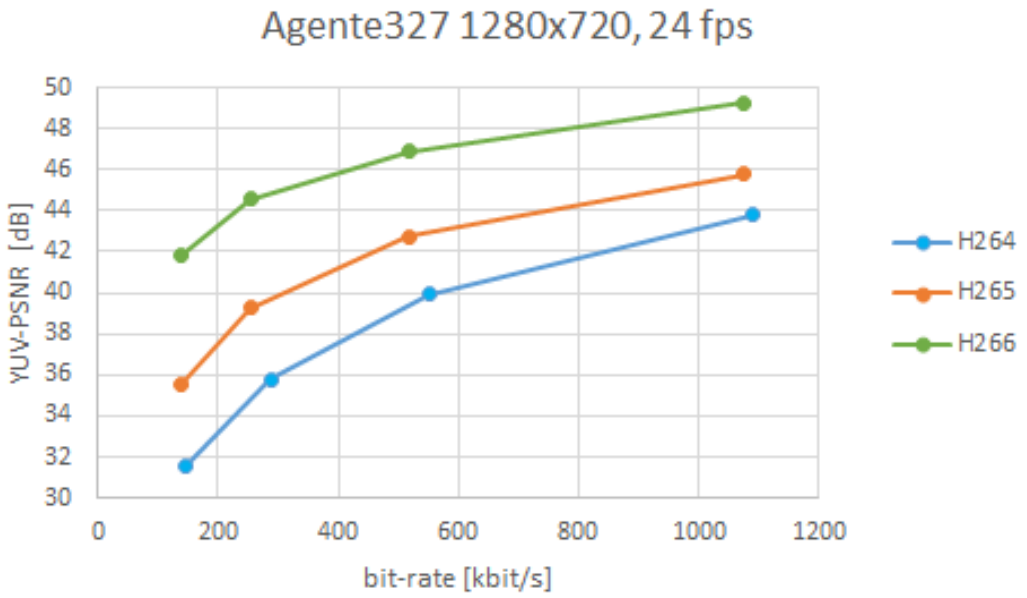

Figura 11. YUV-PSNR para secuencia Agente327 en H.264, H.265 y H.266

Se observa claramente como los valores de PSNR obtenidos con el codificador VVC son mayores que los obtenidos en HEVC y AVC. En concreto, la mejora de PSNR de VVC con respecto a HEVC es de [6,32, 5,29, 4,16, 3,49] dB correspondientes a los *bitrates* de [128k, 256k, 512k, 1024k]. Se observa también que la mejora de calidad es mayor para *bitrates* bajos y menor a medida que incrementamos el *bitrate*.

En cuanto a la mejora que introduce HEVC respecto a AVC esta es también muy significativa. En concreto es de [3,96, 3,49, 2,84, 2,01] dB correspondientes a los *bitrates* de [128k, 256k, 512k, 1024k]. A pesar de ser una mejora significativa, en este vídeo en concreto, VVC tiene una mejora superior en cuanto a PSNR respecto a HEVC que la que tiene HEVC con respecto a AVC. En las Figuras 12 y 13 se muestran los resultados de PSNR para las 10 secuencias de vídeo.

