



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Máster en Ingeniería de Computadores y Redes

Trabajo Fin de Máster

ESTUDIO DE HERRAMIENTAS PARA ANALIZAR Y MEJORAR LA ROBUSTEZ DE VIDEO CODIFICADO CON HEVC

Autor: David Leonardo Pinzón Niño

Director(es): José Oliver

Septiembre 2017

Valencia

Agradecimientos

*A mi familia por su apoyo incondicional en cada uno de los caminos
que he decidido tomar.*

A mi pareja cuyas palabras siempre han sido motivación.

Y a cada profesor que ha sido orientador y día a día da lo mejor en su profesión.

RESUMEN

Durante los últimos años, la UNION INTERNACIONAL DE TELECOMUNICACIONES (en adelante UIT) ha realizado los estudios y desarrollos para crear una nueva forma de compresión de video. Por tal motivo en el 2004 se creó VCEG, para que la UIT tuviera la capacidad de mejorar la compresión para la norma H.264/MPEG-4. A partir de allí, se empezaron los estudios y las investigaciones pertinentes para profundizar en el avance que se buscaba en cuanto a la eficiencia en la compresión de video. Todas estas investigaciones llevaron a la creación de un nuevo estándar llamado H.265/HEVC, el cual fue diseñado con el propósito de revolucionar el streaming de video y mejorar considerablemente las transmisiones y descargas, tanto en la globosidad de Internet como en el caso concreto de los dispositivos móviles. Dicha revolución se ha visto reflejada en la posibilidad de visualizar en streaming productos audiovisuales con resoluciones que pudieran superar el Full HD con un peso reducido a comparación de anteriores estándares.

El éxito notable de este codificador se puede evidenciar en muchos aspectos, pero destaca el hecho de que se alcance una calidad similar a H.264/AVC pero a una tasa de bits muy inferior, logrando de esta manera reducir el tamaño del archivo sin que exista una pérdida perceptible de calidad, y finalmente permitiendo la posibilidad de visualización de archivos 4K en todo tipo de dispositivos, incluyendo los móviles. Este codificador ha sido aprobado por MPEG y por empresas de telecomunicaciones (dentro de las cuales hay empresas del sector de la electrónica y la televisión), logrando así el aceleramiento en la implementación y el uso por parte de las compañías electrónicas, ya que la UIT ha fomentado el uso de este codificador. Este crecimiento exponencial se ve reflejado en la mayoría de televisores que cuentan con el formato 4K, entre otras cosas gracias al soporte de este codificador de forma nativa. De esta manera se espera un crecimiento interesante en muchos otros dispositivos electrónicos capaces de aprovechar al máximo el uso de esta nueva implementación.

Este proyecto tiene como finalidad principal estudiar la paquetización de los datos y buscar, analizar, seleccionar y utilizar las herramientas que nos permitan simular las pérdidas típicas de transmisiones inalámbricas, de forma que se observe el comportamiento del estándar frente a posibles pérdidas y determinar la manera que se comporta tanto el codificador como el decodificador, y utilizar dichas herramientas para intentar mejorar este comportamiento con el objetivo de recuperarse frente a errores.

Índice general

1. INTRODUCCIÓN	9
1.1. MOTIVACIÓN	9
1.2. OBJETIVOS	10
1.1.1. OBJETIVO GENERAL	10
1.1.2. OBJETIVOS ESPECIFICOS	10
1.3. METODOLOGIA APLICADA	10
1.4. ESTRUCTURA DE LA MEMORIA.....	11
2. CONCEPTOS	13
2.1. FOTOGRAMA	13
2.2. FORMATOS DE AUDIO.....	14
2.2.1. MP3	14
2.2.2. WMA.....	15
2.2.3. AAC	16
2.2.4. OGG	16
2.2.5. WAV.....	17
2.2.6. FLAC.....	18
2.3. FORMATOS DE VIDEO	18
2.3.1. MPEG 4	19
2.3.2. XDIV	21
2.3.3. DIVX	23
2.3.4. VP8.....	23
2.4. CONTENEDORES.....	24
2.4.1. AVI	26
2.4.2. FLV	27
2.4.3. MKV	28
2.4.4. MP4	29
2.4.5. OGG	30
2.4.6. ASF.....	31
2.5. VQM	32
2.6. UNIDAD NAL	33
2.7. HIGH EFFICIENCY VIDEO CODING	34
2.7.1. CODIFICACION EFICIENTE	36
2.7.2. CARACTERISTICAS	37
2.8. GROUP OF PICTURES.....	38
3. CONOCIMIENTO DEL SOFTWARE.....	41
3.1. FFMPEG	41
3.1.1. INSTALACION DEL FFMPEG.....	41

3.1.2.	PRUEBAS DE COMANDOS BASICOS DEL SOFTWARE	45
4.	DESARROLLO DEL PROYECTO	49
4.1.	FACTORES QUE AFECTAN LA CALIDAD	49
4.1.1.	Constante QP (CQP).....	49
4.1.2.	Promedio de velocidad de bits (ABR)	51
4.1.3.	Velocidad binaria constante (CBR)	53
4.1.4.	Calidad constante (CQ) / Factor de frecuencia constante (CRF).....	55
4.1.5.	Codificación restringida (VBV)	56
4.2.	TRANSMISION DE FLUJOS DE VIDEO (STREAMING).....	57
4.3.	PERDIDA DE UNIDADES NAL.....	61
4.4.	RECUPERACION DE LA INFORMACION PERDIDA	64
4.4.1.	MEncoder	64
4.5.	RECOMENDACIONES DE OTRA POSIBLES SOLUCIONES	68
4.5.1.	BIT DE PARIDAD	68
4.5.2.	FORWARD CORRECTION	68
4.5.3.	CODIGOS RAPTOR.....	69
5.	CONCLUSIONES.....	71
6.	BIBLIOGRAFIA.....	72

Índice de imágenes

Imagen 1. Ejemplo de ilustración y representación de un celuloide	13
Imagen 2. Principales partes de MPEG-4	21
Imagen 3. Comparación entre contenedores	25
Imagen 4. Visualización de los Chunks en el contenedor AVI	26
Imagen 5. Formatos de compresión soportados por FLV	27
Imagen 6. Diseño del contenedor OGG	30
Imagen 7. Diagrama de bloques de un sistema VQM	33
Imagen 8. Composición de las unidades NAL	34
Imagen 9. Relación consumo de ancho de banda de h.264 y h.265	35
Imagen 10. Diagrama de bloques HEVC	36
Imagen 11. Definición de GOP	38
Imagen 12. Error en la instalación de FFMPEG con librería dinámica	42
Imagen 13. Error por librerías dinámicas	42
Imagen 14. Ejecutables después de la instalación	43
Imagen 15. Prueba de ffmpeg	44
Imagen 16. Funcionamiento del ffmpeg	44
Imagen 17. Comando para recortar video a un determinado tiempo de duración ...	45
Imagen 18. Comando para convertir el video a un formato diferente	45
Imagen 19. Izquierda x264 con mp4 ; Derechas x265 con mp4	46
Imagen 20. Comparación de tamaño para las imagen 19	46
Imagen 21. Izquierda x264 con mkv ; Derecha x265 con mkv	47
Imagen 22. Comparación de tamaño de la imagen 21	47
Imagen 23. Comando para modificar constante QP	51
Imagen 24. Comando para variar velocidad de bits	51

Imagen 25. Codificación con velocidad de bits 10K.....	52
Imagen 26. Codificación con velocidad de 1M.....	52
Imagen 27. Velocidad binaria constante de 10K con buffer de 10K	53
Imagen 28. Velocidad binaria de 1M con buffer de 10K	54
Imagen 29. Comando para ajustar velocidad constante	54
Imagen 30. Comando para modificar la frecuencia constante CRF	55
Imagen 31. Frecuencia constante CRF=1	55
Imagen 32. Frecuencia constante CRF=51	56
Imagen 33. Comando para transmitir el número máximo de bits	56
Imagen 34. Máximo número de bits 1000 con buffer de 2000	57
Imagen 35. Comando para enviar por RTP un archivo de video	58
Imagen 36. Características del archivo enviado por RTP	59
Imagen 37. Asociación de la dirección IP y puerto del streaming.....	59
Imagen 38. URL del archivo enviado por streaming	60
Imagen 39. Lista de reproducción en reproductor VLC	61
Imagen 40. Comando para pasar a modo lectura de hevcbrowser	62
Imagen 41. Archivo en formato lectura hevcbrowser	62
Imagen 42. Interfaz de cada archivo abierto por hevcbrowser	63
Imagen 43. Comando para restablecer al formato original	63
Imagen 44. Resultado con eliminación de unidades NAL.....	65
Imagen 45. Comando de MEncoder utilizando IDX	65
Imagen 46. Resultado con la reconstrucción utilizando IDX.....	66
Imagen 47. Comando de MEncoder utilizando FORCEIDX	66
Imagen 48. Resultado de la reconstrucción utilizando FORCEIDX	67
Imagen 49. Enfoque en la calidad con pérdidas de unidades NAL	67

Capítulo 1

1. INTRODUCCIÓN

La UIT en sus esfuerzos por mejorar la codificación de video avanzada creó un estándar capaz de generar y proporcionar una buena calidad de imagen sin aumentar el nivel de complejidad de sus sucesores, dentro de los cuales se encuentran: MPEG 2, MPEG 4, H.263, H.264, etc. Por tal motivo, el H.265/HEVC es el estándar de video que al día de hoy se utiliza en tabletas, teléfonos móviles, streaming en internet, videos de alta definición en Bluray, etc. Es un estándar altamente extendido de compresión de video por sus beneficios en cuanto a peso del archivo, calidad de la imagen, velocidad por bits, entre otras.

1.1. MOTIVACIÓN

En los últimos años el crecimiento en la demanda del uso de la tecnología ha permitido que ésta se vea aprovechada al máximo en cuanto a su almacenamiento, calidad de la imagen y velocidad de procesamiento; por tal motivo, se requiere trabajar en la implementación de nuevas tendencias que permitan sin lugar a duda mejorar las diferentes características de los dispositivos de uso tecnológico, de allí que la necesidad de implementar y mejorar cada una de ellas y con esto generar de manera adecuada un valor agregado a los diferentes dispositivos de uso tecnológico (ITU, 2013).

Para referirnos en un lenguaje coloquial, podríamos denominar esta necesidad como “la madre de la creación”, pues al tener los dispositivos de uso tecnológico de última generación video en Full HD y 4K, esto trae consigo una resolución mucho más costosa en cuanto a espacio de almacenamiento y procesamiento, como por ejemplo, las imágenes multipíxel. Por tal motivo con el uso del codificador H.265/HEVC se minimizarían en cantidad los requisitos de almacenamiento, transmisión y procesamiento de video, lo que maximizaría los beneficios de los usuarios en cuanto a espacio de almacenamiento, coste de procesamiento en el dispositivo y provecho en imágenes con la calidad necesaria para satisfacer diferentes puntos de vista técnico (Sullivan, Ohm, Han, & Wiegand, 2012).

No obstante, en los últimos años se ha venido introduciendo el uso de éste nuevo codificador, trayendo consigo muchos vacíos en su uso e implementación,

Es por ello que se requiere una mayor investigación y documentación del codificador, para poder garantizar el adecuado uso de éste y el mejoramiento considerable de la calidad de imagen y video que al día de hoy se proyectan. En particular en el ámbito bajo redes propensas a errores, en la que la pérdida de paquetes de red puede llevar un grave deterioro de la calidad del video, sobre todo teniendo en cuenta el alto nivel de la compactación de la información alcanzado con este nuevo estándar.

1.2.OBJETIVOS

1.1.1. OBJETIVO GENERAL

Analizar algunas herramientas para proporcionar robustez a la transmisión de video con hevc principalmente frente a pérdida de información.

1.1.2. OBJETIVOS ESPECIFICOS

- Conocer el software FFmpeg para la codificación y transmisión de archivos de multimedia en diferentes formatos.
- Reconocer y documentar las principales falencias en donde puede haber mejoras en la codificación y decodificación.
- Reconocer la manera de recuperar la información en un escenario de perdida de paquetes.
- Generar recomendaciones para la decodificación en H.265/HEVC.

1.3.METODOLOGIA APLICADA

La metodología utilizada para este trabajo, que se basa en comprobar la robustez del codificador H.265/HEVC, contemplo las siguientes etapas:

- Documentación: Se buscó la información necesaria para documentar y dar validez a las ideas propuestas, mediante artículos y patentes

recolectados de fuentes confiables (IEEE, ITU, entre otras), y de esta manera se estableció los parámetros para la realización de las pruebas.

