Acerca de codificación de video y el nuevo estándar Versatile Video Coding (VVC)

Dr. Ing. José Joskowicz
josej@fing.edu.uy
Facultad de Ingeniería, Universidad de la República, URUGUAY
Diciembre 2020

Recientemente, en julio de 2020, la Unión Internacional de Telecomunicaciones o International Telecommunication Union (ITU) aprobó un nuevo estándar de codificación de video, en la Recomendación "ITU-T H.266". Este nuevo estándar, llamado Versatile Video Coding (VVC), incluye los nuevos y últimos avances técnicos de codificación y compresión de video y ha sido diseñado con dos objetivos principales. El primero de ellos es especificar una tecnología de codificación de video con una capacidad de compresión sustancialmente superior a la de las generaciones anteriores. El segundo es que esta tecnología sea altamente versátil, y permita su uso efectivo en una gama más amplia de aplicaciones que la soportada por estándares anteriores. Algunas áreas de aplicación clave para el uso de este nuevo estándar incluyen video de ultra alta definición (UHD), con resoluciones de imágenes de 3840×2160 pixeles (4K) y 7620×4320 pixeles (8K) y una profundidad de 10 bits. El nuevo estándar también contempla la codificación de video con un alto rango dinámico (HDR) y una amplia gama de colores (WCG). Adicionalmente, permite codificar video para aplicaciones de medios inmersivos, como video omnidireccional de 360°.

Para comprender mejor estos conceptos, haremos un breve repaso de la historia reciente en codificación digital de video, las técnicas utilizadas y las novedades de este nuevo estándar VVC.

Evolución de los estándares de codificación de video

La siguiente figura ilustra en forma resumida y gráfica los formatos más conocidos en codificación de video. Los primeros desarrollos en codificación digital de esta línea de tiempo fueron realizados por el Grupo de Expertos de Imágenes en Movimiento o Moving Picture Experts Group (MPEG). Desde su establecimiento en 1988, y hasta su reciente cierre en junio de 2020, este grupo de expertos ha elaborado estándares en codificación digital de video, aportando gran parte de las ideas y técnicas que se continúan utilizando actualmente. El último códec desarrollado por el grupo MPEG es el MPEG-5, conocido como Essential Video Coding (EVS). Una variante, llamada Low Complexity Enhancement Video Coding (LC-EVS) está aún en desarrollo.

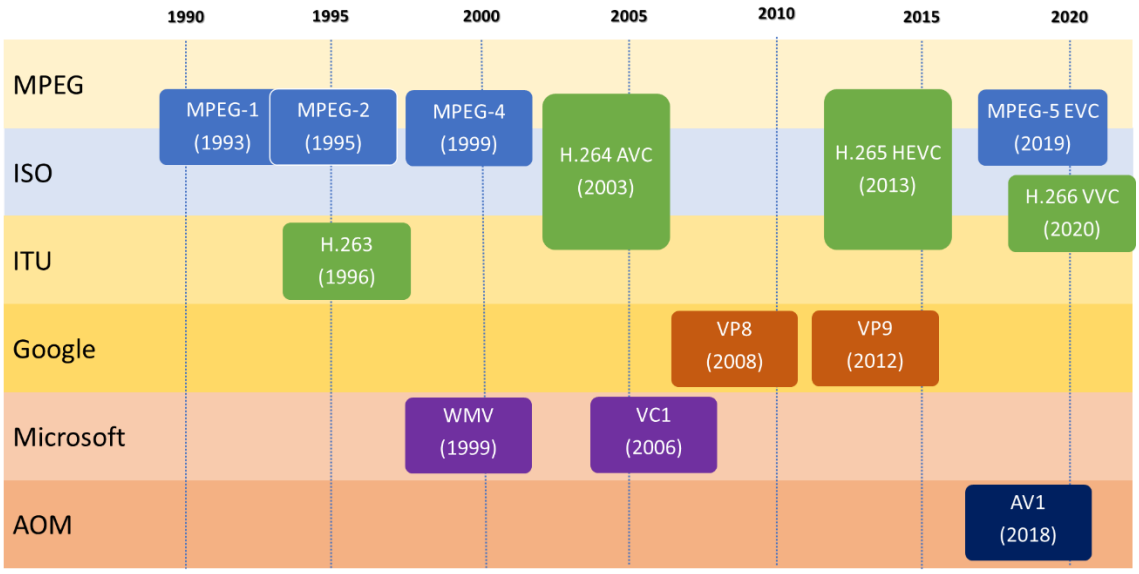
Por su parte, la ITU ha desarrollado su línea de estándares. Dentro de estos, se destaca el ITU-T H.264, conocido como Advanced Video Coding (AVC). Vale la pena mencionar que este estándar fue realizado en conjunto entre el Grupo de Expertos en Codificación de Video o Video Coding Experts Group (VCEG) que formaba parte de ITU y el MPEG, quienes formaron un equipo al que llamaron Equipo de Video Conjunto o Joint Video Team (JVT) en 2001. Si bien la primera versión de H.264 fue estandarizada en 2003, hace ya 17 años, es aun actualmente uno de los códecs más utilizado para gran variedad de aplicaciones, incluyendo broadcasting (por ejemplo, servicios de televisión digital abierta), streaming (por ejemplo, Netflix y Youtube, entre otros) y sistemas de video conferencia (por ejemplo, Zoom y Microsoft Teams, entre otros).

En 2010, ITU, a través del VCEG, y MPEG formaron un nuevo grupo al que llamaron Joint Collaborative Team on Video Coding (JCT-VC), el que tres años luego terminó con la estandarización del códec H.265, o High Efficiency Video Coding (HEVC). Si bien el códec H.265 fue estandarizado hace ya 7 años, aun son pocas las aplicaciones y sistemas que lo soportan. Una de las principales razones de ello es atribuida al complejo esquema de licenciamiento, donde hay un gran número de patentes distribuidas entre todas las organizaciones que participaron en su desarrollo. Otro aspecto ha sido la capacidad de procesamiento necesaria para el proceso de codificación. Por ejemplo, H.265 es soportado a partir de la 7ª generación de procesadores Intel, lanzado al mercado entre 2016 y 2017, entre tres y cuatro años luego de estandarizado este códec.

Muy recientemente, en julio de 2020, fue publicada la última recomendación de ITU, H.266 o Versatile Video Coding (VVC). Al igual que los casos anteriores, este desarrollo comenzó en 2017 en un trabajo en equipo, con el grupo al que llamaron Joint Video Experts Team (JVET). Siendo tan reciente, aún no hay aplicaciones comerciales disponibles que utilicen el nuevo códec VVC.

La Organización Internacional de Estándares o International Standards Organization (ISO) ha adoptado varios de las recomendaciones de MPEG y de ITU, como se muestra en la línea de tiempo.

Otras organizaciones y empresas también desarrollaron sus propios sistemas de codificación de video. Dentro de éstas se destacan el Windows Media Video (WMV) y su evolución, el VC1, desarrollado por Microsoft como alternativa al AVC de ITU. Por su parte, Google desarrolló los códecs VP8 y VP9, que se corresponden con la evolución de los códecs desarrollados por la compañía On2 Technologies, comprada por Google en 2010.

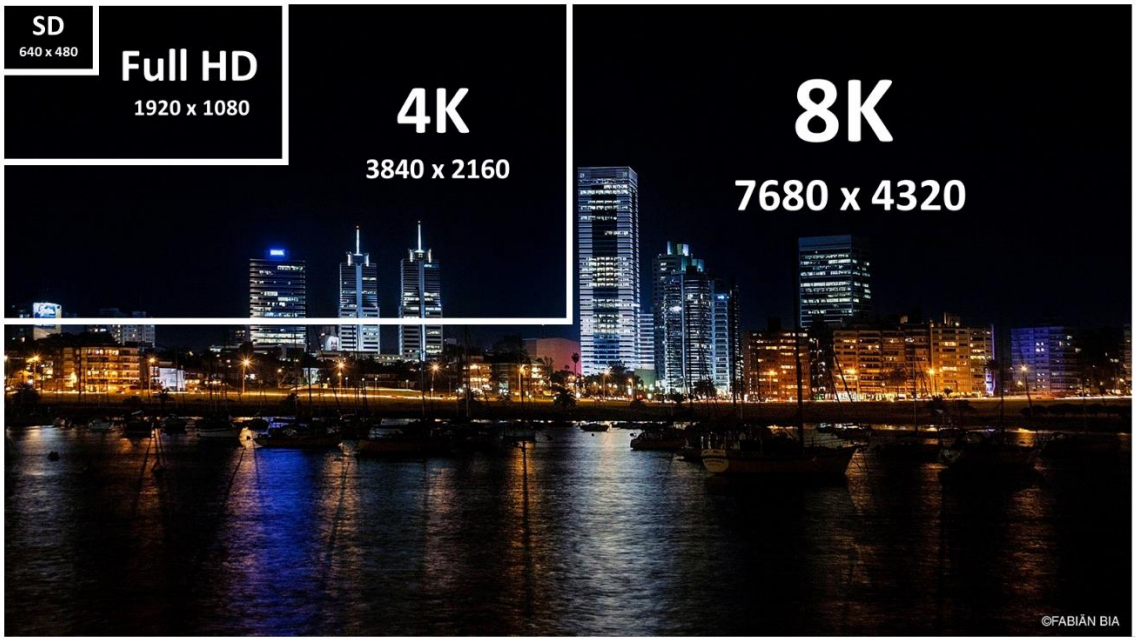


Como respuesta al lento despliegue de H.265, y al esquema de codificadores licenciados, la Alianza para Medios Abiertos (o Alliance for Open Media (AOMedia) propuso el desarrollo de un codificador de video de uso libre (al que llamó AV1), con un esfuerzo colaborativo entre varias empresas y organizaciones. AOM espera que este códec pueda competir con EVC y VVC.