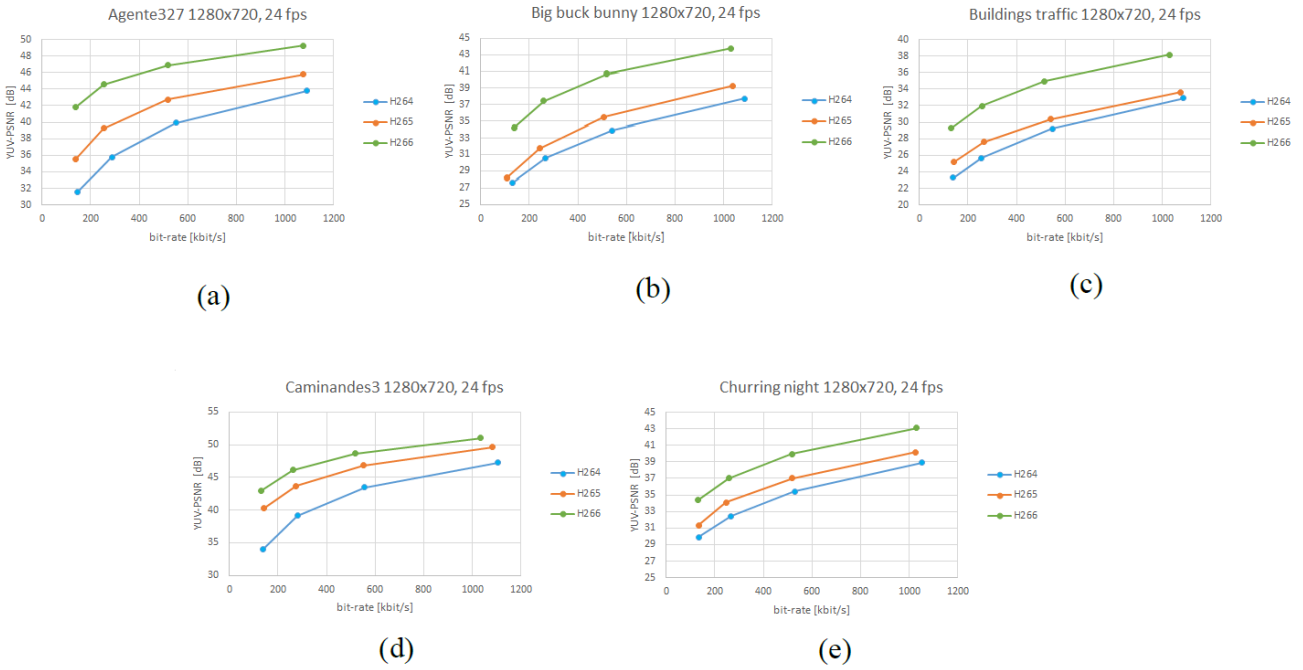

Figura 12. Valores de YUV-PSNR para secuencias [1-5] en H.264, H.265, H266

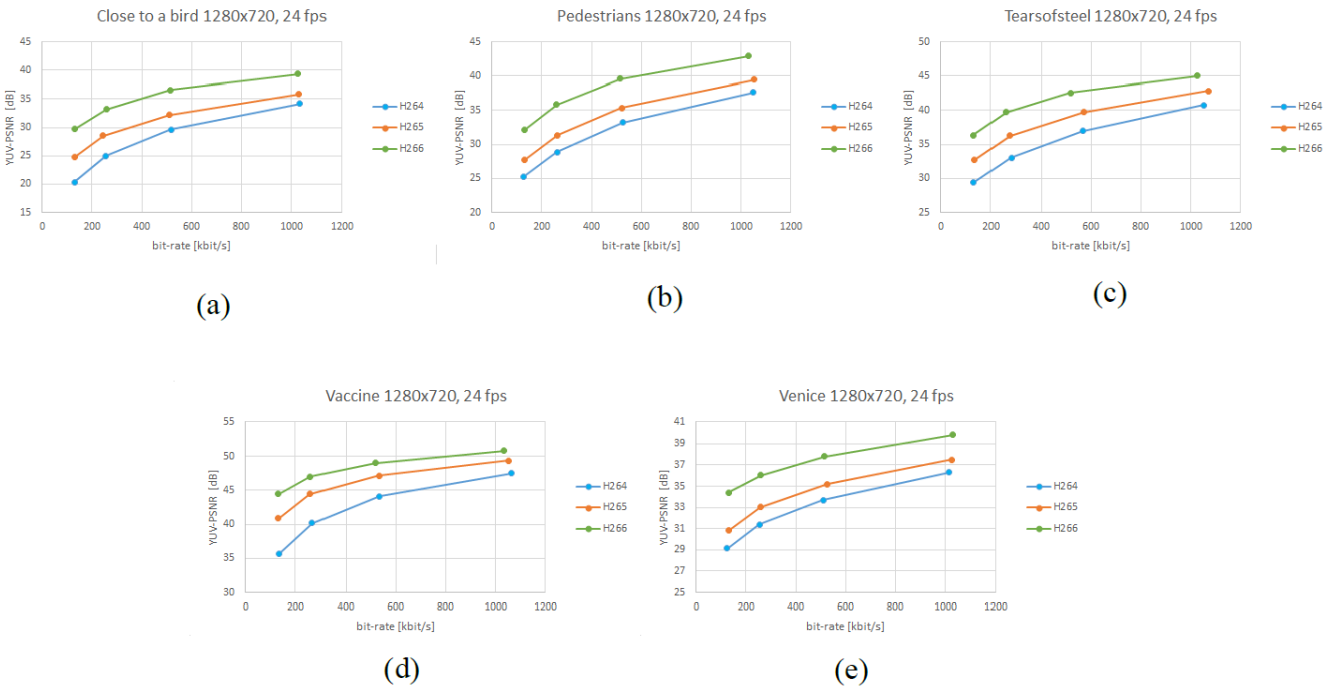

Figura 13. Valores de YUV-PSNR para secuencias [6-10] en H.264, H.265, H266

Se puede observar como para todos los codificadores, a medida que aumenta el *bitrate* el valor del PSNR aumenta. Además, se cumple que en todos los casos VVC tiene un PSNR siempre superior a HEVC y AVC. En la mayoría de las secuencias se confirma también que el salto de PSNR es mayor de HEVC a VVC que de AVC a HEVC.

5.2.2. Resultados SSIM

En la Figura 14 podemos observar los valores de SSIM obtenidos para la secuencia Agente327, con los *bitrates* de 128k, 256k, 512k y 1024k y los codificadores AVC/H264, HEVC/H.265 y VVC/H.266.

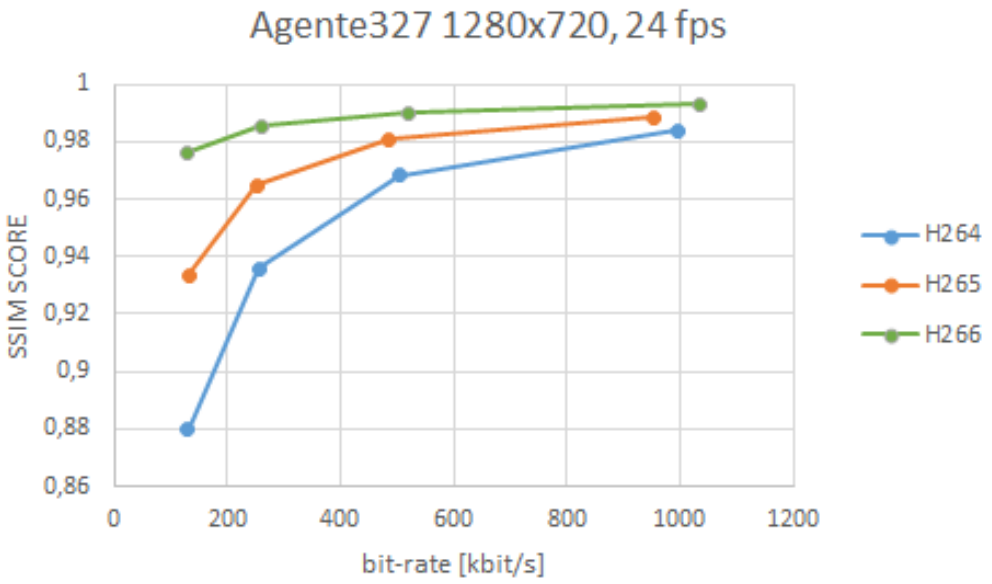

Figura 14. SSIM para secuencia Agente327 en H.264, H.265 y H.266

Se puede observar claramente como en todos los valores de SSIM, VVC superan a los de HEVC y AVC. En concreto, la mejora de SSIM de VVC con respecto a HEVC es de [0,042, 0,021, 0,009, 0,004] puntos correspondientes a los *bitrates* de [128k, 256k, 512k, 1024k]. Se observa que, al igual que sucedía con el PSNR, la mejora es mayor cuando el *bitrate* es menor, es decir, VVC ofrece una mejor calidad visual respecto a HEVC cuando el *bitrate* es menor.