- **Planificación:** Se realizó un análisis previo a la comprobación de la robustez del codificador tomando en consideración diferentes escenarios de prueba.
- **Experimentación:** Se basó en la realización de un conjunto de pruebas con el fin de obtener datos relativos y establecer un punto de partida, para de esta manera poder ejecutar el proyecto y cumplir de forma satisfactoria los objetivos planteados.
- **Recopilación de dato:** Se recolectaron datos y se representaron en imágenes que muestran la calidad del resultado una vez aplicadas las técnicas de recuperación frente a errores.
- **Análisis de los datos:** Se exponen los resultados obtenidos para establecer comparativas en cuando a los diferentes escenarios planteados, y de esta manera poder generar las conclusiones del proyecto.

1.4. ESTRUCTURA DE LA MEMORIA

La memoria del presente trabajo se compone en cinco capítulos, los cuales se desglosan de la siguiente manera:

- **Capítulo 1:** En este primer capítulo se introduce el proyecto realizado, así como sus objetivos aclarando de esta manera la visión general y los objetivos a alcanzar, la estructura del trabajo y la metodología utilizada para la realización del proyecto.
- **Capítulo 2:** En este capítulo se exponen y explican los conceptos y conocimientos necesarios para el entendimiento de los diferentes temas que se abordaran facilitando al lector una comprensión y un flujo continuo en su lectura.
- **Capítulo 3:** En este tercer capítulo se realizará una explicación del conocimiento del software en cuanto a manipulación de diferentes comandos y resultados obtenidos.
- **Capítulo 4:** En el penúltimo capítulo y más importante se mostrará el desarrollo y la demostración de cada una de las pruebas realizadas con el fin de establecer la base fundamental del proyecto.

- **Capítulo 5:** En este último capítulo se establecen las conclusiones del proyecto basadas en el desarrollo y comprobación.

Capítulo 2

2. CONCEPTOS

2.1. FOTOGRAMA

Un fotograma es una imagen que se obtiene por la colocación de objetos sobre una superficie fotosensible, esto significa que un fotograma es un flash de un instante de tiempo que equivale a un cuadro en una película o video (Vesga, 2015).

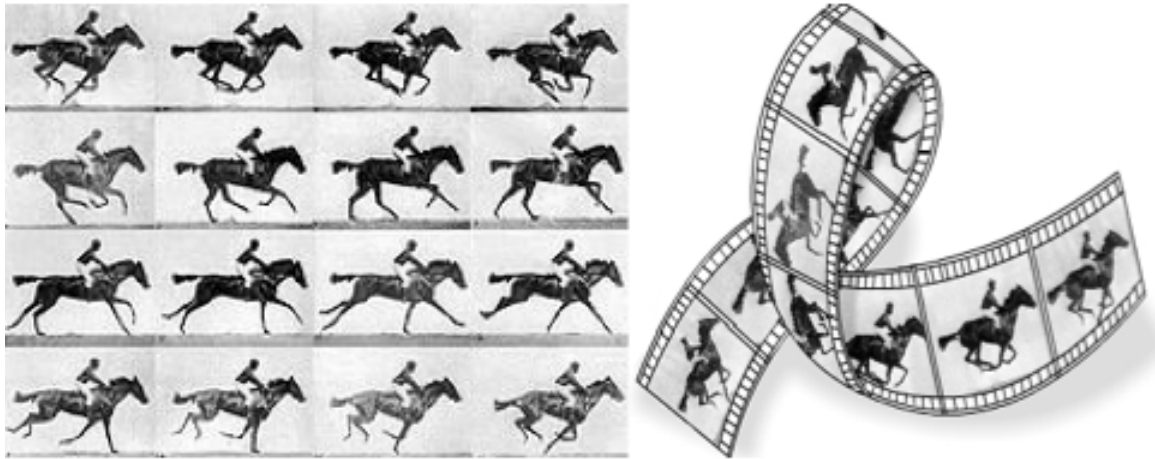


Imagen 1. Ejemplo de ilustración y representación de un celuloide

Sin embargo, existe una confusión cuando se trabaja el término de “fotograma” con el de “cuadro”, debido a que en mucha teoría se habla de estos dos términos como equivalente, pero esto no es del todo cierto, existen una razones por las cuales no se pueden catalogar como iguales, estas son:

1. El fotograma es una mezcla visible de imágenes, el cuadro es una señal eléctrica que se puede grabar.
2. Los fotogramas se adoptaron principalmente por razones logísticas en las cuales se determinaba la cantidad, en cambio el cuadro siempre se ha determinado por las frecuencias eléctricas.

3. El fotograma no esta compuesto por divisiones, por otro lado, el cuadro tiene dos: inferior y superior.

2.2. FORMATOS DE AUDIO

Existen distintos formatos para poder guardar los audios digitales, estos formatos están representados por una extensión en especifica que permite identificarlos con facilidad. Existe muchos formatos de audio, pero no todos son audibles utilizando un mismo reproductor, entre estos se encuentra: QuickTime, Real Player, Windows Media Player, etc.

La diferencia entre estos distintos formatos de audio son sus propiedades, su capacidad para reproducir, la manera como se almacenan los datos y la forma en que se puede utilizar en un sistema de administración de archivos. Por ejemplo, la manera de almacenar un audio digital es por medio de su voltaje de audio, de esta manera al reproducirlo, ese corresponderá a un nivel de señal creando una frecuencia de muestreo por que esta tiene cuenta el número de bits en intervalos regulares. De esta manera se puede o no comprimir el archivo para reducir el espacio de almacenamiento (Martin).

La compresión del audio depende según el formato que se utilice, cabe resaltar que es muy importante distinguir entre un formato y un códec. El códec codifica y decodifica los datos del audio los cuales van a ser almacenados en un archivo que tiene un formato de audio en específico. A continuación, se nombrará alguno de los formatos de audio más utilizados con una breve descripción, para la comprensión de estos.

2.2.1. MP3

MPEG es un grupo de expertos en películas quienes llevan un largo tiempo proponiendo ideas para poder codificar el audio y video de forma digital, este grupo recomendó el uso de MP3 que es la abreviatura de MPEG Audio Layer 3, el cual es un formato de compresión de archivos de audio muy utilizado. Este formato .mp3 se convirtió rápidamente en el estándar utilizado para el streaming de audio y compresión de audio en alta calidad, con la ventaja que permite ajustar la calidad de la compresión modificando el bitrate y de esta manera mejorar el tamaño final del archivo que podría significar una disminución considerable en el espacio que el archivo ocupa frente al original (Arango, 2013).

CARACTERISTICAS:

- Se popularizo rápidamente por su grado de compresión y alta calidad de la información, presentando una mínima perdida de calidad.
- Por su capacidad de ser reproducido por la mayoría de los reproductores de audio, se convirtió en el archivo ideal para publicar audios en la web.
- Tiene la ventaja de poder ser decodificado rápidamente.
- Tiene un mayor nivel de compresión si se compra con otros formatos de audio, si se maneja igual de condiciones con los demás este reduciría en el tamaño del archivo en un factor de 1/10.

2.2.2. WMA

El formato de audio WMA se caracteriza por el tamaño de sus archivos, estos pueden llegar a ser menores a un .MP3, pero por tal razón, la calidad disminuye sin importar que conserve parámetros para la integridad de los datos, sobre todo de la calidad. De lo anteriormente mencionado se puede deducir que es un formato de audio con perdida, pero se mantiene como una de las principales competencias del MP3 por la difusión que mantiene en la web debido al tamaño de sus archivos (Arango, 2013).

Por su importancia dentro del sistema operativo Windows, que viene por defecto en el reproductor Windows Media Player, su popularidad en internet se vio en aumento por la cantidad de demanda que tiene el sistema operativo, por esta razón, el formato por lo general es usado en diversas páginas web para la pre-escucha de la música y en algunos reproductores de música digital.

CARACTERISTICAS:

- Existen varios tipos de WMA dentro de los cuales están: Windows Audio Media Professional y Windows Audio Media sin perdidas, su deficion es muy sencilla, el primero es utilizado a nivel profesional debido a su codificación multicanal y alta calidad; el segundo es usado por la misma familia de Windows que permite perdidas y es soportado por la mayoría de reproductores.
- WMA se encuentra ligado a varios otros formatos, como WMV y WAV, que están en la familia de Windows.

- Para toda la industria musical es un formato beneficioso para los intereses económicos, debido a la protección que puede ser implementada por derechos de autor y no permite ser copiado el archivo.

2.2.3. AAC

AAC representa un formato de codificación de audio avanzado basado en un algoritmo de compresión con pérdida, su funcionamiento en síntesis es muy sencillo, lo que hace es eliminar datos de audio aleatorios para mejorar y obtener una mejor compresión. De esta manera, mejora la compresión y la salida del archivo es muy similar a la original debido a los datos que fueron eliminados.

Como en MP3, el AAC también usa frecuencias de muestreo en este caso son de 8 hZ a 96 kHz usando la transformada discreta del coseno modificada junto con bloques de longitud de 2048 puntos. Esto anteriormente mencionado lo que hace es codificar el audio con corrientes de pulsos y ondas cuadradas comparando con MP3 y de esta manera hacer que su salida sea lo más similar a la original sin importar los datos que se eliminaron en el primer proceso (APPLE, 2016).

CARACTERISTICAS:

- Debido a su buen desempeño, rendimiento y calidad, es el códec de audio de internet, conexiones inalámbricas y también muy usado en radio difusión digital.
- ACC ha sido empleado por Apple para ser utilizado en su plataforma Itunes. También utilizado en Nintendo DSI y Winamp.
- Por ser una codificación de audio avanzada, ha tenido el reconocimiento de MPEG hasta el punto de ser el núcleo del MPEG-4, y de otras como 3GPP y 3GPP2, utilizados principalmente en dispositivos móviles.

2.2.4. OGG

OGG es un formato de audio libre y abierto, el cual no se encuentra restringido en ningún aspecto, su diseño fue enfocado para proporcionar y garantizar una difusión de flujo eficiente manteniendo una alta calidad en el audio. Este formato de audio también se considera con Compresión por pérdida, igual que AAC elimina datos de audio aleatorios, con el objetivo de obtener un buen grado de comprensión y de esta manera se pueda obtener un archivo muy similar al original (Profesorado., 2008).

Una de sus principales ventajas es que se puede utilizar libremente por los productores de software garantizando herramientas y librerías libres de derecho que puedan utilizar este tipo de formato de audio. Como casi todos los formatos de audio, OGG encapsula cada dato de manera que los puede comprimir y descomprimir obteniendo una interpolación de los datos dentro del formato.

CARACTERISTICAS:

- OGG no es compatible con MP3, por lo cual, el usuario deberá utilizar un reproductor que sea compatible con el formato de audio. La ventaja es que la mayoría de reproductores soportan el OGG.
- Una de las ventajas de este formato es que puede codificar audio y video dentro del mismo archivo.
- Debido a la difusión de flujo el archivo puede ser escrito y leído en un solo paso.
- Mecanismos fáciles para la detección y corrección de algunos datos por las posiciones arbitrarias de los flujos de bits.

2.2.5. WAV

WAV es un formato de audio digital creado por Microsoft e IBM, principalmente para el almacenamiento de archivos de audio en el computador, este formato no cuenta con compresión de datos por lo que el espacio de almacenamiento de cada archivo es mayor (computacion, 2015).

Una ventaja del formato WAV es su compatibilidad con casi cualquier codec de audio, y como no presenta ninguna pérdida de calidad ya que no se comprime el archivo es adecuado para el uso profesional. Esto significa que no es popular en internet, esto debido a que los archivos cuando no tienen compresión alguna son muy pesados lo que no permite una transferencia por medio de internet adecuada.

CARACTERISTICAS:

- Su principal ventaja es que es un formato ideal para guardar archivos originales con una excelente calidad y sin pérdidas.
- Su desventaja es que produce archivos de gran tamaño. Un claro ejemplo es una canción extraída de un CD puede estar oscilando en peso entre 20 y 30 Mb.

- Es utilizado en fragmentos muy cortos y con audio mono debido a su buena calidad.
- Admite archivos mono y estéreo teniendo en cuenta que se puede manejar diferentes resoluciones y manejando una frecuencia de muestreo variable.

2.2.6. FLAC

FLAC es un códec de audio que permite comprimir sin pérdidas de esta manera lo que se garantiza es que el tamaño del archivo se reduzca considerablemente sin que se pierda ningún tipo de información. Esta reducción se puede ver reflejada entre un 50 a 60% de su tamaño original, todo debido a algoritmo utilizado el cual garantiza a la compresión y la descompresión del archivo como una copia muy similar a la original (Arango, 2013).

Al igual que OGG, FLAC es un formato abierto que tiene sus licencias libres con la característica que es portable a diferentes sistemas, con una decodificación extremadamente rápida debido al tamaño de los archivos. Contiene soporte de streaming de alta resolución por el algoritmo con el que cuenta y de esta manera puede ser apoyado por dispositivos Hardware.