Más allá de la diversidad de codificadores existentes y de las condiciones de licenciamiento, las mismas ideas básicas y técnicas subyacentes son utilizadas por todos los codificadores. Varias de ellas serán descritas a continuación.

Resolución

Las imágenes se forman en una pantalla mediante el encendido de pequeños rectángulos luminosos, llamados píxeles. La *resolución* se obtiene como el número de píxeles que puede ser mostrado en la pantalla, a lo ancho y a lo alto, y se especifica típicamente como *ancho* × *alto*. En televisión, la resolución más pequeña, llamada Definición Estándar o Standard Definition (SD) se corresponde con 640×480 pixeles. La Alta Definición o High Definition (HD) llega a 1920×1080 píxeles. Le sigue la Ultra Alta Definición o Ultra High Definition (UHD) que llega a 3840×2160 píxeles. A esta resolución se la conoce también como 4K (por el valor de 3840, cercano a 4000). Finalmente, la resolución conocida como 8K tiene 7680×4320 píxeles. La siguiente figura muestra gráficamente las diferentes de resoluciones. En lo que respecta al área, la resolución 8K es aproximadamente 100 veces más grande que la SD.

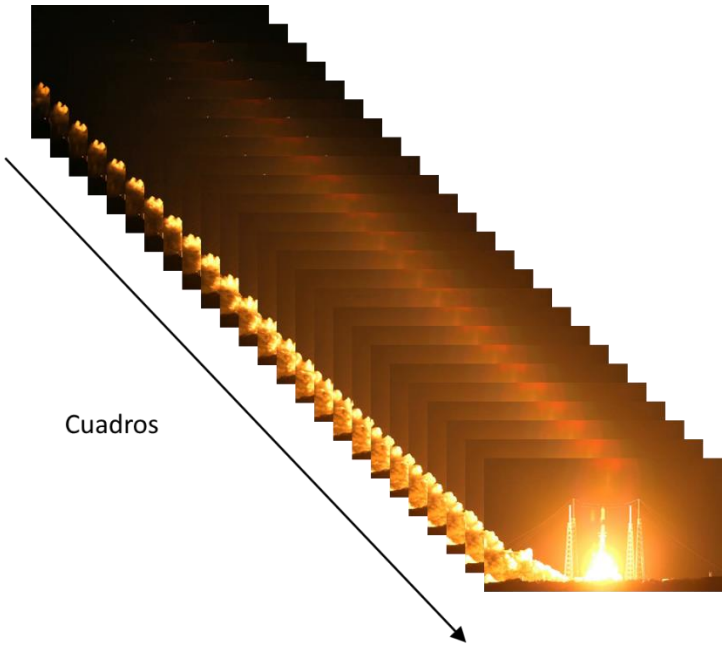


Por el momento, la resolución más popular es la HD. Mirando hacia atrás en los avances en resolución, desde el salto de SD a HD, la progresión a una imagen de mayor calidad se ha producido aproximadamente cada siete años. Recién se están introduciendo en el mercado dispositivos con resolución 4K, y el contenido y tecnología 8K aún no está disponible en el mercado, pero se prevé que cumpla con esta misma regla. El salto entre tecnologías de resolución viene asociado al salto entre los estándares de codificación. H.264 es, por ahora, el códec más popular para la distribución de contenido en HD, mientras que H.265 lo está siendo para 4K. El contenido 8K probablemente sea distribuido con el reciente códec H.266, aunque otros codificadores podrán competir por este nuevo mercado.

Cuadros por segundo

La sensación de movimiento de un video se obtiene cambiando muy rápidamente imágenes estáticas, llamadas *cuadros*. Cada imagen es una foto instantánea, tomada unos instantes luego de su foto predecesora. El sistema visual humano puede identificar imágenes individuales si son presentadas a menos de 10 o 15 cuadros por segundo. Por lo tanto, para lograr la sensación de continuidad se requieren tomar más de 15 cuadros por segundo. La televisión analógica utilizaba 25 o 30 cuadros por segundo (según la norma utilizada), y estos continúan siendo los valores

típicamente utilizados hasta el momento. Sin embargo, es posible utilizar más cuadros por segundo.



Entrelazado

Una de las técnicas utilizadas desde el origen de la televisión analógica es la conocida como *entrelazado*. Esta técnica consiste en dividir cada cuadro en dos *campos*, cada uno con la mitad de las líneas horizontales. En el primer campo se transmiten las líneas pares y en el segundo campo las líneas impares. El cuadro se compone, por lo tanto, del entrelazado de los dos cuadros, como se ve en la imagen. Esta técnica fue efectiva para la televisión analógica, ya que mejoraba la percepción de movimiento. Las imágenes de cada campo no son idénticas, ya que la imagen del segundo campo se toma un tiempo después que la del primer campo. Ésta no es una técnica apropiada para las imágenes digitalizadas, aunque se mantuvo por compatibilidad. El formato entrelazado en codificación digital se indica típicamente con una "**i**", de *interlaced*. Por ejemplo, el formato 1080**i**50 indica que se trata de una resolución de 1080 pixeles de alto (correspondiente a HD, 1920×1080), a 50 *campos* por segundo, lo que equivale a 25 cuadros por segundo. En forma alternativa, la codificación digital permite realizar la codificación de todas las líneas en cada cuadro. En este caso, se utiliza el nombre de formato *progresivo* y se indica con una "**p**". Por ejemplo, el formato 1080**p**50 indica que se trata de una resolución HD (1920×1080), a 50 *cuadros* por segundo, el doble que el anterior.

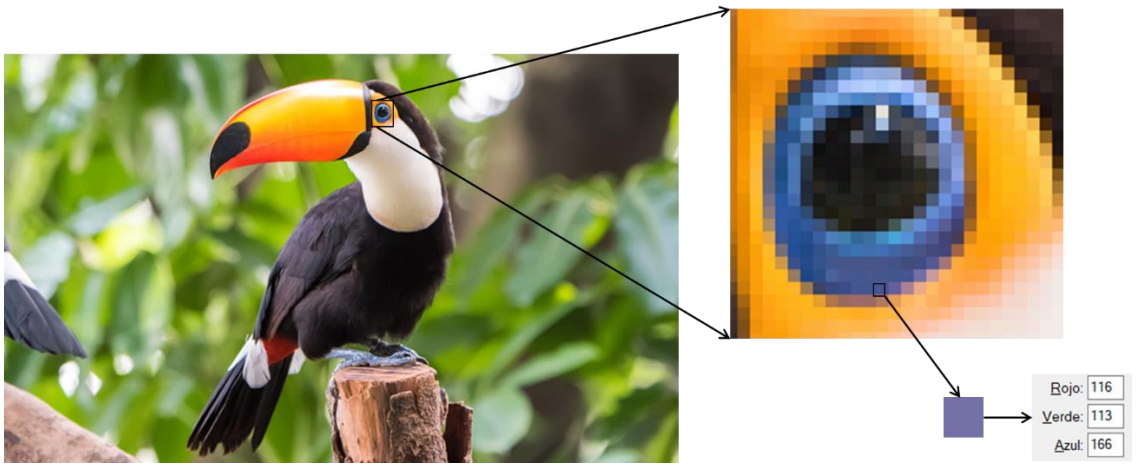
Cuando los videos tomados en forma entrelazada son transformados a un formato progresivo, pueden darse efectos adversos, como se puede ver en la siguiente figura. Como cada campo es tomado en diferentes momentos, las imágenes con movimiento aparecen con bandas horizontales. Esto puede mejorarse con filtros de desentrelazado.