La mejora que introduce HEVC respecto a AVC sigue siendo significativa, aunque en este caso se mantiene el orden de mejora de AVC a HEVC y de HEVC a VVC. En las Figuras 15 y 16 se muestran los resultados de SSIM para las 10 secuencias de vídeo.

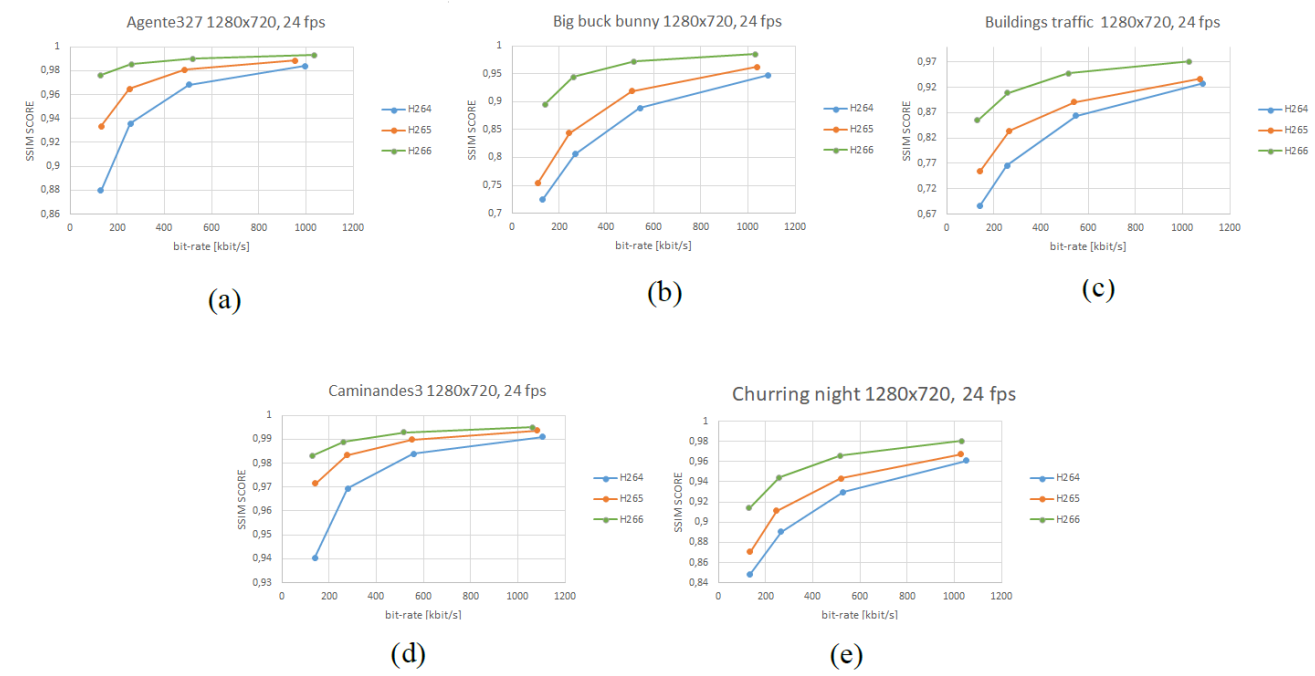

Figura 15. Valores de SSIM para secuencias [1-5] en H.264, H.265, H266

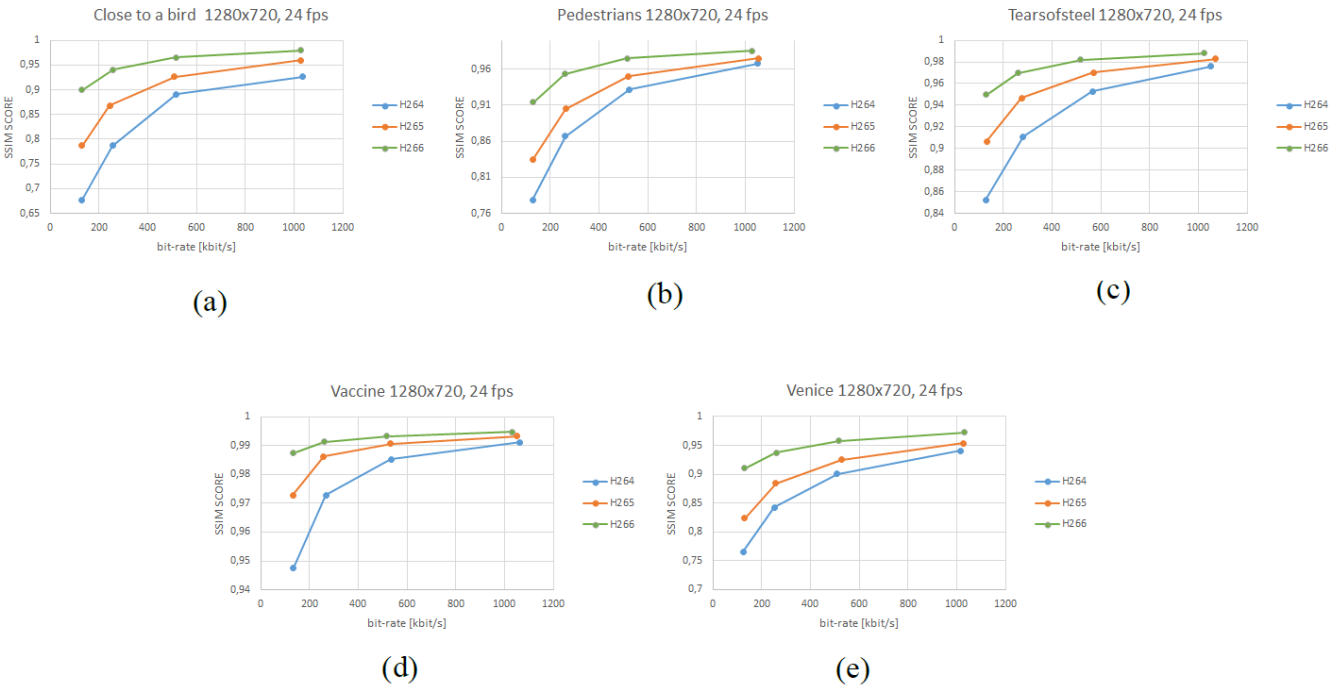

Figura 16. Valores de SSIM para secuencias [6-10] en H.264, H.265, H266

Se puede observar como para todos los codificadores, a medida que aumenta el *bitrate* el valor del SSIM aumenta. Además, se cumple que en todos los casos VVC tiene un SSIM siempre superior a HEVC y AVC.