CARACTERISTICAS:

- Debido a las pérdidas que tiene su compresión es extremadamente lenta comparada con la descompresión.
- Los archivos con este tipo de formato por lo general tienen la extensión .flac, sin embargo, no es raro observar en alguna situación con la extensión .fla.
- Como es un formato sin pérdida, la ratio va a depender fundamentalmente de estilo musical, variedad sonora y diferentes características de los datos del audio para poder ser manejada por las frecuencias de cada variedad.

2.3. FORMATOS DE VIDEO

Debido a todas las revoluciones y actualizaciones tecnológicas presentadas hoy en día, el video entra dentro de esta innovación encontrando de esta manera infinidad de formatos de video, cada uno de ellos dependiendo de la calidad, de la manera que se exporte el video, entre otras. Hoy en día se pueden encontrar diferentes formatos que se encuentran destinados a diferentes cosas, por ejemplo,

formatos enfocados solamente para páginas web o videos en YouTube, para proyectar en calidad de cine o simplemente para poder ser reproducidos en un dispositivo móvil (tecnología, 2008).

La manera de producir de videos se vería limitada si no existieran la variedad de formatos y compresores de video como los hay hoy en día, debido a que ocuparía un espacio considerable de almacenamiento y la calidad no estar reflejada como se puede observar en la actualidad, factores que hacen que el desarrollo en cuanto a formatos de video se vea explotada para satisfacer las necesidades actuales y obtener los resultados que todo usuario desearía, como una buena calidad, buen procesamiento y un buen espacio de almacenamiento.

Por estas razones expresadas anteriormente, se explicará algunos de los formatos de video utilizados en la actualidad, debido a las diferentes características que presentan. No es lo mismo exportar un video para algún canal de internet que para un dispositivo móvil. Presentan características esenciales que hacen que se saque el mayor rendimiento en cuanto a las características que pueden ser explotadas y obtener el mejor resultado (tecnología, 2008).

2.3.1. MPEG 4

MPEG 4 es un codificador de audio y video que lo estandarizo el grupo de investigación y desarrollo MPEG, en la década de los 90's, con la principal función de mejorar el flujo de los medios audiovisuales, optimizarla distribución en CD, y una función en especial la cual consiste en garantizar la transmisión bidireccional en la emisión de televisión.

Para mejorar este tipo de codificador el grupo de investigación se basó y tomó muchas características de las versiones de codificadores anteriores (MPEG-1 y MPEG-2), así como algunos otros estándares. Sin embargo, estos últimos fueron tomados como soporte de VRML para mejorar la visualización 3D, y garantizar una gestión en cuanto a los derechos digitales y ver una variabilidad en cuanto a la interactividad del codificador (QUEVEDO, 2003).

Dentro de las características que conforman el MPEG-4 y lo volvieron innovador fueron:

- Ofrece un mayor ratio de compresión garantizando la forma de codificar tanto video como imágenes.
- Este codificador es sin duda más potente en cuanto a capacidad y diferentes posibilidades de variabilidad.

- Implementa un perfil totalmente diferente, mejorando el nivel de codificación.
- Tiene la capacidad de recuperar errores y de esta manera permitir una transmisión mas robusta.
- Tiene la ventaja de que puede interactuar con el receptor en una escena audiovisual generada por este.

MPEG-4 esta formado por varios estándares llamados partes, cada una de ellas hace parte de la especificación completa del codificador:

- **Parte 1:** (ISO/IEC 14496-1), Sistemas
- **Parte 2:** (ISO/IEC 14496-2), Visual
- **Parte 3:** (ISO/IEC 14496-3), Audio
- **Parte 4:** (ISO/IEC 14496-4), Conformidad
- **Parte 5:** (ISO/IEC 14496-5), Software de referencia
- **Parte 6:** (ISO/IEC 14496-6), DMIF
- **Parte 7:** (ISO/IEC 14496-7), Software optimizado de referencia
- **Parte 8:** (ISO/IEC 14496-8), Transporte sobre redes IP
- **Parte 9:** (ISO/IEC 14496-9), Hardware de referencia
- **Parte 10:** (ISO/IEC 14496-10), AVC
- **Parte 11:** (ISO/IEC 14496-11), Descripción de escena y motor de aplicación
- **Parte 12:** (ISO/IEC 14496-12), Formato para medios audiovisuales basado en ISO
- **Parte 13:** (ISO/IEC 14496-13), Extensiones para el manejo y protección de Propiedad Intelectual
- **Parte 14:** (ISO/IEC 14496-14), Formato de archivo MPEG-4
- **Parte 15:** (ISO/IEC 14496-15), Formato de archivo AVC
- **Parte 16:** (ISO/IEC 14496-16), AFX
- **Parte 17:** (ISO/IEC 14496-17), Formato de subtítulos
- **Parte 18:** (ISO/IEC 14496-18), Compresión y transmisión como flujo de fuentes tipográficas
- **Parte 19:** (ISO/IEC 14496-19), Flujo de texturas sintetizadas
- **Parte 20:** (ISO/IEC 14496-20), Representación liviana de escenas
- **Parte 21:** (ISO/IEC 14496-21), Extensión de MPEG-J para rendering
- **Parte 22:** (ISO/IEC 14496-22), Formato de fuentes abierto
- **Parte 23:** (ISO/IEC 14496-23), Representación simbólica de la música
- **Parte 24:** (ISO/IEC 14496-24), Interacción con el sistema y el audio
- **Parte 25:** (ISO/IEC 14496-25), Modelo de compresión de gráficos 3D
- **Parte 26:** (ISO/IEC 14496-26), Conformidad del audio
- **Parte 27:** Conformidad de los gráficos 3D

MPEG-4

MPEG-4 PARTE 2	VIDEO	H.263
MPEG-4 PARTE 3	AUDIO	AAC
MPEG-4 PARTE 10	VIDEO	H.264
MPEG-4 PARTE 14	CONTENEDOR	MP4
MPEG-4 PARTE 17	SUBTÍTULOS	

Imagen 2. Principales partes de MPEG-4

De esta manera se puede garantizar que cada una de las partes tengan una función fundamental en el estándar, proporcionándole una amplia gama de interacción con diferentes objetos multimedia, mejorando habilidades de flexibilidad y adaptabilidad para mejorar la calidad de los codificadores a medida del tiempo (QUEVEDO, 2003).

Cabe resaltar que MPEG-4 PARTE 10 es H.264/AVC, este es uno de los estándares más empleados hoy en día, por ejemplo, es el estándar utilizado en DBV-T de alta definición y en el Bluray. Sin embargo, con la implementación de los nuevos estándares como HEVC ofrecen diferentes mejoras y optimizaciones que permiten un aprovechamiento mejor de la imagen.

2.3.2. XDIV

XDiv es códec popular debido a que fue desarrollado por programadores de todo el mundo para convertirlo en un software libre, basado en el códec MPEG-4, sin embargo, su formato fue creado de forma libre a otros codecs comerciales.

Mejora considerablemente la calidad y la eficiencia y por estas razones se ha convertido en uno de los codecs más populares. Un claro ejemplo de lo anteriormente dicho es la reproducción de las películas XDiv que están soportadas en los reproductores de DVD que en su época fueron importantes (QUEVEDO, 2003).

Uno de los principales factores de este códec es que logra comprimir una película completa con una calidad similar a la original en un tamaño bastante reducido; las películas que se encuentran con este formato pueden estar en un rango de 700 MB a 1400 MB, dependiendo de la duración y algunos otros factores como idiomas, subtítulos, entre otros. Además de codificar los archivos en un tamaño considerable utiliza un tiempo de compresión menor debido al algoritmo utilizado. Pero no solo el video necesita ser de alta calidad por lo que el audio también tiene que tener esta característica, los formatos de audio utilizados en este códec son MP3 y AC3. Estos factores hicieron que el códec se distribuyera de forma libre con gran acogida y éxito (QUEVEDO, 2003).

Esta gran acogida y éxito se debió a cuatro características que se explicarán a continuación:

- **Mayor rendimiento:** este rendimiento se ve reflejado cuando se realiza una grabación de video en tiempo real o se reproduce videos HD ya que por la manera de utilizar el algoritmo optimiza el uso de la CPU de tal manera que el procesamiento de la computadora nunca se verá afectado.
- **Ventajas de uso:** Este códec es usado en muchas plataformas y tiene la ventaja de que puede ser visualizado en muchos reproductores de video populares. Además los distintos dispositivos (Televisores o primeros celulares) o reproductores DVD también manejan este códec.
- **Fuerte compresión:** Esta característica lo popularizó debido a que podía comprimir archivos y volverlos más pequeños en una proporción de 200:1 o aún más considerable la escala si el video se encuentra sin comprimir, haciendo que la transmisión por medio de la red sea con una velocidad mayor y a su vez ahorrando espacio de almacenamiento.
- **Calidad de imagen:** Sin importar la reducción del archivo al ser comprimido, este no se ve afectado en cuanto a la calidad de la imagen debido a que lo puede realizar sin apenas causar pérdida en la calidad percibida por el usuario conserva una imagen nítida comparada con otros codecs. Para ello, elimina detalles de la imagen que no son importantes para la percepción humana de esta manera garantiza un menor almacenamiento del archivo final.

2.3.3. DIVX

DivX es otro códec de video basado en MPEG-4, permite comprimir videos usando algoritmos con perdida, esto quiere decir, cuando se realiza una copia de cualquier video y se hace una comparación con la comprimida byte a byte se observarían diferencias considerables Por lo general la copia es de menor calidad. Sin embargo, existen excepciones con respecto a la pérdida de calidad cuando se realiza la compresión ya que si se sigue una técnicas adecuadas, se puede comprimir una película en formato DVD en 1 o 2 Cd con una buena calidad debido a las técnicas que fueron usadas (tecnología, 2008).

Técnicamente DivX es un formato de video que principalmente funciona sobre los sistemas operativos de MacOs, Windows y Linux, combinando la compresión del audio con una calidad aceptable de la imagen. En la actualidad es un códec que aún se utiliza para la compresión de películas, sin embargo, pierde terreno con el códec XviD, debido a que este ultimo es un software libre y ofrece una compresión de alta calidad. Pero DivX compite con factores importantes como posibilidades de configuración y ser compatible con múltiples reproductores.

El auge de este códec se vio reflejado en la época de la llegada de los CDROM, que en muchos casos se codificaban con este códec, por tal motivo su éxito fue abismal, tanto en calidad de audio y video para la época como por tamaño de almacenamiento debido a que en un CDROM solo se podía ocupar un máximo de 80 minutos. Por tal motivo el códec fue creado con algoritmo con perdidas para poder acoplarse a la tecnología del momento y ser muy útil en muchos aspectos (QUEVEDO, 2003).

Hubo un cambio considerable en el uso del códec debido a que su funcionalidad en el CDROM se estaba viendo limitada, de tal manera que modificó por la formación de dos software que su funcionalidad estaba enfocada en la extracción de contenido, de esta manera lo que se estaba logrando era romper de alguna forma la encriptación que permitía que los archivos de DVD fueran copiados en el PC, y por otro lado lo que se permitió fue que las películas con formato DivX pudieran ser encontrados con la extensión .avi y de esta manera su ejecución en los diferentes reproductores fuera mas sencilla. Lo único que se tenía que hacer era instalar un códec DivX.

2.3.4. VP8

VP8 es un formato de compresión de video encargado de codificar y decodificar videos de alta definición. Este códec es un contraste de h.264, sin

embargo, es un códec gratuito debido a que Google quien fue su desarrollador libero cada una de las patentes bajo licencias publicas, de esta manera se garantizo el uso y distribución masiva.

El archivo de un video contiene bastante información almacenada que es capturada por una cámara, para poder manejar toda esta información es necesaria una compresión de los datos que no presente perdidas. Por esta razón VP8 logra conseguir un funcionamiento excepcional capaz de garantizar en una misma conexión de internet reproducir videos 720p sin ninguna afectación debido a los algoritmos avanzados que utiliza, garantizando un buen almacenamiento del archivo resultado y un excelente procesamiento (QUEVEDO, 2003).

CARACTERÍSTICAS:

- Permite implementaciones a nivel de decodificación con una memoria relativamente pequeña debido a que realiza una exploración progresiva.
- Soporta señales de video de exploración progresiva comparada con un muestreo cromático, lo que hace que la codificación sea compleja pero muy efectiva en cuanto a perdida debido a la comparación que hace entre bits.
- Para la edición de video utiliza marcos codificados, esto quiere decir que no realiza predicciones sino por el contrario realiza comparaciones en tiempo real realizando un barrido de todos los bits.
- Solo necesita tres buffers de referencia para lograr la codificación.

2.4. CONTENEDORES

Los contenedores son formatos de archivo los cuales contienen diferentes tipos de datos, que se encuentran comprimidos mediante la función de los codecs como los nombrados anteriormente. El archivo contenedor se usa para identificar el contenedor que es usado y de esta manera interpolar los datos que se encuentran almacenados en el archivo (Izquierdo, 2011).