Imagen original, entrelazado

Imagen con filtro de desentrelazado

Ancho de banda

¿Cuánta información por segundo es necesario enviar para la transmisión de video? Cada píxel de cada imagen se puede asociar a 3 valores, correspondientes a la intensidad de los tres colores básicos (en este caso, se utilizan el Rojo, Verde y Azul). Cada uno de estos valores se puede codificar con un mínimo de 8 bits (1 byte), lo que corresponde a 256 valores posibles. O sea, cada píxel se puede codificar con 3 bytes.



En una resolución HD, cada cuadro de imagen contiene 1920×1080 píxeles, lo que lleva a 6 220 880 bytes por cuadro (aproximadamente 6 MBytes por cuadro). Considerando que en televisión digital es habitual enviar 30 cuadros por segundo, la cantidad de información a transmitir llega a 1,4 Gbits/s.

Un servicio típico de internet fijo tiene unos 100 Mbits/s "de bajada", lo que indicaría que un flujo multimedia en HD no podría ser recibido a través de un servicio estándar de Internet. Sin embargo, es habitual que varias personas a la vez estén accediendo a contenido de video en HD simultáneamente, desde nuestros hogares. ¿Cómo puede ser esto posible? La respuesta está en el proceso de codificación de video. Estos procesos hacen uso de varios aspectos que permiten descartar información que resulta irrelevante al sistema visual humano, y que serán descritos en las siguientes secciones. El resultado final logra tasas de bits mucho más bajas que las calculadas en el párrafo anterior. La diferencia básica entre cada generación de codificadores de video es, justamente, el ancho de banda requerido para una resolución y calidad de imagen determinada. Típicamente, cada nueva generación de códecs reduce a la mitad el ancho de banda requerido respecto a la generación anterior, para una misma resolución y calidad de imagen.

Precepción de luz y color

El sistema visual humano es mucho más sensible a la intensidad de luz que a su color. Las células denominadas "conos" son los responsables de la visión del color dentro del ojo humano. Hay tres tipos de conos, sensibles a los colores rojo, verde y azul, respectivamente. Las células denominadas "bastones" son las responsables de la visión de la luminosidad general (intensidad de la luz), y no son sensibles al color. Uno ojo típico tiene del orden de 100 millones de bastones y solo 6 millones de conos. Como consecuencia, el sistema visual humano es mucho más sensible a la intensidad de luz que a su color. Los codificadores de video hacen uso de esta característica, y pueden codificar la información de color (o *crominancia*, como se llama en la jerga) con menos resolución que la información de iluminación (o *luminancia*, como se llama en la jerga).

Como se mencionó anteriormente, cada píxel de cada imagen se puede asociar a 3 valores, correspondientes a la intensidad de los tres colores básicos (Rojo, Verde y Azul). La intensidad total que observa el sistema visual humano se corresponde con la suma ponderada de las intensidades de cada color, según la siguiente fórmula:

$$Y = Y_r R + Y_g G + Y_b B$$

donde Y representa el valor de luminancia percibido por el sistema visual, R representa la intensidad del Rojo (Red), G la intensidad del verde (Green) y B la intensidad del Azul (Blue). Los coeficientes Y_r, Y_g, Y_b son fijos y cumplen $Y_r + Y_g + Y_b = 1$. Estos coeficientes reflejan el hecho que el sistema visual percibe con diferente intensidad a cada color. Los valores típicos históricos son los siguientes:

$$Y_r = 0.299$$

$$Y_g = 0.587$$

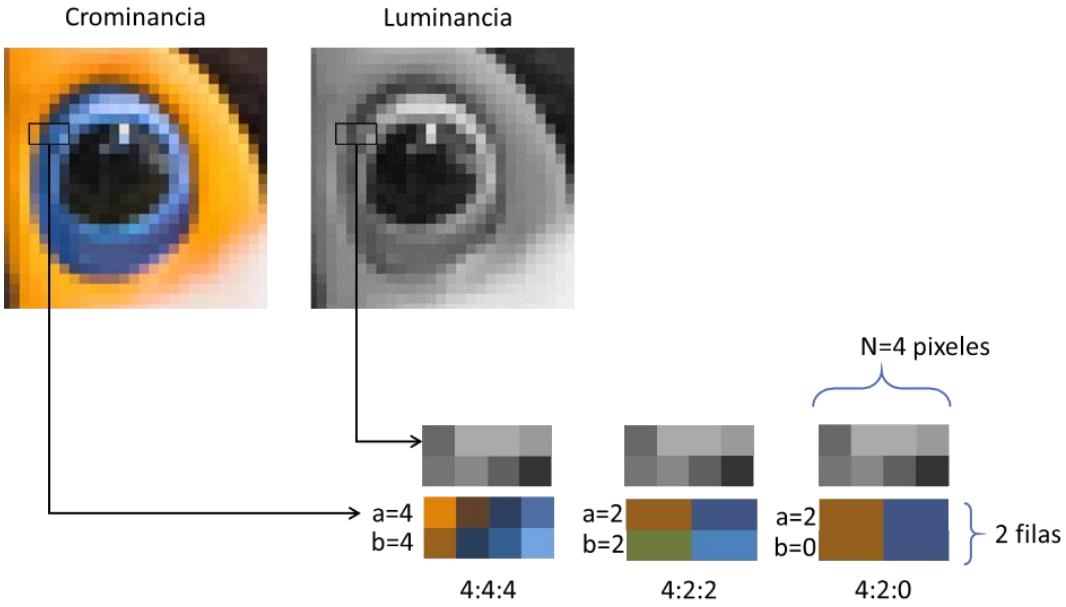
$$Y_b = 0.114$$

Dado que el sistema visual humano es mucho más sensible a la intensidad de luz que a su color, una forma alternativa de representar cada píxel es mediante los valores de Y (la luminancia), y dos nuevos valores, llamando C_r y C_b, definidos como