5.2.3. Resultados VMAF

En la Figura 17 podemos observar los valores de VMAF obtenidos para la secuencia Agente327, con los *bitrates* de 128k, 256k, 512k y 1024k y los codificadores AVC/H264, HEVC/H.265 y VVC/H.266.

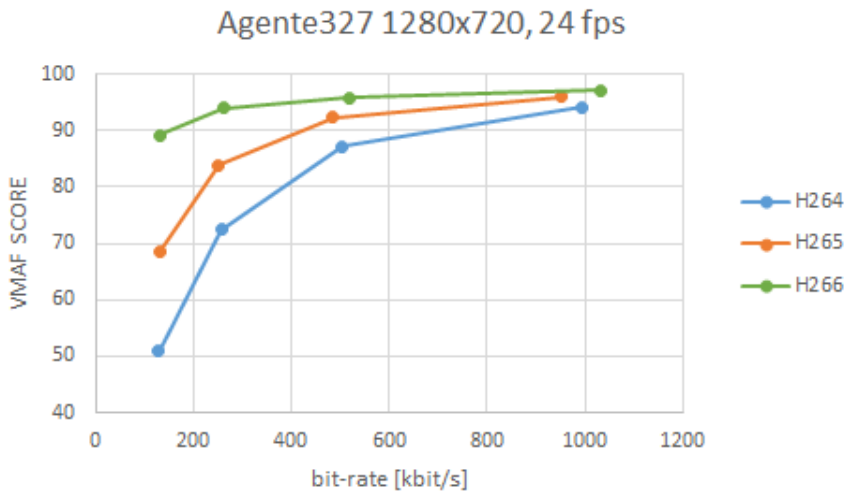

Figura 17. VMAF para secuencia Agente327 en H.264, H.265 y H.266

Respecto a los valores de VMAF obtenidos se sigue cumpliendo que VVC ofrece mejor calidad que HEVC y AVC. En concreto, la mejora de VMAF de VVC con respecto a HEVC es de [20,59, 9,99, 3,51, 1,26] puntos correspondientes a los *bitrates* de [128k, 256k, 512k, 1024k].

Se observa que, al igual que sucedía con el PSNR y SSIM, la mejora de calidad es mayor cuando el *bitrate* es menor. En este caso podemos ver que dicha mejora es mucho mayor en el caso de 128k de *bitrate*. Una mejora del valor de 20,59 puntos indica que la calidad subjetiva percibida por el usuario es mucho mejor en VVC que en HEVC para un *bitrate* de 128k. Para el *bitrate* de 1024k la mejora de VMAF es de apenas 1,26 puntos, lo que indica que el cambio de calidad de vídeo percibido por el usuario al utilizar VVC es muy bajo con respecto a utilizar HEVC o AVC.

En cuanto a la mejora que introduce HEVC respecto a AVC sigue siendo significativa, manteniendo el orden de mejora de HEVC respecto a AVC. En las Figuras 18 y 19 se muestran los resultados de VMAF para las 10 secuencias de vídeo.