Lo primero que se realiza es la creación del contenedor, de esta manera se produce una codificación de las datos que son multiplexados siguiendo diferentes factores para cada formato. De esta manera cuando el archivo se encuentra listo para ser reproducido lo primero que se ejecuta es un *splitter*, cuya función es conocer la función del contenedor y saber como interactúa este, de esta manera

divide el audio y el video. Cuando ya se encuentran separadas el decodificador actúa dependiendo de su algoritmo y se reproduce.

En resumen no solo es necesario saber el formato del contenedor para poder separar tanto el audio como el video, también es necesario poder decodificarlas para poder reproducirlas de manera adecuada, garantizando que durante este proceso no exista pérdida de información que afecte el resultado final de la reproducción (Izquierdo, 2011).

Por tal motivo, existen varios tipos de contenedores y diferencias entre cada uno de ellos que hacen que los usuarios tengan preferencias, entre ellos se encuentran:

	AVI	MP4	Ogg	MKV	ASF
Propietario	Microsoft	MPEG	Xiph.org	Matroska.org	Microsoft
Soporta B-frames	no nativo	Si	Si	Si	Si
Audio VBR	Si*	Si	Si	Si	Si
Frame rate variable	No	Si	No	Si	No
Capítulos	No	No	Si	Si	No
Subtítulos	No	txt	Ogg Writ	Todos	No
Streaming	No	Si	Si	Si	Si

Imagen 3. Comparación entre contenedores.

- **Popularidad:** En esta diferencia se encuentra este aspecto que su función es mirar que tan soportado es el contenedor y que tan viable es su reproducción.
- **Soporte:** Característica que garantiza que pueda soportar formatos antiguos y que de esta manera no se creen problemas de incompatibilidad.
- **Overhead:** Esta característica hace relación en cuanto al tamaño del archivo final que generan diferencias entre cada uno de los contenedores.
- **Contenidos avanzados:** Como el contenedor puede soportar diferentes aspectos entre los cuales están: capítulos, subtítulos, idiomas, entre otros.
- **Streaming media:** La manera de poder realizar la transmisión por medio de internet garantizando un flujo de datos adecuado para su difusión.

A continuación se dará una breve explicación de los contenedores mas usados con una leve reseña de ellos.

2.4.1. AVI

AVI es un formato contenedor de multimedia que fue introducido en la década de los 90's por Microsoft como una iniciativa de mejorar la tecnología de video en el sistema operativo propio. Este archivo es capaz de contener tanto audio como video y permite la reproducción en simultaneo. Para que esta reproducción se realice de manera correcta es necesario que se almacenen de forma intercalada, esto quiere decir, que cada fragmento de archivo puede reproducir fotogramas con su respectivo audio simultáneamente.

Cabe resaltar que el formato AVI tiene la capacidad de admitir varios flujos de datos, estos flujos pueden ser por ejemplo diferentes idiomas o varias bandas sonoras, que al ser reproducidas en un reproductor multimedia el usuario es quien decide según sus preferencias cual es la que tendrá relevancia. Este se logra gracias a que este contenedor se divide en fragmentos que se encuentran claramente diferenciados denominados Chunks (II, 2015).

Cada uno de los Chunks tiene una etiqueta llamada FourCC, cuya función es darle una identificación a cada fragmento. El primer fragmento o Chunk es llamado cabecera y contiene toda la información respecto al archivo, entre las cuales se puede encontrar: tamaño del archivo, dimensiones de la imagen, velocidad de fotogramas por segundo, etc. El segundo chunk contiene la información entrelazada de audio y video mencionada anteriormente; en algunas ocasiones puede existir un tercer chunk el cual actúa como una lista de reproducción de los demás fragmentos.

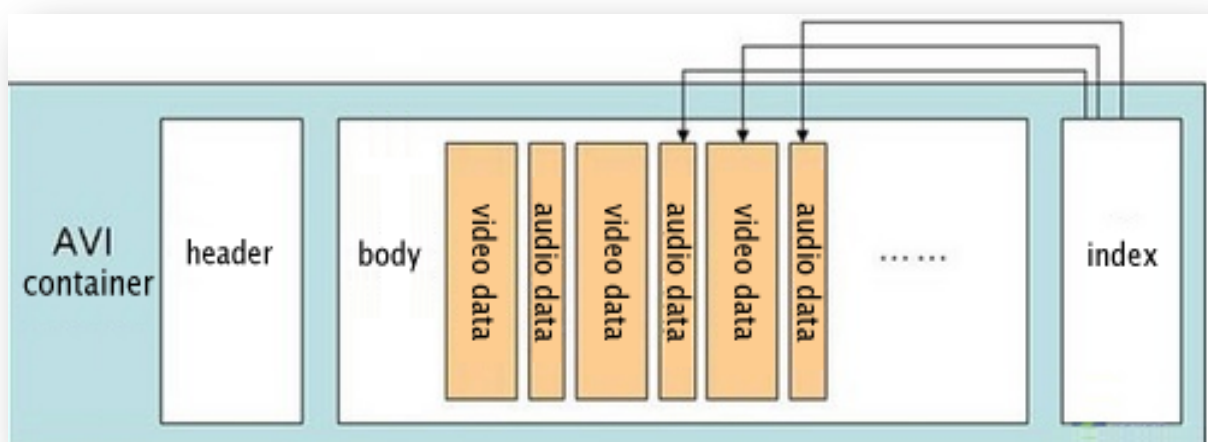


Imagen 4. Visualización de los Chunks en el contenedor AVI

En conclusión, el reproductor tiene la capacidad de leer los fragmentos consecutivamente de los archivos de audio y video que se encuentran intercalados en el segundo chunk, estos son separados y almacenados en un Buffer de memoria que luego pasan al códec correspondiente. Este códec después de realizar su tarea devuelve la información a otro buffer que contiene cada uno de los fotogramas que se van a reproducir. Algo similar sucede con la información de sonido, solamente que se tiene que sincronizar los fotogramas con cada uno de la información de sonido y luego reproducirla a una velocidad adecuada para su sincronización (II, 2015).

2.4.2. FLV

FLV es un contenedor cuyo formato y extensión de archivo tiene la principal función de ser utilizado para la transmisión de videos por internet empleando el reproductor de Adobe. Tiene la capacidad de ser reproducidos en la mayoría de los sistemas operativos que incluyan adobe o tengan la posibilidad de instalar el plugin en el navegador, o simplemente con otras aplicaciones o programas de reproducción como VLC, Mplayer, entre otros. Este contenedor fue muy usado en sitios web reconocidos como Google, Youtube, Yahoo entre otros que lo empleaban para la reproducción de sus videos (Concha, 2014). Como punto a recalcar ahora se usa más http en streaming con formato mp4.

Formatos de compresión multimedia			
	ISO/IEC	ITU-T	Otros
Formatos de compresión de video	MJPEG • MPEG-1 • MPEG-2 • MPEG-4 ASP • MPEG-4/AVC • Motion JPEG 2000	H.120 • H.261 • H.262 • H.263 • H.264	AMV • AVS • Bink • Dirac • Indeo • Pixlet • RealVideo • RTVideo • SheerVideo • Smacker • Snow • Theora • VC-1 • VP6 • VP7 • WMV
	ISO/IEC MPEG	ITU-T	Otros
Formatos de compresión de audio	MPEG-1 Layer III (MP3) • MPEG-1 Layer II • AAC • HE- AAC	G.711 • G.722 • G.722.1 • G.722.2 • G.723 • G.723.1 • G.726 • G.728 • G.729 • G.729.1 • G.729a	AC3 • AMR • Apple Lossless • ATRAC • FLAC • iLBC • Monkey's Audio • Mu-law • Musepack • Nellymoser • OptimFROG • RealAudio • RTAudio • SHN • Siren • Speex • TAK • Vorbis • WavPack • WMA
	ISO/IEC/ITU-T		Otros
Formatos de compresión de imagen	JPEG • JPEG 2000 • lossless JPEG • JBIG • JBIG2 • PNG • WBMP		BMP • GIF • ICER • ILBM • PCX • PGF • TGA • TIFF • HD Photo

Imagen 5. Formatos de compresión soportados por FLV

Los archivos FLV son distribuidos en muchas maneras, sin embargo, a continuación se nombrarán las más utilizadas.

- Aunque los archivos con el contenedor FLV son distribuidos por Adobe, existe cierta similitud con AVI o MPEG, teniendo la característica de ser reproducido o convertido a otros tipos de contratos.
- Una de las desventajas según muchos usuarios es que el archivo debe ser transferido en su totalidad para que la reproducción pueda empezar, debido a que no contiene un buffer de memoria que contenga la información decodificada.
- Puede tener una descarga progresiva via http, la cual tiene la ventaja de incluir buffer de datos para de esta manera se puede buscar cualquier tipo del fragmento para recuperar cierta información antes de ser reproducida, sin embargo, no contiene la funcionalidad de soportar transmisiones en tiempo real por las razones anteriormente nombradas.
- Soporta Streaming via RTMP que garantiza que el usuario tenga el control de la calidad de la distribución y la seguridad de la información.

2.4.3. MKV

MKV es un formato de archivo de video flexible con código abierto que se popularizó rápidamente y de esta manera se convirtió en la extensión de archivo preferida para ser usada en internet en los videos de alta definición. Este contenedor tiene la característica que puede incluir video, audio, subtítulos en múltiples lenguajes, sección de capítulos, descripciones, entre otras. Tiene la cualidad que garantiza una experiencia multimedia en cualquier formato que se maneje en la actualidad (Dias, 2015).

La finalidad de este contenedor es servir como un formato universal para poder almacenar cualquier tipo de contenido de multimedia (programas de televisión, video juegos, imágenes, etc.), de esta manera envuelve audio, video, entre otros dentro de un mismo paquete y garantiza la reproducción del archivo en cualquier tipo de dispositivo que contenga la capacidad de procesamiento necesaria para la visualización de este.

Su desarrollo y difusión se debió al apoyo que generó en la transmisión de videos por la web, debido a que utiliza un sistema de menús similares a los del DVD, creando de esta manera una interacción del usuario con el archivo. De esta manera, fue necesario el uso de distintas herramientas para la edición de los archivos .mkv (Dias, 2015), para poder trabajar con fabricantes de hardware y estos

pudiesen utilizar el contenedor en los dispositivos de multimedia. Cabe resaltar que por ser código abierto a los distintos contenedores existentes tienen la ventaja de poderse utilizar en cualquier tipo de formato. Por estas razones se describirán unas características que lo vuelven uno de los contenedores más usados en la actualidad:

- Capacidad de almacenar varias pistas de video (capítulos) programadas para reproducirse en secuencia sin la necesidad de perder calidad.
- Capacidad de almacenar cantidad de pistas de audios (Diferentes lenguajes) para la multiculturalidad.
- Cantidad de capítulos y subtítulos en un mismo fichero, permitiendo de esta manera el uso de funciones avanzadas.
- Posibilidad de menús inteligentes en el futuro debido a la capacidad de almacenamiento en cuanto a ficheros e información.
- La reproducción de video o streaming se puede realizar en tiempo real por la ventaja que tiene respecto a .avi al cambiar de posición el video no se detendrá ni se observará como si la imagen se congelara, por eso una de sus ventajas al manejar capítulos y poderlos sincronizar de manera adecuada.
- Mejor tolerancia y capacidad de mejora en cuanto a errores.
- Capacidad de sobreponer una imagen en el contenedor de esta manera puede servir como portada o como el índice de un menú. Gracias a esto brinda la posibilidad de añadir archivos de cualquier tipo sin importar la ubicación dentro de este.

2.4.4. MP4

El contenedor mp4 es un formato de multimedia que se encuentra definido y estandarizado dentro de MPEG-4. De esta manera, permite almacenar archivos de audio y video con algunas adiciones como lo son los subtítulos. El nombre de la extensión que se aplica es .mp4 para reproductores de audio y video, simulando como una mejora considerable sobre los archivos mp3 (II, 2015).

La utilización de este contenedor, es similar a la de muchos anteriormente nombrados, ya que permite streaming de contenido por medio de internet para reproducir contenidos que tienen muy baja demanda, pero con la característica que

multiplexa pistas de video y de audio en un mismo archivo; contiene cuadros por segundo variables, subtítulos, etc.

2.4.5. OGG

Ogg es un contenedor el cual fue desarrollado con código libre y abierto, fue desarrollado con la intención de proporcionar una difusión en cuanto al flujo de eficiencia y garantizando la manipulación de todo tipo de medios audiovisuales que sean de alta calidad. Esta manipulación se debe a la multiplexación de varios flujos de forma independiente entre los cuales caben resaltar el audio, video, subtítulos, etc (Formatos de video).