$$C_r = R - Y$$

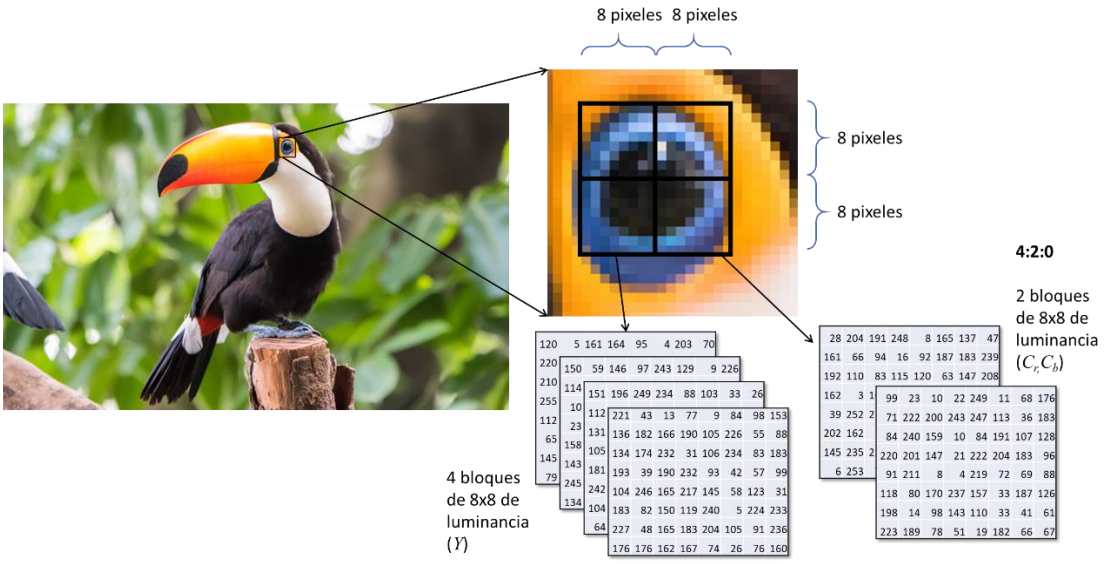
$$C_b = B - Y$$

De esta manera, cada píxel puede estar representado por la terna de valores RGB o por YC_rC_b. Ambas representaciones son en principio equivalentes, y se puede pasar de una a otra con una sencilla fórmula aritmética de conversión. Sin embargo, utilizando la representación YC_rC_b, la información de crominancia (C_r y C_b) se puede codificar con hasta 4 veces menos detalle que la de luminancia, sin que el sistema visual humano lo perciba. Esta técnica se llama submuestreo de crominancia, o "chroma subsampling". El esquema de submuestreo se expresa comúnmente como una relación de tres números, con el formato **N:a:b** (por ejemplo, 4:2:0). Esta nomenclatura describe el número de muestras de luminancia y crominancia en una región de **N** píxeles de ancho y 2 píxeles de alto. Normalmente se utiliza el valor de **N** = 4, o sea, un área de 4×2 píxeles. El valor de **a** representa la cantidad de muestras de crominancia en la primera fila (de ancho **N**) y **b** representa el número de cambios de muestras de crominancia entre la primera y la segunda fila de **N** píxeles. Por ejemplo, el formato 4:4:4 indica que se utiliza un área **4** × 2 píxeles (el primer 4), en la primera fila hay **4** (el segundo 4) muestras de crominancia, y en la segunda fila hay otras **4** (el tercer 4) muestras de crominancia (diferentes a la primera). En este caso, por cada cuatro muestra de luminancia hay una muestra de crominancia. En cambio, el formato 4:2:0 indica que se utiliza un área de **4** × 2 píxeles (el primer 4), en la primera fila hay **2** muestras de crominancia, y en la segunda fila hay **0** muestras de crominancia diferentes a la primera (o sea, se mantienen las mismas muestras de crominancia que la primera fila). En este caso, por cada cuatro muestras de luminancia hay solo una de crominancia.



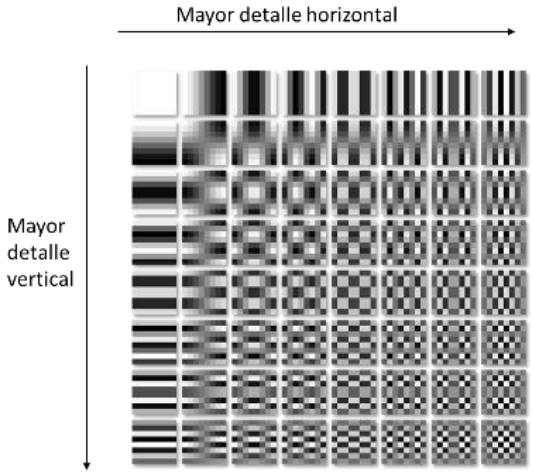
Representación eficiente de la información

Como se ha visto en las secciones anteriores, cualquier cuadro de video puede dividirse en pequeños pixeles, que pueden ser representados por su información de luminosidad y de color (o luminancia y crominancia). A los efectos de realizar una representación eficiente de la información, los cuadros de video se dividen en bloques que típicamente contienen 8×8 pixeles. Cuatro de estos bloques conforman una unidad de 16×16 pixeles llamada *macrobloque*. Utilizando un formato 4:4:4, un macrobloque se puede representar con cuatro bloques de 8×8 valores de luminancia Y, cuatro bloques de 8×8 valores de crominancia C_r y otros cuatro bloques de 8×8 valores de crominancia C_b.. En total, se requieren 4×8×8 + 4×8×8 + 4×8×8 = 768 valores. Sin embargo, utilizando un formato 4:2:0, cada macrobloque se puede representar con cuatro bloques de 8×8 valores de luminancia Y, un solo bloque de 8×8 valores de crominancia C_r y otro bloque de 8×8 valores de crominancia C_b.. En total, se requieren 4×8×8 + 8×8 + 8×8 = 384 valores, la mitad que en el caso anterior.

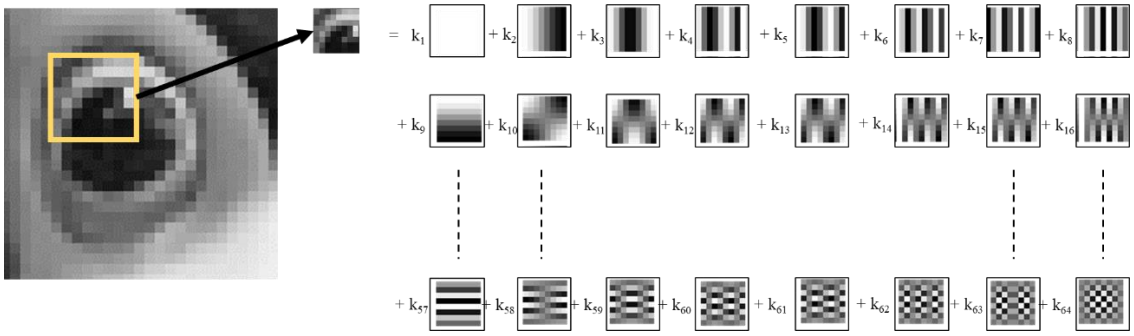


A los efectos de representar de una manera más eficiente la información, el ingeniero Nasir Ahmed propuso en un breve trabajo publicado en 1974, una técnica matemática llamada Transformada Discreta de Coseno, o Discrete Cosine Transform (DCT). El objetivo de la técnica es transformar los 64 valores de información de cada bloque por otros 64 valores, pero que de alguna manera "ocupen menos lugar". ¿Cómo puede ser esto posible?

La idea básica consiste en representar la información del bloque ordenada según su grado de *detalle visual*. A modo de ejemplo, el menor grado de detalle posible en un bloque de 8×8 es su luminancia media. Es decir, podemos representamos al bloque completo con un solo valor, que represente el promedio de todos los valores de los 64 píxeles. Esto, evidentemente, divide por 64 la información total, pero se pierde todo el detalle del contenido de los píxeles. A este valor de luminancia media se le pueden ir agregando otros valores, que vayan representando progresivamente mayor grado de *detalle visual*. La DCT propone un conjunto de bloques con *dibujos patrones*, con grado creciente de detalle visual, con la característica particular que cualquier bloque de 8×8 píxeles puede obtenerse mediante una *combinación lineal* de los 64 bloques de los dibujos patrones fijos. Estos bloques patrones se definen de manera que tengan un diseño progresivamente complejo. La siguiente figura muestra los diseños elegidos para los 64 bloques patrones (cada uno de 8×8 píxeles). Notar que, de izquierda a derecha, y de arriba hacia abajo, los bloques patrones van tomando diseños más complejos y con más "detalles".



De esta manera, el valor de cada píxel de un bloque de 8×8 se puede obtener como la suma de los valores de los píxeles correspondientes en los bloques patrón, ponderados por coeficientes k_i, como se esquematiza en la siguiente figura.



Los 64 valores de los coeficientes k_i conforman una nueva representación de los 64 píxeles. Se puede demostrar que a partir de estos 64 coeficientes k_i se puede reconstruir los 64 valores originales de los píxeles. Un ejemplo se muestra en la siguiente figura. El primer cuadro contiene los valores de luminancia de un bloque de 8×8 píxeles. El segundo cuadro contiene los valores de la transformada DCT (notar que estos valores pueden ser decimales). Los coeficientes DCT decrecen hacia la esquina inferior derecha. Esto no es casual. Si se observa la disposición de los patrones usados para obtener los coeficientes DCT, se puede ver que cuanto más cerca de la esquina inferior derecha, más detalles se agregan a la imagen. Podemos leer los valores de DCT de una manera interesante: los que tienen valores más altos primero (menor nivel de detalle), y los valores más bajos al final (los que tienen mayor nivel de detalle), en un esquema de zig-zag, como se muestra en el tercer cuadro. Esta forma de representar la información es más eficiente y útil, como se verá en la siguiente sección.