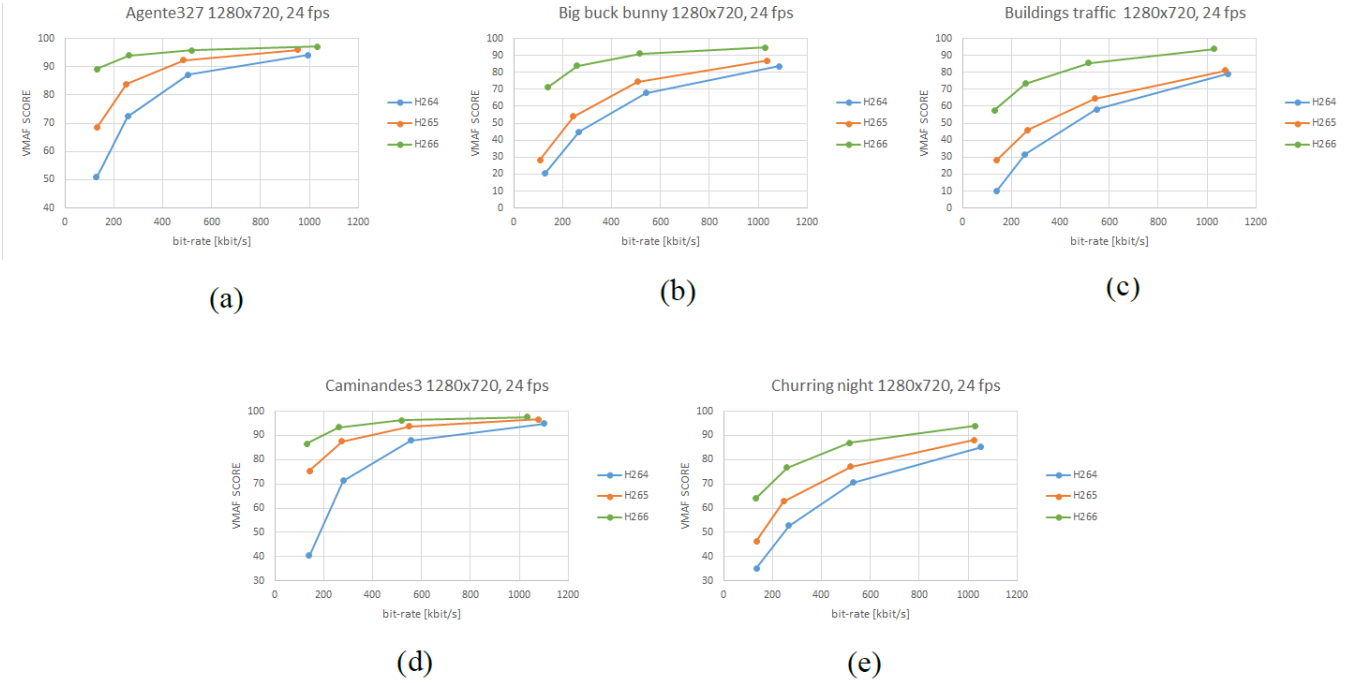

Figura 18. Valores de VMAF para secuencias [1-5] en H.264, H.265, H266

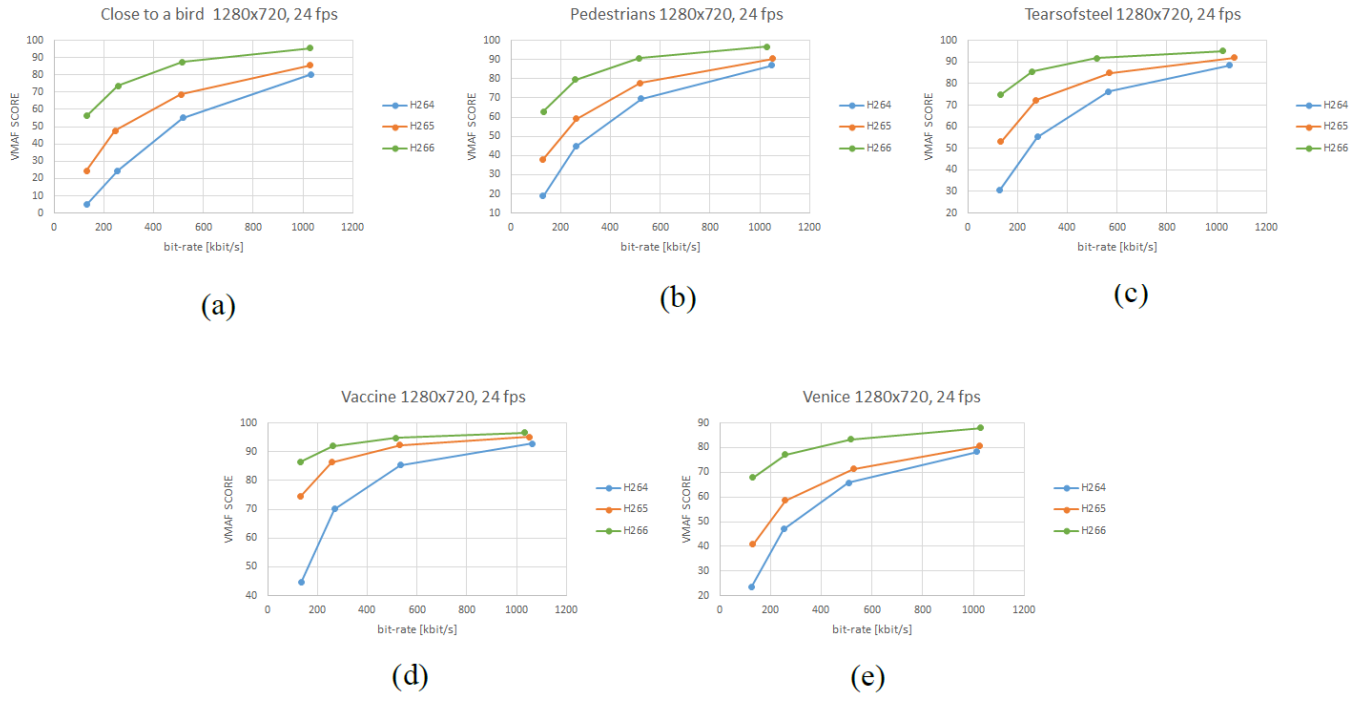

Figura 19. Valores de VMAF para secuencias [6-10] en H.264, H.265, H266

Se puede observar como para todos los codificadores, a medida que aumenta el *bitrate* el valor del VMAF aumenta. Además, se cumple que en todos los casos VVC tiene un VMAF siempre superior a HEVC y AVC.

5.3. Tiempo de codificación y otros

En las tablas 2, 3, 4, 5 se muestran para los *bitrates* de 128k, 256k, 512k y 1024k, las 10 secuencias de vídeos y los 3 codificadores los siguientes parámetros: *bitrate* real (kbps), la velocidad de codificación (Ej. 0,5x), tiempo de codificación (duración vídeo/velocidad codificación), tamaño del fichero original (KB), tamaño fichero codificado (KB), ratio de compresión (tamaño fichero original/ tamaño fichero codificado) y media de *frames* Intra (I).

Video	Codificador	Bitrate real (kbps)	Velocidad codificación	Tiempo codificación	Tamaño inicial del fichero	Tamaño final del fichero	Ratio compresión	Average QP (frames I)	PSNR
Agente327	H.264	132,70	11,2000	2,68	5014	484	10,36	43,77	31,55
	H.265	136,10	3,6700	8,17	5014	497	10,09	48,15	35,51
	H.266	130,79	0,0109	2744,97	5014	479	10,47	16,33	41,83
Big Buck Bunny	H.264	132,60	10,4000	2,88	6587	485	13,58	46,00	27,62
	H.265	126,70	1,4200	21,13	6587	390	16,89	45,79	28,24
	H.266	130,29	0,0116	2590,66	6587	478	13,78	37,13	34,24
Building traffic	H.264	142,20	9,3900	3,19	11069	521	21,25	49,90	23,28
	H.265	141,80	1,3700	21,90	11069	519	21,33	45,92	25,20
	H.266	129,96	0,0107	2812,48	11069	478	23,16	41,78	29,24
Caminandes 3	H.264	143,10	12,0000	2,50	3446	524	6,58	40,87	34,04
	H.265	186,50	1,2000	25,00	3446	682	5,05	33,53	40,26
	H.266	130,89	0,0112	2673,84	3446	481	7,16	30,82	43,00
Churring night	H.264	136,80	1,5700	19,11	7924	500	15,85	43,40	29,89
	H.265	133,80	1,3500	22,22	7924	489	16,20	43,25	31,33
	H.266	130,19	0,0111	2696,71	7924	478	16,58	38,65	34,38
Close to bird	H.264	132,80	9,8700	3,04	9532	486	19,61	50,86	20,37
	H.265	125,20	1,3100	22,90	9532	451	21,14	48,93	24,85
	H.266	130,27	0,0135	2220,83	9532	479	19,90	41,08	29,76
Pedestrians	H.264	130,80	10,8000	2,78	13158	499	26,37	50,10	25,26
	H.265	129,00	1,3400	22,39	13158	492	26,74	46,68	27,70
	H.266	130,08	0,0100	3011,61	13158	498	26,42	41,76	32,11
Tearsofsteel	H.264	142,10	11,5000	2,61	5912	520	11,37	44,79	29,43
	H.265	156,50	1,3300	22,56	5912	573	10,32	40,17	32,73
	H.266	130,23	0,0115	2613,75	5912	479	12,34	34,37	36,39
Vaccine	H.264	136,90	11,2000	2,68	3191	503	6,34	40,04	35,66
	H.265	156,10	1,1900	25,21	3191	577	5,53	34,01	40,91
	H.266	132,51	0,0108	2783,56	3191	488	6,54	29,51	44,47
Venice	H.264	128,50	10,3000	2,91	9142	476	19,21	43,84	29,16
	H.265	124,90	1,4100	21,28	9142	433	21,11	41,39	30,81
	H.266	130,51	0,0095	3161,81	9142	485	18,85	36,88	34,42