El contenedor ogg fue utilizados por todos los archivos de la organización Ogg Vorbis, la cual decidió crear nuevos conjuntos de extensiones para poder describir de manera adecuada y concreta los diferentes tipos de contenido, por ejemplo: .oga para audio, .ogv para video, ogx para multiplexado, entre otros con diferentes cualidades. Debido a esta variedad que presenta y la ventaja de ser un formato libre, muchos codecs han incorporado una serie de reproductores de medios libres para reproducir este tipo de archivos en los GPS y otros medios portátiles para la reproducción de medios audiovisuales.

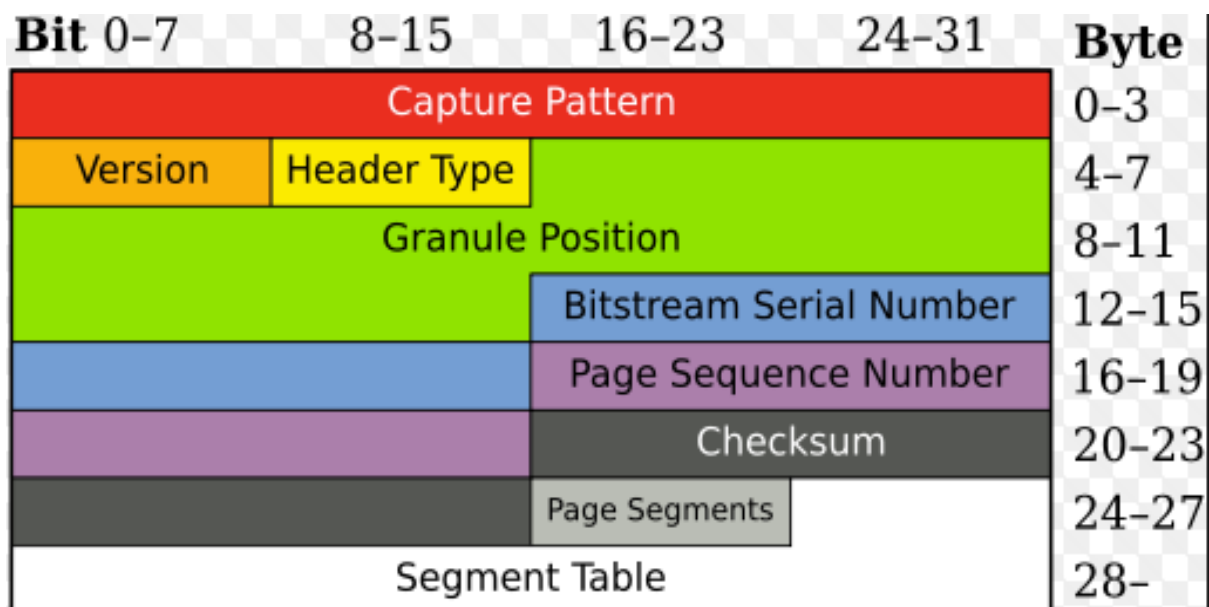


Imagen 6. Diseño del contenedor OGG

Su diseño se encuentra distribuido en varios campos los cuales cabe resaltar: la versión en la cual se manejara, el tipo de cabecera en donde se especificara los diferentes tipos de contenido, la posición del gránulo representa el tiempo absoluto dentro de una secuencia en pocas palabras representa los fotogramas transcurridos en una división de tiempo, los últimos campos ya representan los números seriales del bitstream para llevar un orden en cuanto a continuidad de la secuencia y por ultimo todo lo relacionado a las paginas que terminan siendo las diferentes secciones que componen el archivo (audio, video, subtítulos, etc) (Formatos de video).

Tiene características que lo diferencian de otros contenedores, entre las cuales resaltan:

- Si principal desarrollo y función fue la difusión de archivos de multimedia, teniendo la ventaja de que no necesita un flujo de bits completo, sino que se pueden ir construyendo con diferentes parámetros.
- Tiene una marca para limitar el paquete en cuanto a sincronización y búsqueda del 2% del ancho de banda del flujo de los bits, de esta manera es recomendado para diferentes dispositivos que sirven para la reproducción de archivos audiovisuales.
- Permite la detección de contenido corrupto debido a la organización aleatoria de los datos en posiciones arbitrarias en los distintos flujos que se pueden presentar dentro del contenedor.
- Corrección sencilla debido al uso de un mecanismo de encadenamiento.

2.4.6. ASF

ASF es conocido actualmente como Advanced Systems Format. Sin embargo, este nombre nunca ha sido el oficial, debido a que antes era conocido como Advanced Streaming Format. Su creador y propietario es Microsoft con el fin de contener video y audio, diseñado con el objetivo de ser utilizado en la transmisión de baja demanda que se observa en internet. Debido a que Microsoft es su creador es usado por Windows, los formatos mas comunes de este sistema operativo son Windows Media Audio y Windows Media Video.

Una de sus desventajas es que el formato no especifica la manera en la cual se debe codificar los archivos de audio y video, en otras palabras esto significa que los archivos .asf tienen la posibilidad de contener archivos comprimidos de cualquier tipo de códec y aun así se mantiene con el mismo formato del contenedor. Algo

similar a lo que hacen diferentes contenedores como .avi, .ogg, entre otros (Formatos de video).

Este contenedor tiene características que lo identifican:

- Al igual que otros contenedores, el formato asf tiene la capacidad de contener múltiples datos, entre los cuales sobresalen: nombre del artista o del álbum en caso de ser una pista de audio, longitud y genero.
- La transmisión o reproducción solo se hace posible siempre y cuando se garantice que se ha descargado una cantidad de datos suficientes para iniciar cualquier tarea.
- Tiene la desventaja de solo contener una resolución máxima de 352x288 debido a las capacidades de transferencia con las que cuenta el contenedor.
- Es posible la reproducción en servidores de dispositivos audiovisuales, dispositivos de almacenamiento y algunos servidores web.
- Utiliza un algoritmo de codificación sencillo que permite el intercambio de claves criptográficas, sin embargo, no es tan potente y por tal razón este ha sido sustituido por algunos otros contenedores como el WMA y el WMV.

2.5. VQM

La calidad del video (VQM) es una característica de un video el cual es transmitido por medio de cualquier sistema de transmisión. Esta característica es una medida de la degradación que presenta el video al ser percibido por el usuario teniendo en cuenta un punto de referencia como lo es el video original. El VQM puede tener una métrica basándose en los sistemas de procesamiento que son capaces de introducir la cantidad de distorsión o ruido que sea necesaria en los dispositivos para alterar la señal de video, lo que al ser percibido por el usuario tendrá una recepción negativa de la señal (Wang, 2015). Por tal motivo, es una característica importante para los proveedores de contenido y servicios debido a que influye en la calidad del servicio basándose en la satisfacción del cliente.

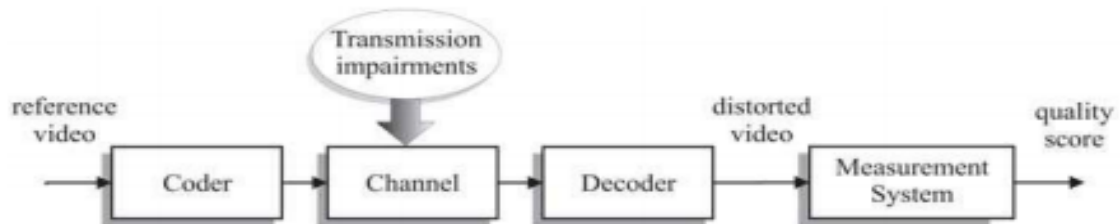


Imagen 7. Diagrama de bloques de un sistema VQM

Como se observa en la imagen 13, la evaluación que se realiza para determinar la métrica de la calidad del video es un conjunto de secuencias evaluadas objetivamente o subjetivamente, esto depende si se basan en modelos matemáticos o pidiendo la opinión de los usuarios respectivamente. La ventaja de esta característica es que puede ser evaluada sin tener acceso a la red, simplemente realizando pruebas en un laboratorio con servicios instalados o simplemente monitoreando para asegurar cierto nivel de calidad.

2.6. UNIDAD NAL

La unidad NAL o más conocido como la capa de abstracción de red, es la parte del estándar H.264/AVC utilizada para la paquetización. El principal objetivo de esta unidad es facilitar y dar entendimiento del contenido del archivo, en cuanto al soporte que se tiene de almacenamiento y de la transmisión de la información. Este soporte se puede realizar tanto en dispositivos audio-visuales como en distintas aplicaciones en las cuales intervenga una comunicación P2P (video conferencias) o también para aplicaciones en las cuales no se mantenga una comunicación, como almacenamiento (Thomas Wiegand, 2003).

Las mejoras y eficiencias de la tasa de transmisión se han visto en mejoría debido a las unidades NAL, ya que se analiza la información del archivo para maximizar la eficiencia de la codificación y de esta manera mejorar la transmisión por la red y poder robustizar la pérdida de paquetes y la corrección de errores. Todas estas mejoras se han venido implementando en aplicaciones de telecomunicaciones en las cuales se utilicen medios audiovisuales lo que abarca redes telefónicas, redes inalámbricas y hasta el mismo internet.

Una de las ventajas significativas de la unidad NAL es que brinda la posibilidad de desarrollar nuevas aplicaciones sobre las redes, de manera que se pueden gestionar y realizar diversidad sobre estas, con el objetivo de buscar necesidades de flexibilidad y personalización. Por estas razones la unidad NAL está diseñada con el objetivo principal de proporcionar un ambiente de confianza entre las redes que gestione y garantice a una integración para el uso de la VCL, (que es parte del estándar de video de H.264/AVC dedicada a la compresión) (Thomas Wiegand, 2003).

La unidad NAL simplifica la información que se encuentra almacenada y de esta manera mapear los datos como se muestra en la siguiente imagen:

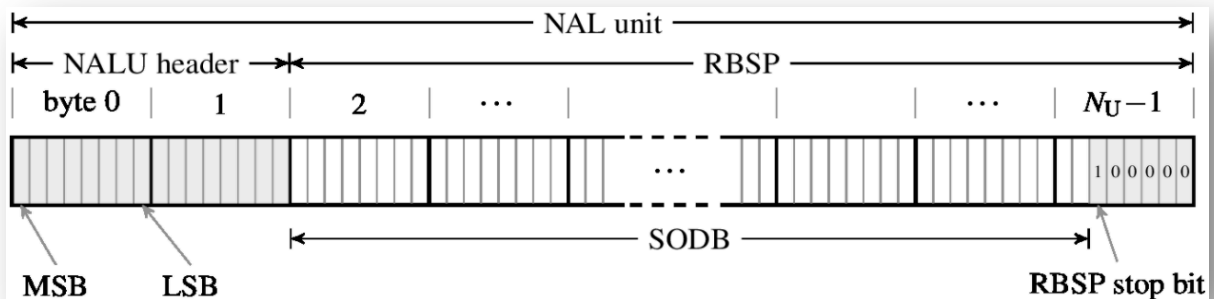


Imagen 8. Composición de las unidades NAL

La unidad NAL se divide en un encabezado de dos bytes y seguida de la carga útil sin procesar. Este encabezado incluye variedad de información, pero la que sobresale es un elemento sintaxis (NUT) el cual indica el contenido de la Unidad. El último byte de la carga útil es el bit de parada seguido de dos bits en cero "0". Esta unión de múltiples bits, antes del último bit, constituye una cadena de datos (SODB). Esta cadena representa de manera sencilla toda la información que se encuentra codificada y está lista para el proceso de decodificación.

2.7. HIGH EFFICIENCY VIDEO CODING

El mayor consumidor de ancho de banda del internet en la actualidad es el llamado video streaming. Según estudios e investigaciones realizadas para el 2015 cerca del 90% del ancho de banda que se maneja en la red es utilizado por la transmisión y descargas de archivos de video. Por esta razón, una de las iniciativas y principales motivaciones de las empresas es reducir el tamaño de los archivos para poder mejorar la navegabilidad y la portabilidad de todos los archivos; las investigaciones que realizan las empresas arrojaron resultados satisfactorios a tal punto que lograron comprimir aun mas las secuencias de imágenes, un claro ejemplo de estos estudios es HEVC (ITU- H.265) (Thomas Wiegand, 2003).

Este codificador trabajado por MPEG mejora la compresión de video y adoptó un nombre que se volvió comercial conocido como H.265, garantiza de manera eficiente mejorar la eficiencia con respecto a otros códec como H.264/AVC el cual era hasta hace unos años el más usado en Internet y los medios de almacenamiento para archivos audiovisuales.