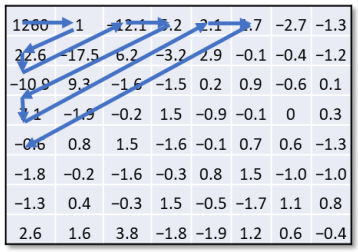
Valores originales

158	158	158	163	161	161	162	162
157	157	157	162	163	161	162	162
157	157	157	160	161	161	161	161
155	155	155	162	162	161	160	159
159	159	159	160	160	162	161	159
156	156	156	158	163	160	155	150
156	156	156	159	156	153	151	144
155	155	155	155	153	149	144	139

Valores DCT

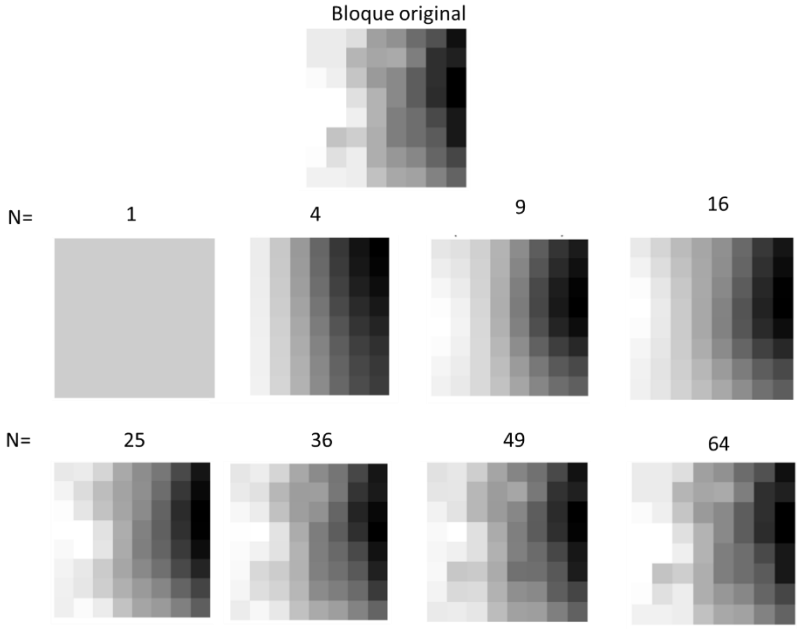
1260	1	−12.1	5.2	2.1	1.7	−2.7	−1.3
22.6	−17.5	6.2	−3.2	2.9	−0.1	−0.4	−1.2
−10.9	9.3	−1.6	−1.5	0.2	0.9	−0.6	0.1
7.1	−1.9	−0.2	1.5	−0.9	−0.1	0	0.3
−0.6	0.8	1.5	−1.6	−0.1	0.7	0.6	−1.3
−1.8	−0.2	−1.6	−0.3	0.8	1.5	−1.0	−1.0
−1.3	0.4	−0.3	1.5	−0.5	−1.7	1.1	0.8
2.6	1.6	3.8	−1.8	−1.9	1.2	0.6	−0.4

Recorrido en zig-zag

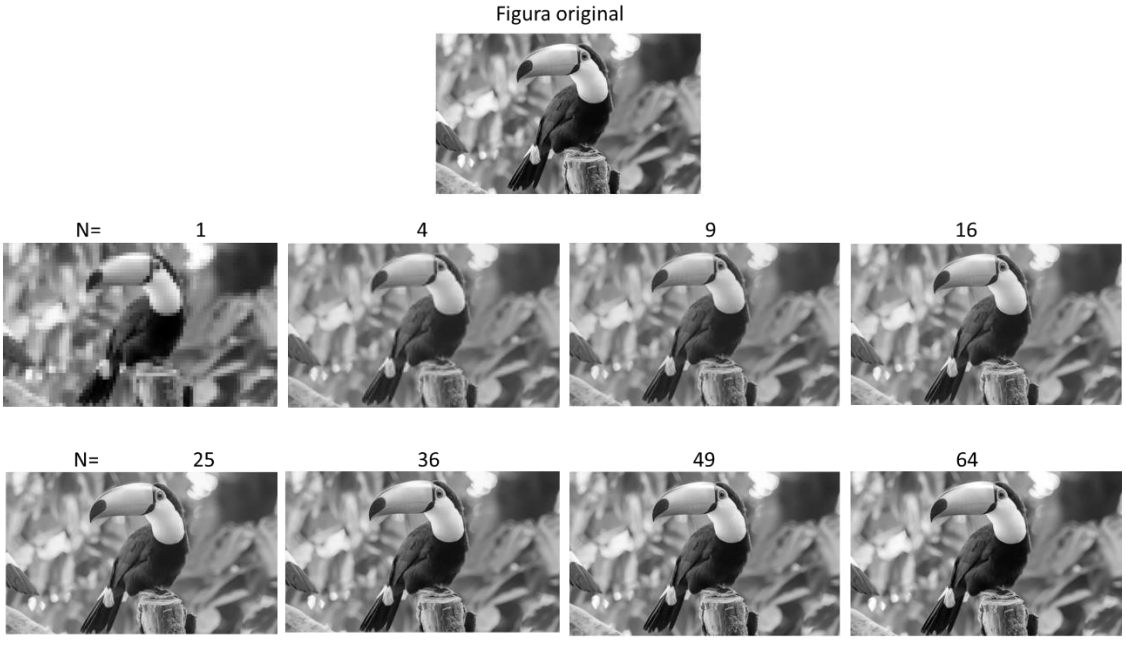


Compresión

Una manera de *comprimir* la información es quedarnos únicamente con los primeros coeficientes DCT, y descartar los últimos. Se perderá algo de detalle del bloque, pero se podría tener una representación bastante buena. Esto se esquematiza en la siguiente figura, donde se muestra un bloque de 8×8 píxeles, y el bloque que resulta de pasarlo por la DCT, quedarnos únicamente con los primeros N valores, y aplicar la DCT inversa. Se puede ver que, al quedarse con un solo valor, perdemos todos los detalles del bloque. Pero a medida que se agregan coeficientes, rápidamente se obtienen resultados cada vez más similares al original. En este ejemplo, con los primeros 25 coeficientes ya resulta difícil, en una primera mirada, distinguir el bloque reconstruido respecto del original.



Al poner todos los bloques juntos, formando la imagen, los efectos aún parecen menos perceptibles. En la siguiente figura se puede ver cómo, aún dejando el promedio de luminancia de cada bloque (o sea, manteniendo únicamente el primer valor de la DCT, o N=1 en este ejemplo), la imagen es perfectamente reconocible, aunque se ve claramente "pixelada". Al aumentar la cantidad de coeficientes DCT la imagen rápidamente gana definición.



Una técnica que permite comprimir aún más la información consiste en dividir los coeficientes DCT por un factor de escala llamado matriz de cuantización o Quantization Matrix. Los valores de escalado pueden ser diferentes para cada coeficiente (de allí que se hable de una *matriz*).

Cada coeficiente de DCT se divide por un factor, y se trunca a valores enteros, que puedan ser fácilmente representados por bytes, como se muestra en la siguiente figura de ejemplo.

Valores DCT

1260	1	−12.1	5.2	2.1	1.7	−2.7	−1.3
22.6	−17.5	6.2	−3.2	2.9	−0.1	−0.4	−1.2
−10.9	9.3	−1.6	−1.5	0.2	0.9	−0.6	0.1
7.1	−1.9	−0.2	1.5	−0.9	−0.1	0	0.3
−0.6	0.8	1.5	−1.6	−0.1	0.7	0.6	−1.3
−1.8	−0.2	−1.6	−0.3	0.8	1.5	−1.0	−1.0
−1.3	0.4	−0.3	1.5	−0.5	−1.7	1.1	0.8
2.6	1.6	3.8	−1.8	−1.9	1.2	0.6	−0.4

Valores escalados DCT

126	0	-1	1	0	0	0	0
2	-2	1	0	0	0	0	0
-1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

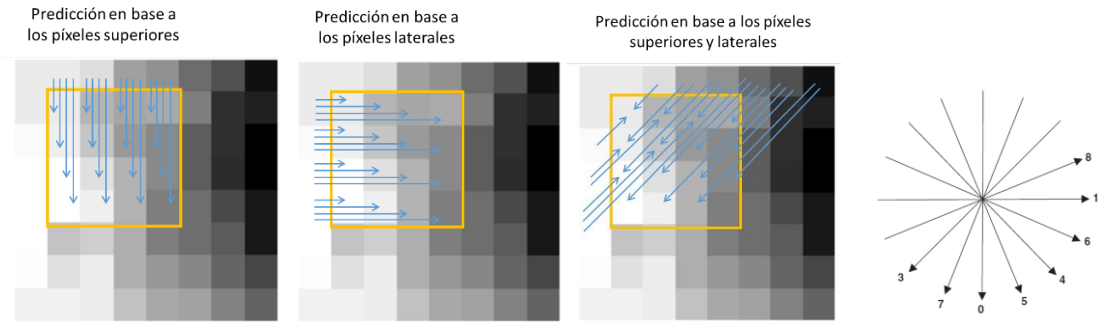
¡Notar que, en el último cuadro, solamente hay 10 valores diferentes de cero! Esto valores se concentran sobre la esquina superior izquierda. En este ejemplo, los 64 valores originales se redujeron a 10. Ajustando el factor de escala, es posible reducir aún más la cantidad de valores, a costa de perder más *detalles finos* de la imagen. Este factor de escala es conocido como el Quantization Parameter (QP). Cuanto más alto el QP, peor es la calidad de la imagen resultante (recordar que los valores se dividen entre el QP).

Una vez que se obtienen los valores finales de los coeficientes DCT, se puede aplicar una técnica de codificación entrópica o entropy coding. Esta técnica consiste en presentar los valores más frecuentes con pocos bits, y los valores menos frecuentes con más bits. Dado que el *cero* es uno de los valores más frecuentes, bastaría representarlo con un solo bit 0, y utilizar más bits para otros valores.