Tabla 2: Resultados de codificación para *bitrate* de 128k

Video	Codificador	Bitrate real (kbps)	Velocidad codificación	Tiempo codificación	Tamaño inicial del fichero	Tamaño final del fichero	Ratio compresión	Average QP (frames I)	PSNR
Agente327	H.264	259,70	10,0000	3,00	5014	948	5,29	36,63	35,77
	H.265	257,30	3,2900	9,12	5014	939	5,34	36,51	39,26
	H.266	260,75	0,0085	3539,46	5014	955	5,25	31,38	44,55
Big Buck Bunny	H.264	271,10	9,4900	3,16	6587	990	6,65	39,76	30,56
	H.265	239,80	1,1800	25,42	6587	876	7,52	39,53	31,77
	H.266	258,96	0,0089	3361,37	6587	950	6,93	31,95	37,46
Building traffic	H.264	279,60	0,7600	39,47	11069	1023	10,82	44,44	25,65
	H.265	286,30	1,1500	26,09	11069	1048	10,56	41,01	27,61
	H.266	258,38	0,0085	3530,76	11069	949	11,66	36,47	31,97
Caminandes 3	H.264	283,50	10,7000	2,80	3446	1037	3,32	32,35	39,15
	H.265	291,40	1,0300	29,13	3446	1310	2,63	27,36	43,70
	H.266	261,05	0,0086	3470,35	3446	959	3,59	24,69	46,13
Churring night	H.264	269,60	10,2000	2,94	7924	985	8,04	38,35	32,45
	H.265	270,90	1,1200	26,79	7924	990	8,00	38,06	34,09
	H.266	258,50	0,0086	3473,45	7924	948	8,36	33,04	37,02
Close to bird	H.264	259,20	8,5600	3,50	9532	948	10,05	48,93	24,99
	H.265	257,20	1,0200	29,41	9532	941	10,13	42,85	28,57
	H.266	258,28	0,0102	2947,68	9532	949	10,04	36,00	33,20
Pedestrians	H.264	264,50	10,0000	3,00	13158	1008	13,05	43,05	28,89
	H.265	263,10	1,1100	27,03	13158	1005	13,09	40,37	31,37
	H.266	258,87	0,0077	3902,81	13158	991	13,28	35,49	35,87
Tearsofsteel	H.264	284,50	10,5000	2,86	5912	1041	5,68	38,15	33,03
	H.265	310,30	1,1200	26,79	5912	1154	5,12	34,05	36,33
	H.266	259,64	0,0089	3374,37	5912	956	6,18	29,12	39,79
Vaccine	H.264	271,60	9,8800	3,04	3191	997	3,20	33,08	40,17
	H.265	295,90	0,9840	30,49	3191	1148	2,78	27,57	44,51
	H.266	260,32	0,0079	3786,23	3191	959	3,33	21,58	47,06
Venice	H.264	257,10	9,4500	3,17	9142	952	9,60	38,05	31,38
	H.265	261,60	1,2200	24,59	9142	900	10,16	36,43	33,03
	H.266	260,10	0,0081	3717,31	9142	962	9,50	31,29	36,04

Tabla 3: Resultados de codificación para *bitrate* de 256k

Video	Codificador	Bitrate real (kbps)	Velocidad codificación	Tiempo codificación	Tamaño inicial del fichero	Tamaño final del fichero	Ratio compresión	Average QP (frames I)	PSNR
Agente327	H.264	508,90	8,6200	3,48	5014	1856	2,70	34,96	39,90
	H.265	492,20	2,8500	10,53	5014	1796	2,79	29,72	42,74
	H.266	518,27	0,0068	4442,01	5014	1898	2,64	20,04	46,90
Big Buck Bunny	H.264	546,30	8,3500	3,59	6587	1995	3,30	33,75	33,95
	H.265	530,30	0,9980	30,06	6587	1853	3,55	33,21	35,52
	H.266	513,31	0,0074	4042,21	6587	1894	3,48	26,34	40,72
Building traffic	H.264	553,60	7,7000	3,90	11069	2025	5,47	38,03	29,19
	H.265	563,60	0,9400	31,91	11069	2061	5,37	36,13	30,35
	H.266	515,81	0,0070	4264,56	11069	1895	5,84	30,95	34,97
Caminandes 3	H.264	663,40	0,8900	33,71	3446	2055	1,68	21,51	43,48
	H.265	561,70	9,1400	3,28	3446	2509	1,37	25,02	46,80
	H.266	517,87	0,0069	4371,04	3446	1902	1,81	19,43	48,63
Churring night	H.264	533,40	8,8000	3,41	7924	1948	4,07	33,27	35,41
	H.265	540,30	0,9180	32,68	7924	1974	4,01	32,89	37,00
	H.266	516,04	0,0070	4286,78	7924	1893	4,19	27,82	39,93
Close to bird	H.264	522,40	7,6300	3,93	9532	1911	4,99	40,92	29,61
	H.265	505,10	0,8300	36,14	9532	1841	5,18	37,15	32,17
	H.266	515,17	0,0082	3660,69	9532	1892	5,04	30,65	36,58
Pedestrians	H.264	528,50	8,9800	3,34	13158	2014	6,53	36,02	33,21
	H.265	512,50	0,9010	33,30	13158	2007	6,56	34,10	35,38
	H.266	515,83	0,0062	4827,07	13158	1973	6,67	29,71	39,56
Tearsofsteel	H.264	571,40	9,1400	3,28	5912	2090	2,83	31,47	36,97
	H.265	587,80	0,9170	32,72	5912	2310	2,56	28,17	39,79
	H.266	517,98	0,0070	4272,64	5912	1903	3,11	23,12	42,57
Vaccine	H.264	538,70	8,1000	3,70	3191	1976	1,61	26,31	44,16
	H.265	587,20	0,8020	37,41	3191	2245	1,42	21,86	47,19
	H.266	516,45	0,0063	4773,41	3191	1902	1,68	17,81	49,03
Venice	H.264	513,90	8,3300	3,60	9142	1904	4,80	32,75	33,70
	H.265	497,10	1,0100	29,70	9142	1814	5,04	32,31	35,18
	H.266	515,31	0,0073	4132,51	9142	1917	4,77	28,65	37,82