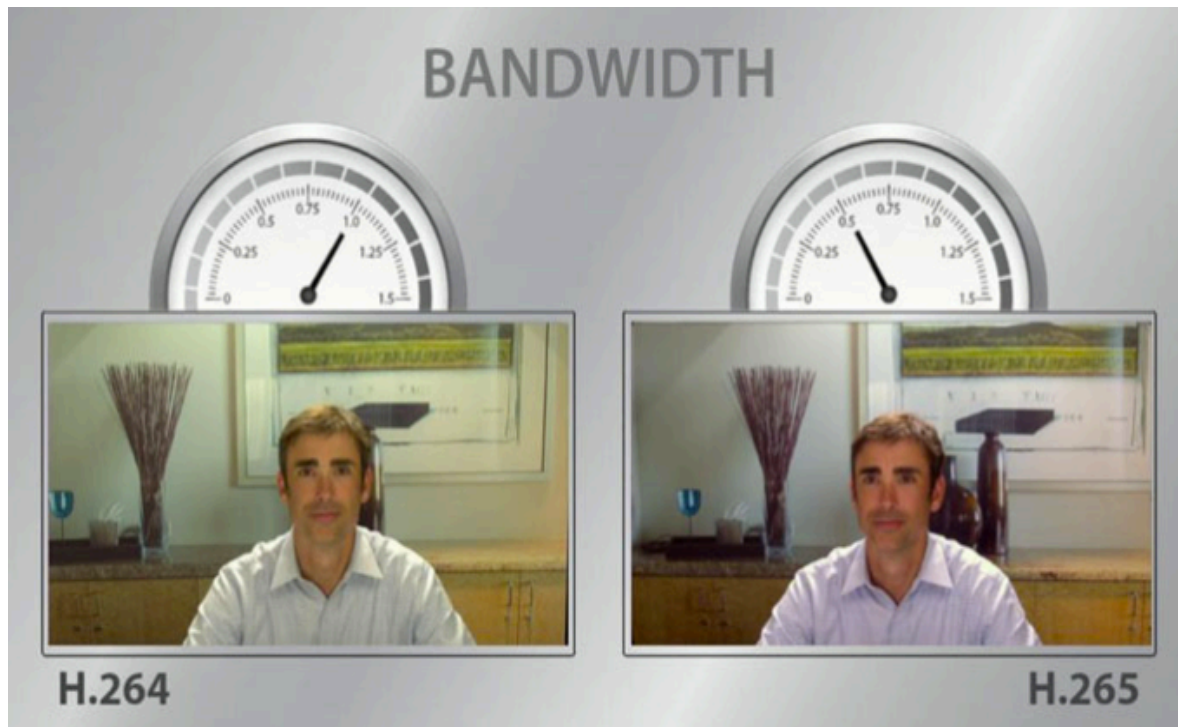


Imagen 9. Relación consumo de ancho de banda de h.264 y h.265

Lo que se esperó con la creación de este códec fue reducir de manera considerable el consumo de ancho de banda que se necesitaba para transmitir cualquier archivo de video por la red, sobre todo los de alta definición que se están manejando actualmente. De esta manera se está observando una mejora en las empresas de distribución de contenidos, principalmente en que redujeron el gasto de ancho de banda y por lo tanto podían contratar un menor servicio o aún mejor, con el mismo gasto de ancho de banda tener la capacidad de brindar una experiencia satisfactoria a los usuarios y poder reducir los tiempos de transmisión y otros servicios de multimedia.

HEVC brinda el beneficio de contener tecnologías que se encontraron bajo patentes de las organizaciones que fueron participes en la creación del codificador. Por tal motivo cualquier dispositivo que requiera la utilización del codificador tiene que adquirir o requerir una licencia, sin embargo, estas licencias deben ser razonables y no discriminatorias porque tiene la finalidad de brindar un beneficio general para cualquier tipo de persona sin importar el nivel socioeconómico.

Estas licencias asequibles han logrado que la tecnología cuente con una difusión considerable, logrando de esta manera que diferentes desarrolladores empiecen a optimizar el codificador y mejores parámetros que garanticen un beneficio para el usuario, a continuación se realizará una introducción de los parámetros de mas relevancia del codificador y una breve explicación de cada uno.

2.7.1. CODIFICACION EFICIENTE

La codificación eficiente es la capacidad de codificar cualquier tipo de archivo a una velocidad de bits mucho mas baja, teniendo en cuenta que no se debe afectar o mantener un cierto nivel en cuanto a la calidad del video. Los diseños que se presentan para los codificadores son basados en normas o formas estándar para medir la eficiencia bajo métricas como : relación señal a ruido o la subjetividad de la calidad de video.

EL codificador de HEVC genera beneficios debido a la manera como codifica, esto esta representado bajo un diagrama de bloques como se observa en la imagen 10, (pero no se realizará un análisis de cada uno de los bloques debido a la complejidad que representa la codificación), estos beneficios han sido comparados con diferentes codificadores como H.264/AVC. Estas comparaciones se realizaron con diferentes aplicaciones de entretenimiento y se hicieron a diferentes velocidades de bits. Las diferentes secuencias que se manejaron para la comprobación del codificador se encontraban en diferentes formatos y contenedores para garantizar una homogenización de procesamiento. Al finalizar las pruebas realizadas se observo una reducción de velocidad de bits basados en PSNR de aproximadamente 35,4% (Sullivan, Boyce, Chen, Ohm, & Segall, 2013).

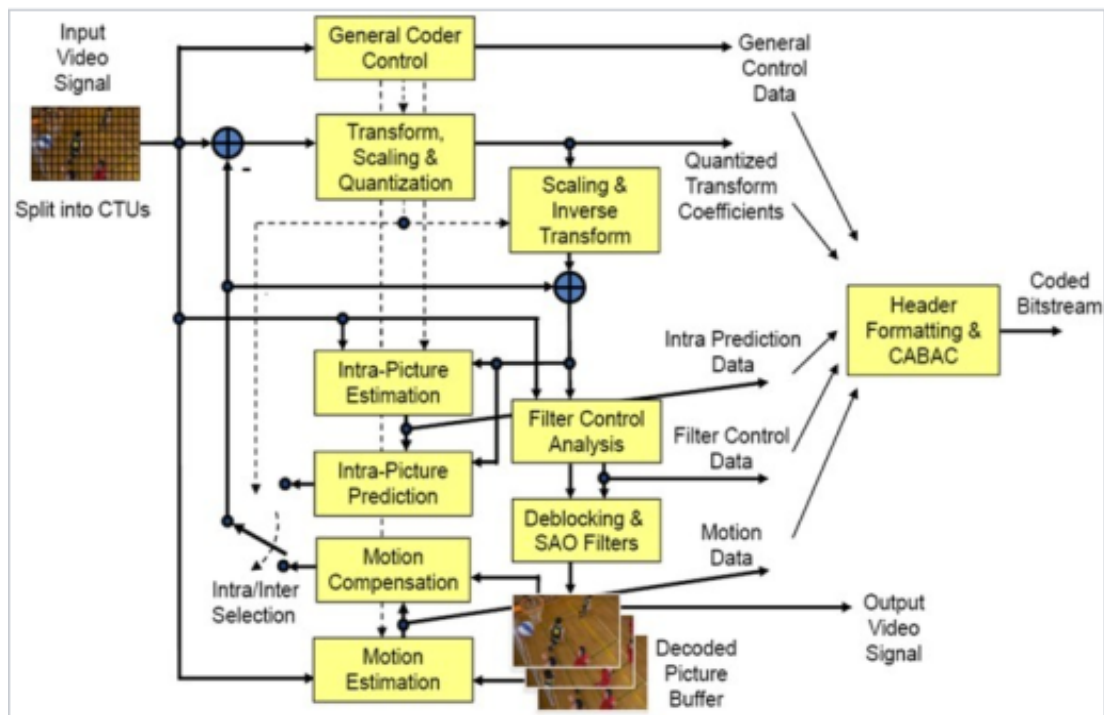


Imagen 10. Diagrama de bloques HEVC

HEVC también fue comparado en otros parámetros como lo es la calidad, sin embargo, hay que tener en cuenta que esta calidad de video es subjetiva, debido a que las pruebas se realizaron para aplicaciones se hicieron a diferentes velocidades y para calificar los resultados era necesario ampliar y sectorizar la parte de la imagen a evaluar.

2.7.2. CARACTERISTICAS

HEVC fue diseñado para mejorar considerablemente la eficiencia de la codificación sobre los demás codificadores existentes, mejoró parámetros importantes como velocidad de bits, buena calidad de la imagen, aumento de la complejidad de operación. Por lo tanto, su principal objetivo es que el video tenga una relación de compresión de datos muy elevada, teniendo en cuenta los requisitos de las diferentes aplicaciones. Esta mejora lo llevó a impulsar el codificador a pantallas de televisión, sistemas de seguridad y otras aplicaciones que conservan características similares, dentro de las cuales cabe resaltar:

- Reduce considerablemente la cantidad de espacio necesario para poder almacenar un video, debido a que H.265 reduce significativamente el tamaño del archivo.
- Por esta reducción de almacenamiento a su vez requiere menor ancho de banda para transmitir por streaming.
- Tiene la capacidad de soportar archivos en 4K y 8K, siendo el único en mercado capaz de lograrlo.
- Por su complejidad del algoritmo para la decodificación, la potencia requerida en el procesamiento es mayor. Sin embargo, justifica ese precio debido a las funcionalidades y garantías que brinda.
- Puede ser codificado tanto por Hardware como por Software, sin embargo, la segunda es mas eficiente comparativamente debido a la eficiencia y fiabilidad que brinda en cuanto a recuperación de errores.
- Reduce considerablemente la aparición de pixeles o niveles de color no deseados por una mala codificación, debido a la funcionalidad de ocultar errores.

2.8. GROUP OF PICTURES

Cuando se maneja el tema de codificación de video, vale la pena resaltar el termino de “Grupo de Imágenes” o estructura GOP, el cual es el encargado de determinar y especificar el orden que se maneja en un grupo de imágenes tipo intra e inter ordenadas adecuadamente. En otras palabras el GOP es la unidad que representa la aleatoriedad de una secuencia de imágenes que se utiliza para declarar si las imágenes son tipo I, B o P. (Marshall, 2001)

Esta secuencia de imágenes se encuentran dentro de un flujo de paquetes (por ejemplo: Unidades NAL), de tal manera que se generen de manera correcta las imágenes visibles y por lo tanto se garantice el orden adecuado de cada secuencia. A continuación se describe la definición de GOP explicado en una imagen la cual contiene los tres tipos nombrados anteriormente:

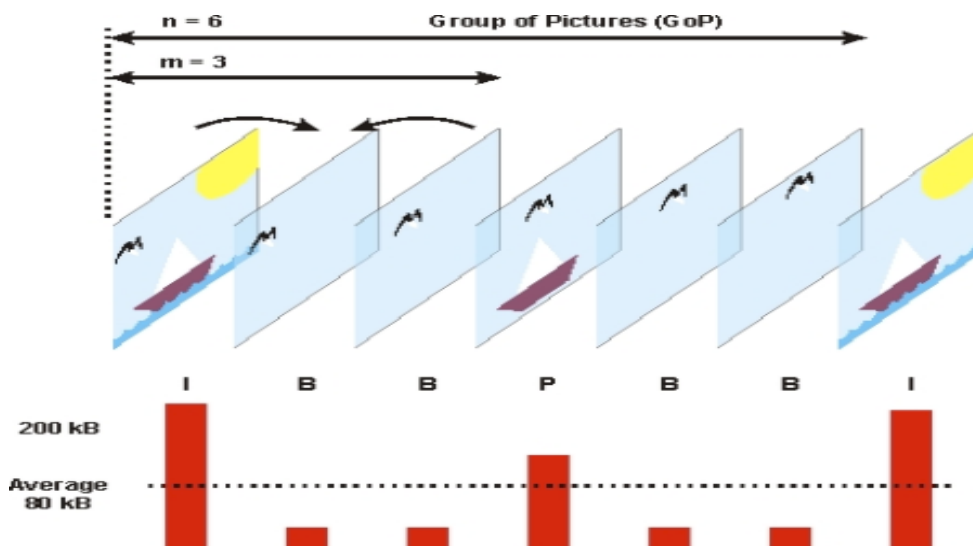


Imagen 11. Definición de GOP

En este orden de ideas, una GOP puede contener distintos tipos de imágenes:

- TIPO I (Codificación intra): es la imagen fija que representa un valor de referencia con el objetivo inicializar la secuencia, teniendo en cuenta que es independiente de los demás tipo de imágenes.
- TIPO P (Predicción): Este tipo de imagen es la encargada del movimiento, por lo tanto contiene la información de la compensación de dicho movimiento procedente de una imagen tipo P o I.

- TIPO B (Predicción bidireccional): es la encargada de separar las imágenes tipo I y P para brindar una información extra procedente de alguna de estas imágenes.
- Este tipo de imágenes son manejadas a baja frecuencia por el fin de ser utilizadas en buscadores de alta velocidad, idea que fue desarrollada por sus creadores.