Combinando el submuestreo de crominancia, la transformación DCT y la codificación entrópica, se puede obtener una representación mucho más compacta de cada macrobloque de cada imagen del video.

Predicción dentro de cuadros

Las partes cercanas dentro de un mismo cuadro pueden ser muy parecidas. Por ello es posible intentar *predecir* el valor de la luminancia y crominancia de cada píxel en función de los pixeles cercanos. Las técnicas de codificación de video pueden hacer uso de esta característica. Por ejemplo, el valor de la luminancia de los píxeles dentro de un bloque puede ser predicho en función de los píxeles superiores o laterales, cómo se muestra en la figura (en este ejemplo, en un bloque de 4×4 píxeles).



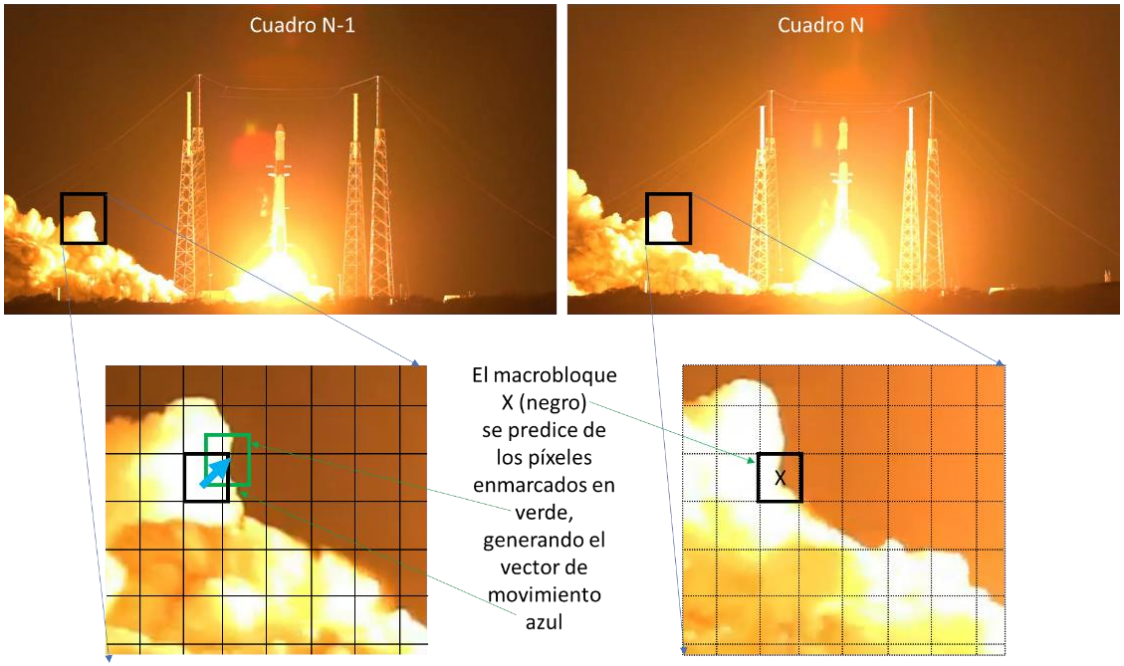
Para cada bloque se debe elegir la mejor predicción. Por ejemplo, en H.264 la predicción se puede realizar en bloques de 4×4 o 16×16 y se puede elegir entre 8 posibles direcciones, identificadas como “modo 0” a “modo 8”. El “modo 0” predice en función de los píxeles superiores, el “modo 1” en función de los píxeles laterales izquierdos y el “modo 3” en función de ambos, como se mostró en la figura anterior. Otros modos utilizan diferentes combinaciones de direcciones.

Se toma como referencia el píxel, o el promedio de los valores de otros píxeles, para predecir el valor del píxel a codificar, y se codifica la diferencia. Luego se aplican las técnicas ya descritas, de DCT, codificación entrópica, logrando aún mayores compresiones de la información.

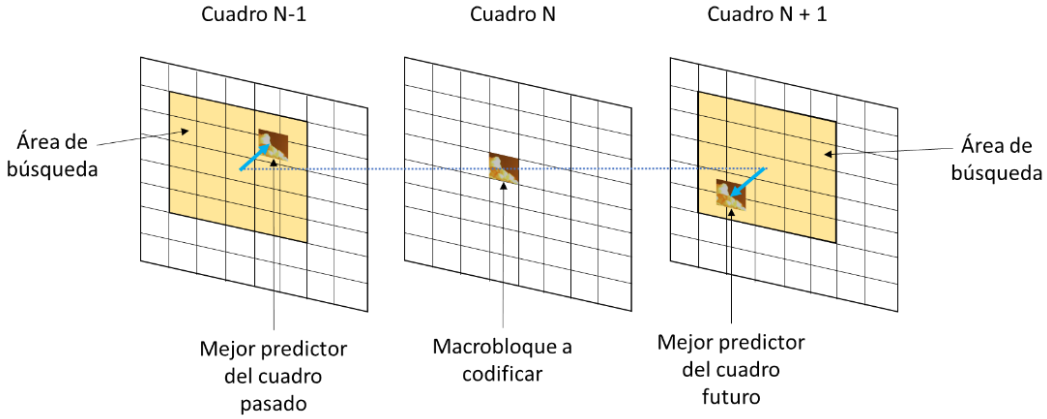
Predicción entre cuadros

Cada cuadro en un video es tomado a pocos milisegundos del cuadro anterior. Por ejemplo, si se utiliza una tasa de 25 cuadros por segundo, cada cuadro se toma cada 40 milisegundos. Es de esperar que gran parte de estos cuadros sean muy parecidos a los cuadros anteriores. Evidentemente el movimiento de los objetos hará que los cuadros no sean idénticos, pero típicamente un mismo objeto de un cuadro estará en el cuadro siguiente, en un lugar ligeramente diferente. Por lo tanto, la información de cada cuadro puede estar altamente correlacionada con los cuadros anteriores y también con los cuadros futuros (es decir, los que serán tomados en los próximos instantes). Esto permite utilizar técnicas que eliminen información redundante *temporal*. Esta vez, las técnicas tienen que ver con la estimación del movimiento o Motion Estimation (ME) y con la compensación del movimiento o Motion Comensation (MC).

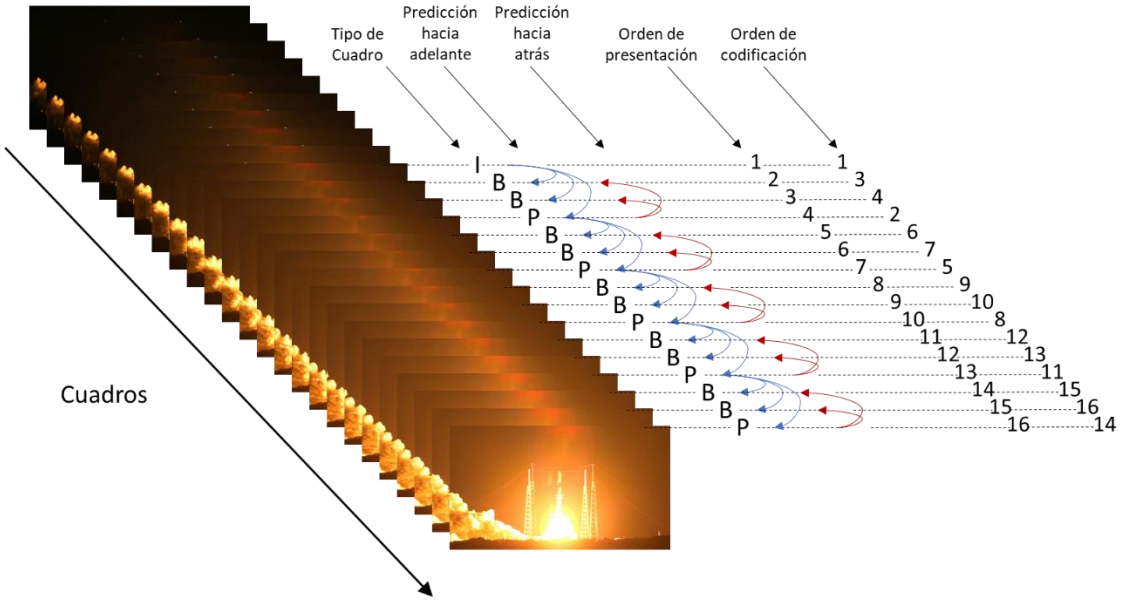
Cada cuadro se divide en bloques (en este caso típicamente se usan bloques de 16×16, llamados macrobloques, como ya se definió anteriormente). Para cada macrobloque se busca su correspondiente en el o los cuadros anteriores. Esta búsqueda se realiza dentro de una ventana de cierto tamaño, centrada en el macrobloque en cuestión. Si el codificador encuentra con éxito un bloque razonablemente coincidente, el macrobloque actual se codifica utilizando un vector de movimiento o Motion Vector (MV) que representa el movimiento entre el bloque de referencia y el actual, como se muestra en la siguiente figura. Este proceso, consistente en determinar los vectores de movimiento, se conoce como estimación de movimiento (ME). La región del macrobloque elegido se utiliza como predicción para el macrobloque actual y se resta del macrobloque actual para formar un bloque residual. Este proceso de formar un bloque residual se conoce como compensación de movimiento (MC). El bloque residual se codifica (nuevamente, se pude aplicar DCT y codificación entrópica) y se transmite junto con los vectores de movimiento que indican el origen de la predicción.