Tabla 4: Resultados de codificación para *bitrate* de 512k

Video	Codificador	Bitrate real (kbps)	Velocidad codificación	Tiempo codificación	Tamaño fichero original	Tamaño fichero codificado	Ratio compresión	Average QP (frames I)
Agente327	H.264	995,27	7,1200	4,21	5014	3654	1,37	23,17
	H.265	951,82	2,2600	13,26	5014	3502	1,43	21,57
	H.266	1032,66	0,0059	5029,00	5014	3782	1,33	15,56
Big Buck Bunny	H.264	1085,57	5,8400	5,14	6587	3990	1,65	27,59
	H.265	1037,20	0,8020	37,41	6587	3788	1,74	27,02
	H.266	1029,82	0,0064	4716,65	6587	3777	1,74	20,47
Building traffic	H.264	1086,40	6,1600	4,87	11069	3999	2,77	31,80
	H.265	1083,10	0,7440	40,32	11069	4053	2,73	30,93
	H.266	1028,14	0,0062	4841,30	11069	3776	2,93	25,04
Caminandes 3	H.264	1110,60	7,6300	3,93	3446	4062	0,85	18,71
	H.265	1305,10	0,7270	41,27	3446	4818	0,72	15,98
	H.266	1032,76	0,0060	4984,60	3446	3793	0,91	14,69
Churring night	H.264	1058,40	7,0500	4,26	7924	3866	2,05	27,66
	H.265	1069,60	0,7540	39,79	7924	3907	2,03	27,33
	H.266	1030,05	0,0061	4921,19	7924	3778	2,10	22,04
Close to bird	H.264	1039,50	6,2500	4,80	9532	3802	2,51	33,54
	H.265	997,30	0,6700	44,78	9532	3648	2,61	31,16
	H.266	1026,77	0,0068	4381,17	9532	3771	2,53	25,26
Pedestrians	H.264	1055,50	7,2800	4,12	13158	4022	3,27	28,97
	H.265	1053,00	0,7150	41,96	13158	4012	3,28	27,55
	H.266	1027,85	0,0054	5590,67	13158	3932	3,35	21,31
Tearsofsteel	H.264	1142,50	7,5200	3,99	5912	4179	1,41	25,08
	H.265	1163,70	0,7420	40,43	5912	4510	1,31	22,64
	H.266	1023,77	0,0059	5081,23	5912	3760	1,57	16,15
Vaccine	H.264	1068,80	6,2500	4,80	3191	3920	0,81	19,90
	H.265	1173,90	0,6380	47,02	3191	4410	0,72	16,51
	H.266	1031,82	0,0054	5549,09	3191	3800	0,84	12,61
Venice	H.264	1021,10	7,2800	4,12	9142	3781	2,42	28,10
	H.265	960,80	0,8090	37,08	9142	3596	2,54	28,07
	H.266	1029,04	0,0062	4849,09	9142	3827	2,39	22,51

Tabla 5: Resultados de codificación para *bitrate* de 1024k

En media, el tiempo de codificación de VVC es de 128,5 mayor que HEVC y 1121,1 veces mayor que el de AVC. Respecto al ratio de compresión, este es del mismo orden para los 3 codificadores. En cuanto al *quantizer* o QP aplicado a las tramas I en VVC es siempre menor que en HEVC y AVC indicando menor reducción de la calidad visual en el nuevo codificador VVC. Por último, el tiempo de codificación total en VVC ha sido de 43 horas, 10 minutos y 30 segundos.

6. Conclusiones y trabajo futuro

Tras realizar un estudio partiendo de 10 secuencias de vídeo con diferente información espacial (SI) y temporal (TI), codificación a cuatro *bitrates* diferentes (128k, 256k, 512k y 1024k) y obtención de las métricas PSNR, SSIM y VMAF para los codificadores AVC/H.264, HEVC/H.265 y AVC/H.266 se ha podido llegar a las siguientes conclusiones:

- El nuevo codificador VVC/H.264 ofrece para el mismo *bitrate*, una calidad visual de los vídeos notablemente superior. Esto se ha podido comprobar en todas las gráficas de las métricas de PSNR, SSIM y VMAF.
- La mejora de calidad visual es más significativa cuando el valor de *bitrate* es más bajo. Esto se ha podido corroborar gracias a la métrica de VMAF puesto que muestra como para el mismo bitrate de 128k la mejora de calidad visual es mucho mayor que para *bitrates* de 256k, 512k y 1024k. En métricas como PSNR y SSIM esta característica se observa, aunque no es tan pronunciada puesto que no tienen en cuenta la evaluación del usuario.
- El tiempo de codificación de VVC/H.264 con la implementación de Fraunhofer (Versatile Vídeo Encoder; VVenC) es mucho mayor que el de HEVC/H.265 y AVC/H.264. En concreto, VVC tarda 128,5 veces más que HEVC y 1121,1 veces más que AVC en codificar el mismo vídeo.