Todos los GOP tienen la misma secuencia o formato, las imágenes tipo I contiene la imagen completa de tal manera que no necesitan información extra para poder recomponerla o reconstruirla. Por lo tanto, los errores que existan dentro de la estructura serán corregidos por las imágenes tipo B que pueden ser referenciadas por otras y de esta manera obtener una mayor eficiencia en cuanto a la comprensión (Marshall, 2001).

Como conclusión de este tipo de estructura, entre más imágenes tipo I existan en un flujo de información o un stream de video, será más fácil la visualización de la información, sin embargo ocupará un tamaño mayor en el archivo ya comprimido. De tal manera que para poder ahorrar ancho de banda y espacio de almacenamiento solo tiene que haber un tipo de imagen I por GOP y con los demás tipos se podrá recuperar la información y obtener la calidad adecuada.

Capítulo 3

3. CONOCIMIENTO DEL SOFTWARE

3.1. FFMPEG

FFmpeg es un software libre capaz de grabar, convertir y hacer streaming tanto de video como de audio. Todo esto es posible porque incluye librerías que contienen bibliotecas de codecs (libavcodec). Una ventaja de este software es que puede ser compilado en todos los sistemas operativos, a pesar de que se encuentre desarrollado en Linux (FFMPEG).

FFmpeg es el software líder en multimedia capaz de codificar, decodificar, transcodificar, transmitir, filtrar y reproducir casi cualquier cosa que los seres humanos hayan creado; es bastante sencillo de utilizar, debido a que esta enfocado tanto en personas con conocimientos avanzados como en personas que nunca han tenido la oportunidad de manipular el programa.

Soporta todos los formatos y es capaz de elegir cual códec utilizar simplemente escribiendo la extensión de cada uno. Compila, ejecuta y pasa la infraestructura en Linux, Mac OS X, Microsoft Windows, BSD, Solaris, etc, bajo una amplia variedad de entornos de compilación, arquitectura de maquinas y configuraciones.

3.1.1. INSTALACION DEL FFMPEG

Existe dos maneras para realizar la instalación del FFMpeg, una es de manera estática donde cada una de las librerías se encuentra implícita en el paquete de instalación y la otra de forma dinámica donde se van instalando manualmente las librerías a utilizar.

- Instalación dinámica.

La primera prueba de instalación que se realizó para este proyecto fue una instalación del FFMpeg dinámica, esto conllevó algunos problemas y errores que no permitían asociar la librería x265 (librería de uso libre) debido a que para realizarla de manera correcta era necesario que el software se encontrara en su última

versión y la librería a descargar estuviera en compatibilidad con este. Las primeras pruebas a realizar arrojaban este error:

```
MacBook-Pro-de-David-2:Cellar davidpinzon$ ffmpeg -i ejemplo.mp4 -c:v x265 salida.mp4
ffmpeg version 3.3.2 Copyright (c) 2000-2017 the FFmpeg developers
built with Apple LLVM version 8.1.0 (clang-802.0.42)
configuration: --prefix=/usr/local/Cellar/ffmpeg/3.3.2 --enable-shared --enable-pthreads --enable-gpl --enable-version3 --enable-hardcoded-tables --enable-avr
esample --cc=clang --host-cflags= --host-ldflags= --enable-libmp3lame --enable-libx264 --enable-libxvid --enable-openc1 --disable-lzma --enable-vda
libavutil      55. 58.100 / 55. 58.100
libavcodec     57. 89.100 / 57. 89.100
libavformat    57. 71.100 / 57. 71.100
libavdevice    57.  6.100 / 57.  6.100
libavfilter     6. 82.100 /  6. 82.100
libavresample  3.  5.  0 /  3.  5.  0
libswscale     4.  6.100 /  4.  6.100
libswresample  2.  7.100 /  2.  7.100
libpostproc   54.  5.100 / 54.  5.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'ejemplo.mp4':
Metadata:
  major_brand      : isom
  minor_version   : 512
  compatible_brands: isomiso2avc1mp41
  encoder         : Lavf57.71.100
Duration: 00:05:08.55, start: 0.000000, bitrate: 2019 kb/s
  Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 1280x720 [SAR 1:1 DAR 16:9], 1883 kb/s, 29.97 fps, 29.97 tbr, 30k tbn, 59.94 tbc (default
)
Metadata:
  handler_name    : VideoHandler
  Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo, fltp, 128 kb/s (default)
Metadata:
  handler_name    : SoundHandler
Unknown encoder 'x265'
```

Imagen 12. Error en la instalación de FFMPEG con librería dinámica

```
Metadata:
  handler_name    : VideoHandler
  Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo, fltp, 128 kb/s (default)
Metadata:
  handler_name    : SoundHandler
Unknown encoder 'libx265'
```

Imagen 13. Error por librerías dinámicas

Este error se debía a la inconsistencia que generaba el programa con la librería asociada, la principal causa es la incompatibilidad, la segunda la mala ubicación de la librería dentro de la carpeta de instalación lo que impedía el llamado del ejecutable de manera correcta. Se buscaron soluciones en internet de manera que no se logró corregirlo debido a que es difícil poder encontrar la librería acorde a

la versión del software, por esta razón se decidió por una instalación estática como se menciona a continuación.

- Instalación estática.

El primer paso que se realizó durante la instalación estática fue descargar el archivo de instalación para Mac (en este caso utilizamos sistema operativo de MacOs) en el enlace <http://ffmpegmac.net> (Pagina principal de ffmpeg), en el botón “download now” se empieza la descarga del instalador. Luego se descomprime el .zip y se obtiene directamente el ejecutable ffmpeg que ya contiene por defecto la librería x265 junto con dos archivos, el primero ffprobe y el segundo ffmpeg, como se muestra a continuación:



Imagen 14. Ejecutables después de la instalación

- El primero de los tres archivos es ffmpeg. Este archivo es el utilizado para realizar cada una de las pruebas del software. Tiene la función por medio de la consola del sistema operativo a realizar cada uno de los comandos que se especifiquen en la línea de descripción, entre ellas se encuentra el formato a trabajar, el bitrate, la librería y demás funcionalidades que tiene el software.
- El segundo archivo es el ffprobe, es un analizador de flujo de multimedia, tiene la función de generar todo tipo de información acerca de la entrada, la duración, la velocidad de fotogramas, el tamaño de cada uno de estos, etc. También es muy útil y sirve para recopilar información específica sobre la entrada que utilizara cualquier script que se ejecute. A continuación se muestra la información que se genera con un video que se encuentra en cierta ubicación:

```

MacBook-Pro-de-David-2:Lion_Mountain_Lion_Mavericks_Yosemite_El-Captain_15.05.2017 (1) davidpinzon$ ./ffprobe salida.mp4
ffprobe version 3.3.1 Copyright (c) 2007-2017 the FFmpeg developers
  built with llvm-gcc 4.2.1 (LLVM build 2336.11.00)
  configuration: --prefix=/Volumes/Ramdisk/sw --enable-gpl --enable-pthreads --enable-version3 --enable-libspeex --enable-libv
px --disable-decoder=libvpx --enable-libp3lame --enable-libtheora --enable-libvorbis --enable-libx264 --enable-avfilter --ena
ble-libopencore_amrwb --enable-libopencore_amrnb --enable-filters --enable-libgsm --enable-libvidstab --enable-libx265 --disab
le-doc --arch=x86_64 --enable-runtime-cpudetect
  libavutil      55. 58.100 / 55. 58.100
  libavcodec     57. 89.100 / 57. 89.100
  libavformat    57. 71.100 / 57. 71.100
  libavdevice    57.  6.100 / 57.  6.100
  libavfilter     6. 82.100 /  6. 82.100
  libswscale      4.  6.100 /  4.  6.100
  libswresample  2.  7.100 /  2.  7.100
  libpostproc   54.  5.100 / 54.  5.100
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'salida.mp4':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2mp41
    encoder          : Lavf57.71.100
  Duration: 00:05:08.55, start: 0.000000, bitrate: 889 kb/s
  Stream #0:0(und): Video: hevc (Main) (hev1 / 0x31766568), yuv420p(tv, progressive), 1280x720 [SAR 1:1 DAR 16:9], 752 kb/s,
  29.97 fps, 29.97 tbr, 30k tbn, 29.97 tbc (default)
  Metadata:
    handler_name     : VideoHandler
  Stream #0:1(und): Audio: aac (LC) (mp4a / 0x6134706D), 44100 Hz, stereo, fltp, 128 kb/s (default)
  Metadata:
MacBook-Pro-de-David-2:Lion_Mountain_Lion_Mavericks_Yosemite_El-Captain_15.05.2017 (1) davidpinzon$ █
    
```

Imagen 15. Prueba de ffprobe

- El último archivo es ffmpeg, es un servidor de streaming para audio y video, es compatible con varios canales en vivo y streaming de archivos. Esto quiere decir que si se necesita transmitir contenido de audio y video a través de internet. Este servidor servirá debido a que funciona como un sistema de difusión, es capaz de recopilar múltiples fuentes de entrada y las analiza de manera que se pueden utilizar de la manera que sea mas adecuada.

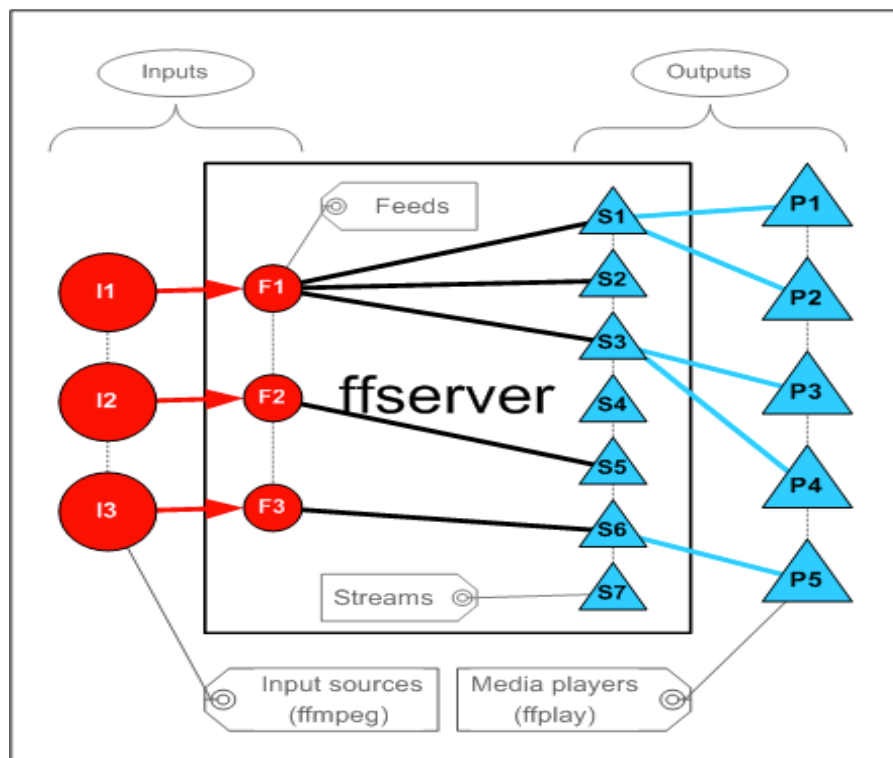


Imagen 16. Funcionamiento del ffserver

3.1.2. PRUEBAS DE COMANDOS BASICOS DEL SOFTWARE

Para el desarrollo del proyecto se empezó con el reconocimiento del programa en donde se realizaron pruebas básicas, entre las cuales se encuentran: cambiar el formato del video y cambiar la duración del video para agilizar pruebas, entre otras.

- Cambiar duración:

El primer paso a realizar es descargar un video de alta definición de cualquier página web, con el objetivo de manipular este para las múltiples pruebas y cambios a realizar. El video descargado tiene una duración de 7:05 minutos lo cual para alguna pruebas es un tiempo bastante largo debido a los diferentes algoritmos que utilizan los distintos codificadores. Para recortar el video a un tiempo considerable se utiliza el siguiente comando:

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i Video_Original.mp4 -acodec copy -vcodec copy -t 25 Video_salida.mp4
```

Imagen 17. Comando para recortar video a un determinado tiempo de duración

El “./ffmpeg” lo que hace es ejecutar el programa que se descargó anteriormente, “-i Video_original.mp4” es el video de entrada que se manejará durante la ejecución del programa, “-acodec copy -vcodec copy” se determina de esta manera con el fin de simplemente utilizar el mismo formato de video y de audio del video original y solo manejar el tiempo, “-t 25” es el tiempo en segundos al cual se quiere recortar el video y por ultimo se nombra el video salida con el cual se quiere renombrar.

- Cambiar formato.

Para cambiar el formato del video se utiliza un comando similar al descrito en la imagen 17, sin embargo, presenta variaciones debido a que solo se tendrán en cuenta los formatos x264 y x265, contenidos en mp4 y mkv que son los mas utilizados hoy en día. El comando a utilizar es el siguiente:

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i Video.mp4 -c:v libx265 Video_x265.mkv
```

Imagen 18. Comando para convertir el video a un formato diferente

Los resultados de estas pruebas arrojaron resultados muy similares en cuanto a la calidad de la imagen, sin embargo, presentan variaciones con respecto al tamaño de los archivos. Los resultados en cuanto a imagen y tamaño se denotarán a continuación.