Este mismo proceso se puede refinar, prediciendo los valores de un macrobloque determinado en función de varios cuadros anteriores, pero también de varios cuadros futuros, como se esquematiza en la siguiente figura. Para que esto sea posible, el codificador debe almacenar en su memoria varios cuadros, antes de realizar su codificación.

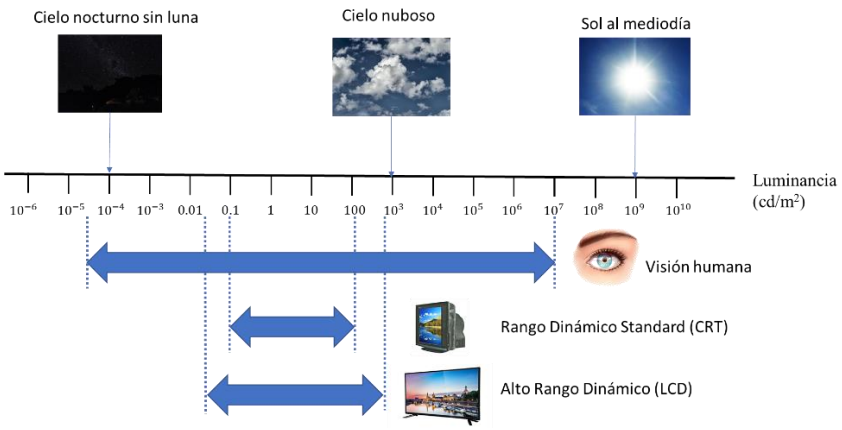


Es claro que, al inicio, se requiere enviar un cuadro completo sin predicción temporal. También puede ser más eficiente enviar cuadros sin predicción de movimiento cuando hay cambios bruscos de escena. Al utilizar las técnicas de predicción entre cuadros, los primeros codificadores MPEG introdujeron el concepto de cuadros llamados I, P y B. Un cuadro del tipo ***I*** (Intraprediction) es un cuadro que se codifica sin predicción temporal. Se puede aplicar predicción dentro del cuadro, como se explicó en la sección anterior, pero no se utilizan otros cuadros para su codificación. Un cuadro del tipo ***P*** (Predictive) utiliza cuadros I o cuadros P anteriores para su codificación. Un cuadro del tipo ***B*** (Bidirectional predictive) utiliza cuadros I o cuadros P anteriores y futuros para su codificación. La secuencia de cuadros I, B y P se conoce como Group of Pictures (GoP), y se ejemplifica en la siguiente figura.



Alto rango dinámico

El sistema visual humano puede observar y diferenciar un rango muy amplio de iluminación o luminosidad, desde un cielo nocturno en una noche cerrada, hasta un brillante día de sol en la arena. La luminosidad o intensidad luminosa se mida en *candelas* por metro cuadrado (cd/m^2). Sus medidas varían entre valores muy altos (por ejemplo 1 000 000 000 cd/m^2 para el sol al medio día) y muy pequeños (por ejemplo, 0.00001 cd/m^2 para un cielo nocturno sin luces). Cualquier técnica de codificación de imágenes o video permite diferenciar la luminosidad solo dentro de un cierto rango limitado, comprendido entre un mínimo y un máximo. Fuera de este rango, las áreas más brillantes aparecen como blanco puro y las áreas más oscuras como negro puro. La relación entre el máximo y el mínimo de los posibles valores se conoce como rango dinámico (concepto que es aplicable a cualquier tipo de medida). Históricamente los televisores y pantallas podían representar variaciones de luminosidad en un rango dinámico que oscilaba entre 0.1 y 100 cd/m^2 aproximadamente. Los actuales LCD permiten un rango dinámico mucho mayor, llegando hasta 1000 cd/m^2. Se puede ver un esquema ilustrativo en la siguiente figura (notar que la escala de luminancia es logarítmica, cada marca en la escala es 10 veces mayor que la anterior).

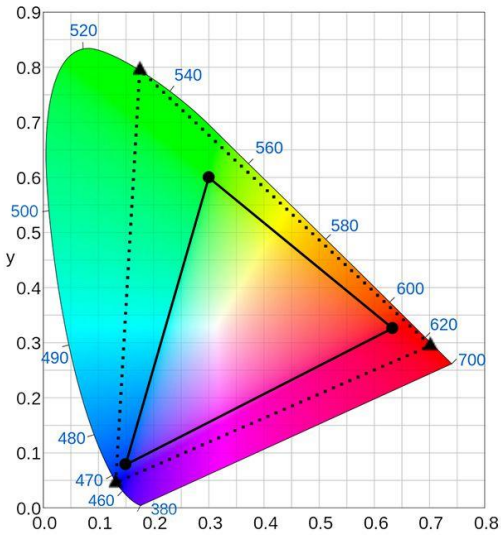


Esto permite reproducir imágenes con mayor variación de luminosidad y por tanto, brindar sensaciones visuales más reales. El video en Alto Rango Dinámico, o High Dynamic Range (HDR) consiste en capturar, codificar y presentar el contenido visual de luminancia con mayor precisión y resolución que el anterior Rango Dinámico Estándar o Standard Dynamic Range (SDR). Esto permite obtener blancos más brillantes y negros más intensos. Los nuevos estándares de codificación HDR permiten una luminancia máxima más alta y utilizan al menos 10 bits para su codificación, en comparación con 8 bits de los estándares anteriores.

Amplia gama de colores

Así como la técnica HDR permite un aumento en el rango dinámico de la iluminación, la técnica conocida como Amplia Gama de Colores o Wide Color Gamut (WCG) permite aumentar el rango dinámico de los colores, de manera que los rojos sean *más rojos*, los verdes *más verdes*, y los azules *más azules*.

Los colores son normalmente especificados con tres valores, por ejemplo, RGB o YC_rC_b, como se indicó en secciones anteriores. Si quisiéramos representar gráficamente todos los posibles colores (todas las posibles variaciones de los tres valores), se requeriría utilizar un gráfico en tres dimensiones. No es fácil imaginar cómo se ve un color en una coordenada específica dentro de un espacio tridimensional. Sin embargo, los tres valores incluyen también la luminosidad. Si nos enfocamos únicamente en el color, y estandarizamos la luminosidad, es posible eliminar una de las coordenadas, lo que permite realizar una representación gráfica en un plano de 2 dimensiones. Así surge el diagrama de cromaticidad *CIE 1931* mostrado en la siguiente figura. Este diagrama es una representación matemática y gráfica de los colores que el ojo humano es capaz de ver. No vamos a profundizar en esta sección en explicar la forma del diagrama, o el significado de sus coordenadas (x, y).