Teniendo en cuenta los resultados obtenidos se puede concluir que VVC ofrece una mejor calidad visual para el mismo *bitrate* o bien un *bitrate* menor para la misma calidad que los codificadores de generaciones anteriores.

Algunos de los inconvenientes que tiene son que no tiene soporte en *ffmpeg* con lo que existe baja compatibilidad con los *players* de hoy en día y que a pesar de que la implementación de Fraunhofer (VVenC) ofrece mejores prestaciones que los modelos de prueba de VVC (VTM) el tiempo de codificación sigue siendo muy elevado comparado con los codificadores HEVC y AVC en *ffmpeg*. Teniendo en cuenta estos dos factores, la incapacidad de reproducción de contenido en VVC y su alto tiempo de codificación resulta inviable la incorporación en sistemas reales en un futuro muy próximo.

Respecto al trabajo futuro, se propone la realización del mismo estudio cuando exista soporte del codificador VVC/H.266 en la herramienta de *ffmpeg*. De esta manera se podrá evaluar las prestaciones del nuevo codificador con respecto a AVC y HEVC bajo la misma herramienta y las mismas condiciones.

A. Bibliografía

[1] Ticao Zhang and Shiwen Mao Dept. Electrical & Computer Engineering, Auburn University "AN OVERVIEW OF EMERGING VÍDEO CODING STANDARDS" doi: 10.1145/3325867.3325873

[2] Javier Ortiz. Encoder de vídeo, Concepto de Vídeo. Fuente: javierortiz.mx/glosario/conceptos-vídeo-online/encoder-de-vídeo/

[3] J. C. Guerri Cebollada, "QoE y codificación avanzada de vídeo." 2020

[4] V. Sze and M. Budagavi, "High Throughput CABAC Entropy Coding in HEVC," in *IEEE Transactions on Circuits and Systems for Vídeo Technology*, vol. 22, no. 12, pp. 1778-1791, Dec. 2012, doi: 10.1109/TCSVT.2012.2221526

[5] S. K. Kwon, A. Tamhankar, and K. R. Rao, "Overview of H.264/MPEG-4 part 10," J. Vis. Commun. Image Represent., vol. 17, no. 2, pp. 186–216, 2006, doi: 10.1016/j.jvcir.2005.05.010

[6] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency vídeo coding (HEVC) standard," IEEE Trans. Circuits Syst. Vídeo Technol., vol. 22, no. 12, pp. 1649–1668, 2012, doi: 10.1109/TCSVT.2012.2221191

[7] D. Mukherjee et al., "The latest open-source vídeo codec VP9 - An overview and preliminary results," 2013 Pict. Coding Symp. PCS 2013 - Proc., pp. 390–393, 2013, doi: 10.1109/PCS.2013.6737765

[8] MPEG-5 part 2 LCEVC (Low Complexity Enhancement Vídeo Coding). Fuente: https://www.lcevc.org/

[9] B. Bross *et al*., "Overview of the Versatile Vídeo Coding (VVC) Standard and its Applications," in *IEEE Transactions on Circuits and Systems for Vídeo Technology*, doi: 10.1109/TCSVT.2021.3101953

[10] N. Sidaty, W. Hamidouche, O. Déforges, P. Philippe and J. Fournier, "Compression Performance of the Versatile Vídeo Coding: HD and UHD Visual Quality Monitoring," *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1-5, doi: 10.1109/PCS48520.2019.8954562.

[11] Dmitriy Teplyakov, "What is VVC (Versatile Vídeo Coding)? Overview and Comparison with HEVC". Fuente: https://ottverse.com/what-is-vvc-h266-versatile-vídeo-coding-compare-hevc/#Spatial_Block_Prediction_in_VVC

[12] B. Bross, J. Chen, J. -R. Ohm, G. J. Sullivan and Y. -K. Wang, "Developments in International Vídeo Coding Standardization After AVC, With an Overview of Versatile Vídeo Coding (VVC)," in *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1463-1493, Sept. 2021, doi: 10.1109/JPROC.2020.3043399.

[13] Battista, S.; Conti, M.; Orcioni, S. Methodology for Modeling and Comparing Vídeo Codecs: HEVC, EVC, and VVC. Electronics 2020, 9, 1579, doi: 10.3390/electronics9101579

[14] P.910 : Subjective vídeo quality assessment methods for multimedia applications. Fuente: https://www.itu.int/rec/T-REC-P.910-200804-I

[15] Q. Huynh-Thu and M. Ghanbari, “Scope of validity of PSNR in image/vídeo quality assessment,” Electron. Lett., vol. 44, no. 13, p. 800, 2008, doi:10.1049/el:20080522

[16] Agata Jung, "Comparison of Vídeo Quality Assessment Methods". Fuente: https://www.diva-portal.org/smash/get/diva2:1135305/FULLTEXT01.pdf

[17] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy and Megha Manohara, "Toward A Practical Perceptual Vídeo Quality Metric". Fuente: https://netflixtechblog.com/toward-a-practical-perceptual-vídeo-quality-metric-653f208b9652

[18] Ffmpeg, A complete, cross-platform solution to record, convert and stream audio and vídeo. Fuente: https://www.ffmpeg.org/

[19] Fraunhofer Versatile Vídeo Encoder (VVenC). Fuente: https://github.com/fraunhoferhhi/vvenc

[20] Fraunhofer Versatile Vídeo Decoder (VVdeC). Fuente: https://github.com/fraunhoferhhi/vvdec

[21] Herramienta SITI: Spatial Information / Temporal Information. Fuente: https://pypi.org/project/siti/#requirements

[22] Herramientra VQMT - Vídeo Quality Measurement Tool. Fuente: https://github.com/rolinh/VQMT

[23] Herramientra VMAF - Vídeo Multi-Method Assessment Fusion. Fuente: https://github.com/Netflix/vmaf