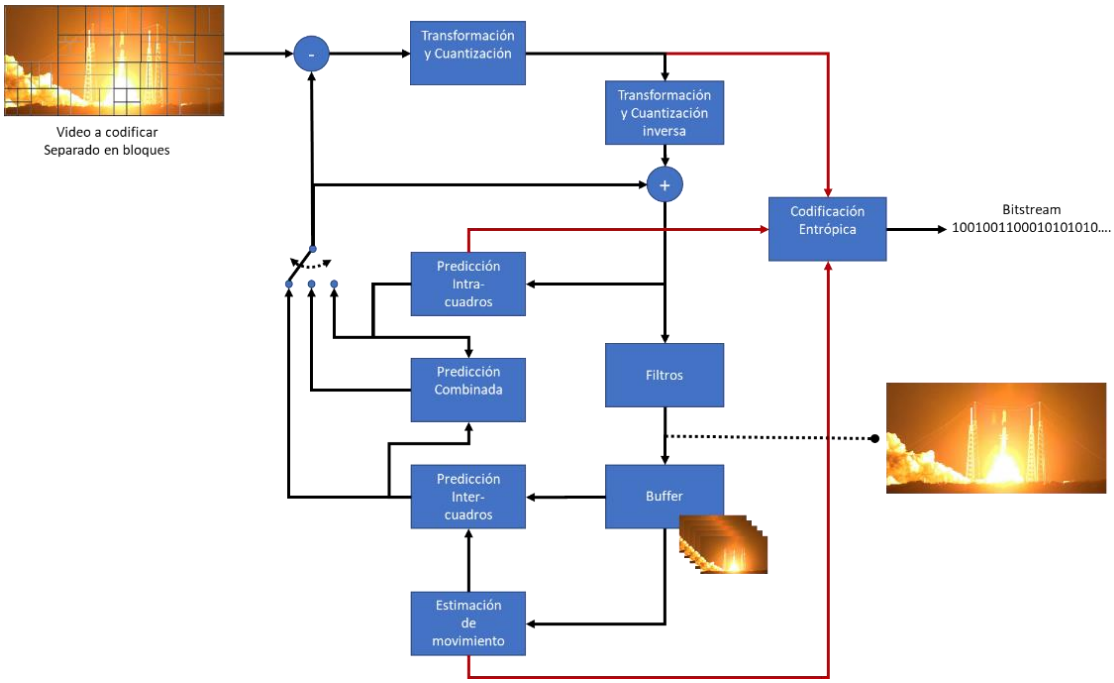


El triángulo de lados sólidos dentro de la figura indica el área de color utilizado por los monitores y televisores que soporta una gama de colores estándar (según la recomendación ITU-R BT 709). El triángulo de lados punteados indica el área de color que debe ser soportado según la recomendación ITU-R BT 2020, para resolución UHD (hasta 8K). Como se ve, se incluyen una

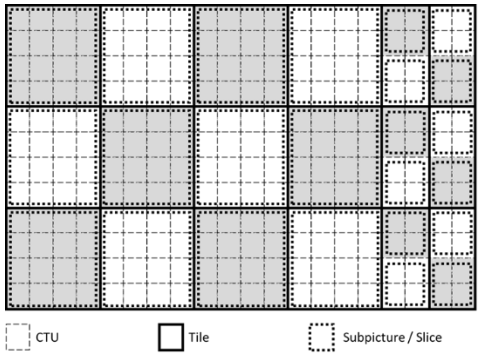
gama de colores mucho mayor. Entre ambas recomendaciones, algunos dispositivos soportan áreas intermedias, por ejemplo, según el estándar DCI-P3.

Novedades del estándar VVC

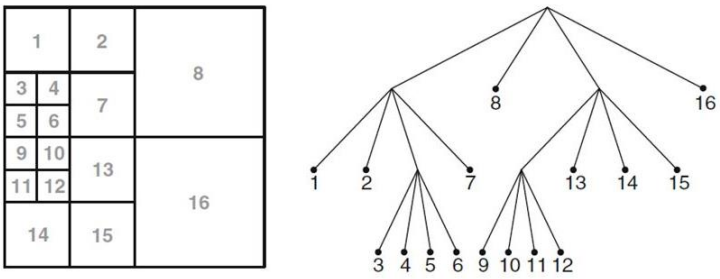
Las técnicas utilizadas en el nuevo codificador VVC son básicamente las mismas que se describieron anteriormente, pero *refinadas*. En la siguiente figura se muestra un diagrama de bloques, de alto nivel, donde se esquematiza el funcionamiento del códec de video.



En este nuevo codec, los cuadros se pueden dividir en subimágenes (subpictures), rebanadas (slices) y mosaicos (tiles). Cada mosaico (tile) se divide, a su vez, en **bloques** llamados *unidad de árbol de codificación* o Coding Tree Unit (CTU), como se muestra en la figura.



Los CTU reemplazan a los bloques de tamaño fijo de codificadores anteriores, y pueden tener subdivisiones, generando una estructura del tipo árbol, como se muestra en la siguiente figura. Cada CTU se puede codificar en forma independiente, como se describirá más adelante.

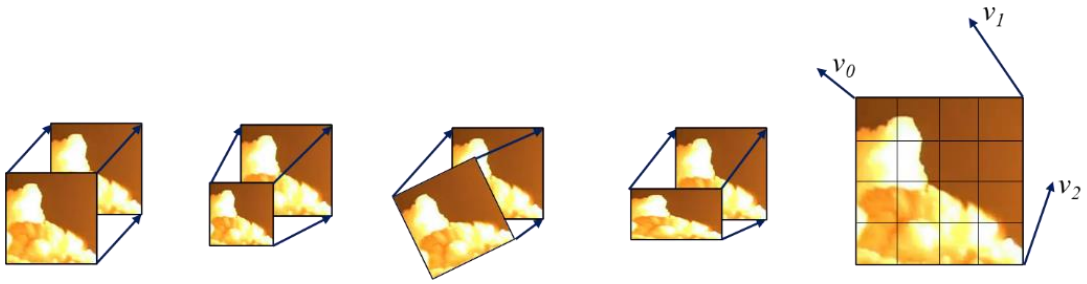


El concepto de CTU había sido introducido en el códec HEVC (H.265). En este códec, cada sub-bloque de un CTU podía ser dividido en cuatro cuadrados iguales (como se ve en la figura anterior). VVC permite definir CTUs de hasta 128×128 píxeles (en comparación con el tamaño máximo de 64×64 en HEVC), y nuevos formatos dentro del CTU. Cada CTU puede ser dividido en dos, tres o cuatro sub-bloques. Esto permite seleccionar con mayor precisión regiones donde los valores de cada sub-bloque sean muy similares, y de esta manera, mejorar las técnicas de predicción. Adicionalmente, VVC permite tener una estructura de CTU diferente para luminancia y crominancia.

El proceso de codificación puede seleccionar entre utilizar predicción dentro de cuadros, entre cuadros o una combinación de ambos. Esto se simboliza en el diagrama con la "llave" que conecta la salida de los módulos "Predicción Intra-cuadros", "Predicción Combinada" o "Predicción Inter-cuadros" con los sumadores (indicados como círculos en el diagrama). La mejor predicción del bloque se compara con el bloque original, y la diferencia es pasada al módulo "Transformación y Cuantización".

Dentro de cada sub-bloque, VVC permite seleccionar hasta 67 direcciones de predicción (en comparación con las 9 posibles direcciones permitidas en AVC (H.264), y con las 35 soportadas en HEVC (H.265). Al utilizar los sub-bloques se pueden realizar predicciones sobre formas rectangulares, además de cuadradas.

En lo que respecta a la predicción entre cuadros, VVC propone varias mejoras a los vectores de movimiento. En códecs anteriores, cada macrobloque se predecía con un único vector de movimiento, identificado por dos valores (las diferencias de coordenadas horizontales Δx y verticales Δy entre ambos bloques). VVC introduce el concepto de movimiento afín o Affine Motion. Esta técnica permite identificar rotaciones, ampliaciones e incluso deformaciones, utilizando hasta tres vectores de movimiento diferentes por cada macrobloque, como se esquematiza en la siguiente figura.



A los efectos de comprimir la información, VVC introduce la posibilidad de selección de múltiples transformadas o Multiple Transform Selection (MTS). Además de la clásica transformada DCT aplicable a bloques cuadrados, se permiten otras transformadas, basadas también en DCT o en la Discrete Sine Transform (DST), admitiendo bloques más grandes que los códecs anteriores, y de aspecto rectangular (además de cuadrado).

VVC también se extiende el rango del factor de escala o cuantización QP, permitiendo, por tanto, mayores compresiones.

En comparación con HEVC, que incluye un filtro de desbloqueo o deblocking filter y un filtro de compensación adaptativa o Sample Adaptive Offset Filter (SAO), VVC introduce un tercer filtro dentro del codificador, llamado filtro de bucle adaptativo o Adaptive loop filter (ALF). Este nuevo esquema permite seleccionar entre 25 posibles filtros para cada bloque de luminancia, según sea más conveniente. Junto con los filtros ya existentes en HEVC, se logran mejoras en las imágenes reconstruidas, lo que redunda en obtener luego mejores predictores, y por lo tanto, bajar la cantidad de información residual a transformar y cuantizar.

Con la combinación de todas las mejoras, el códec VVC introduce mejoras en la compresión de hasta un 30% respecto a su predecesor HEVC. Estas mejoras provienen principalmente de permitir estructuras de bloques más flexibles, las nuevas tecnologías en predicción intra e inter cuadros, los filtros mejorados y la posibilidad de seleccionar múltiples transformadas.