Imagen 19. Izquierda x264 con mp4 ; Derechas x265 con mp4

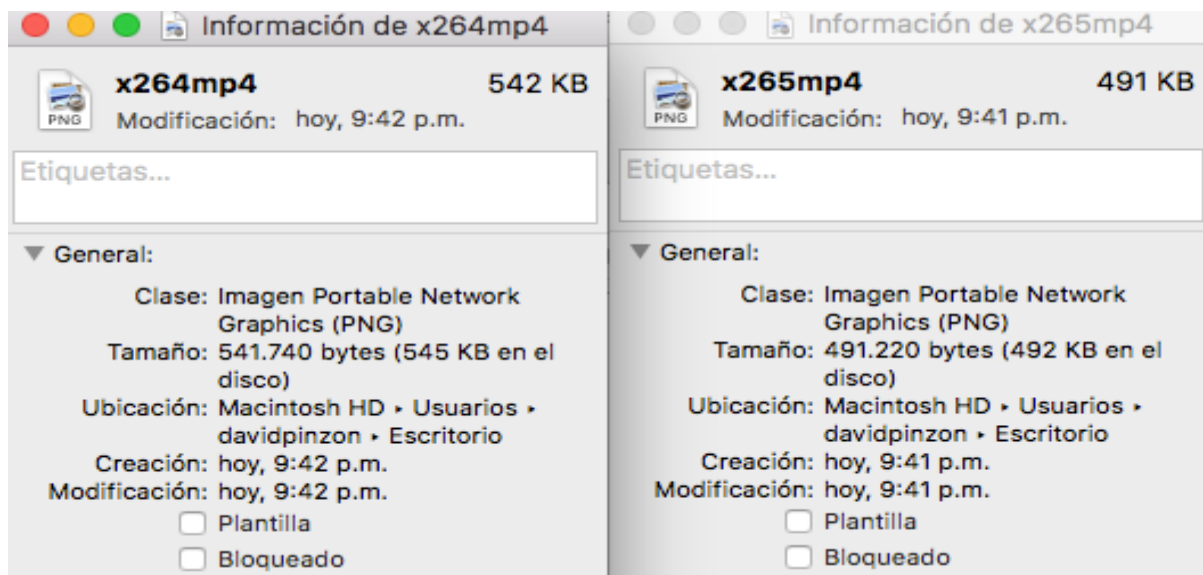


Imagen 20. Comparación de tamaño para las imagen 19

Como se denota en la imagen anterior, la diferencia en tamaño en cuanto a los dos archivos se alcanzan a distinguir, x265 es mas pesada que x264 y como se observa en la imagen 19, en cuanto a calidad no se determina, por el contrario, se

alcanza a visualizar una intensidad en los colores del x265 debido a los algoritmos que utiliza para la codificación.



Imagen 21. Izquierda x264 con mkv ; Derecha x265 con mkv

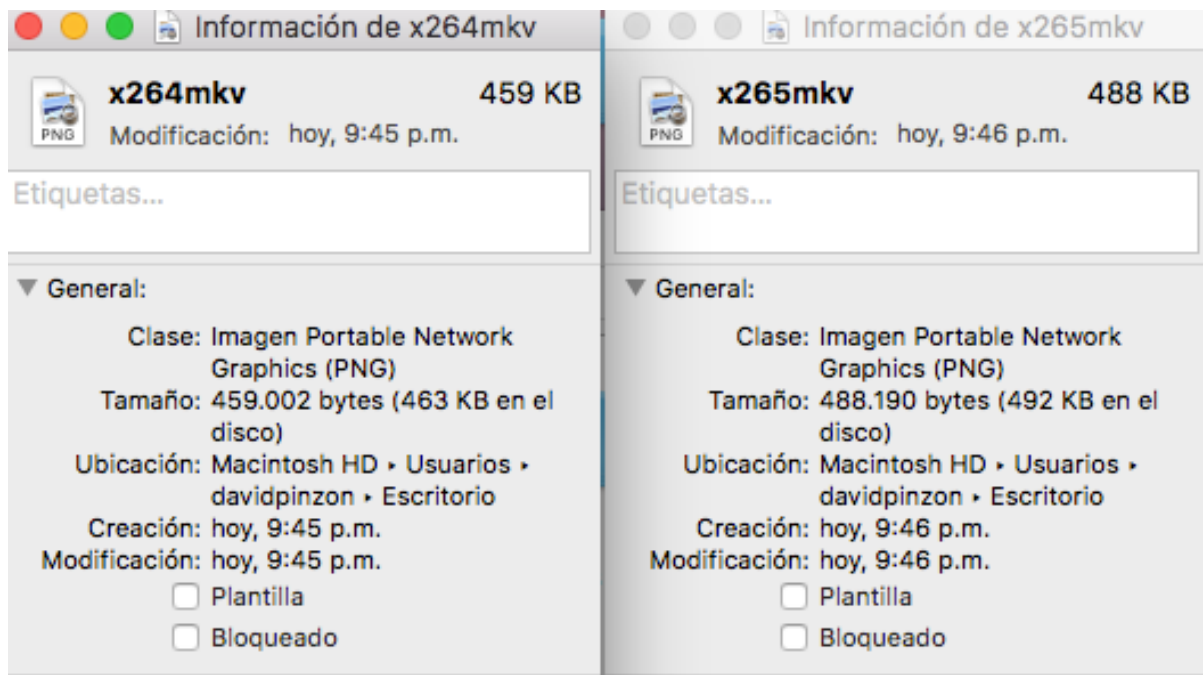


Imagen 22. Comparación de tamaño de la imagen 21

A comparación de las pruebas anteriores se observa una calidad muy similar, sin embargo, la imagen de la derecha que corresponde a x265 es un poco mas definida en sus colores, pero a diferencia de la otra prueba, en este caso el archivo es un poco más pesado debido a que mkv es un contenedor caracterizado por resaltar detalles de alta calidad lo que hacen que la imagen tenga esa resolución y de esta manera sea un poco más pesado el archivo.

Capítulo 4

4. DESARROLLO DEL PROYECTO

Después de haber evaluado la importancia de los paquetes en la red y analizar las diferentes características que se pueden observar con la unidad NAL, se continúa con modelos basados en la red para utilizar paquetes con todos los datos codificados. A continuación se evaluará la idea de utilizar sistemas computacionales de bajo nivel y, de esta manera, características modificables que garanticen optimizar y garantizar el objetivo general del TFM.

4.1. FACTORES QUE AFECTAN LA CALIDAD

Cada paquete que es transmitido por la red puede ser entendido como un impacto en la calidad visual que observan los usuarios de la transmisión. Por tal motivo se evaluará diferentes características para mejorar y hacer más robusto el sistema de transmisión.

4.1.1. Constante QP (CQP)

La constante de QP es un parámetro de cuantización el cual es el encargado de controlar la cantidad de compresión por cada macrobloque en un marco. A medida que el valor es mas grande significa que habrá una mayor cuantización, mas compresión y menor calidad. Por el contrario, los valores mas bajos significan lo opuesto a lo anteriormente mencionado. El rango de valores que maneja el parámetro QP oscilan entre 0 y 51, de tal manera que existe la posibilidad de establecer un valor fijo para el proceso de codificación.

COMPARACIÓN CODIFICACION CONSTANTE QP	
	QP=1
	QP=23
	QP=50

Tabla 1. Comparación constante QP

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i Video.mp4 -c:v libx265 -x265-params qp=50 out2.mp4
```

Imagen 23. Comando para modificar constante QP

Modificando el anterior comando en la parte que dice “qp” se puede variar la constante y de esta manera mejorar o disminuir la calidad de la codificación y por lo tanto la manera de visualizar el video con alta resolución, como se muestra en la tabla 1, donde se realizan tres pruebas con valores de la constante qp diferentes (1, 23 y 50) para mostrar una comparación en cuanto a resultados.

Este método es muy usado en la investigación de la codificación de video, pero si solamente se quiere realizar esto, es efectivo el método ya que, al momento de establecer una constante qp fija lo que se hace es variar el bitrate dependiendo de la complejidad de la escena.

4.1.2. Promedio de velocidad de bits (ABR)

El promedio de velocidad de bits es un parámetro que sirve para calcular la mejor velocidad para codificar el archivo de salida, esta velocidad es basada bajo patrones como duración del archivo, contenedores, y codificadores utilizados. Por tal motivo, se puede ajustar estas velocidades, regulándolas de manera que la velocidad promedio sea relativa a la cantidad de bits a procesar.

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i Video.mp4 -c:v libx265 -b:v 10K videorate.mp4
```

Imagen 24. Comando para variar velocidad de bits

Con el comando de la imagen 24 lo que se logra es variar la velocidad de bits de la codificación y de esta manera interactuar con esta. Entre mas pequeña sea esta velocidad la codificación se hará con una velocidad menor disminuyéndole calidad a la resolución del video. Para obtener una calidad buena se puede considerar por encima de 1M como velocidad.

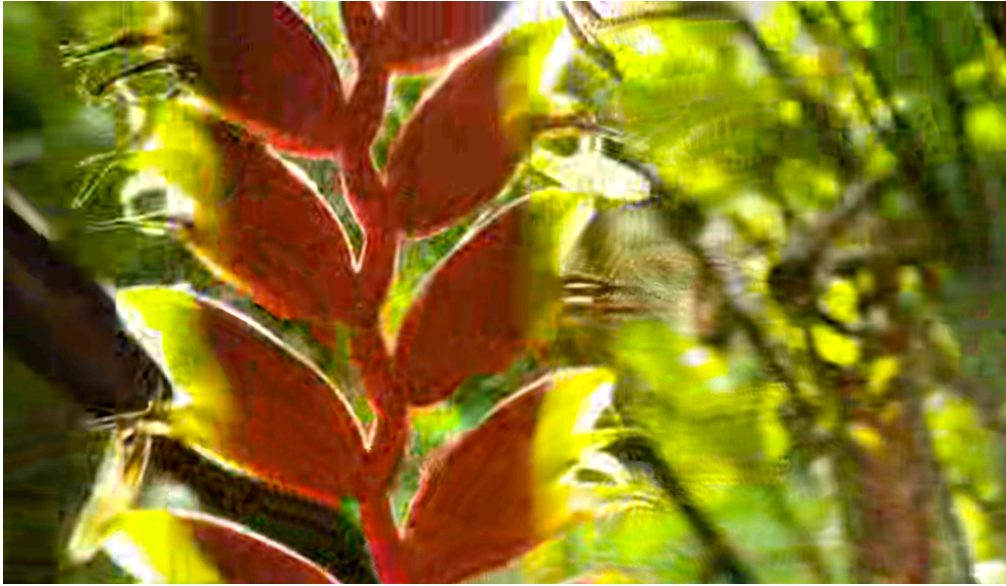


Imagen 25. Codificación con velocidad de bits 10K



Imagen 26. Codificación con velocidad de 1M

ABR también es muy utilizado para las conexiones en donde se necesite transmitir poco tráfico debido a que no existe una dependencia en cuanto a la sincronización de tiempo entre el tráfico, ancho de banda o latencia. Es especial para TCP debido a que se puede modificar el comportamiento de transmisión con el manejo de un control de frecuencia en cuanto a la velocidad de bits.

Teniendo en cuenta opiniones de muchos desarrolladores, se recomienda usar el método anterior, debido a que el codificador desconoce en su totalidad lo que esta delante del tiempo, por lo tanto, tendrá que adivinar como llegar al bitrate que se especifique. Lo anteriormente dicho significa que la velocidad variará especialmente al principio de la codificación del video original alcanzando en algún punto su objetivo principal.

4.1.3. Velocidad binaria constante (CBR)

La velocidad binaria constante esta diseñada con el objetivo de trabajar de manera ideal en varios escenarios de transmisión. De esta manera, se puede restringir la velocidad de bits con el objetivo de garantizar una reproducción coherente. Esta coherencia se ve determinada en las partes de los contenidos en donde es mas difícil comprimir, debido a que existe partes de menor calidad en donde no se necesita tanto tiempo de compresión a comparación de otras, por tal motivo se generan variaciones mas pronunciadas teniendo en cuenta una velocidad constante.



Imagen 27. Velocidad binaria constante de 10K con buffer de 10K

La velocidad binaria constante es utilizada en conexiones las cuales sean utilizadas para transmitir archivos multimedia a una velocidad de bits constante, donde no se tiene en cuenta factores como la sincronización entre los destinos. Tiene la ventaja de una adaptabilidad en cuanto a los datos que necesiten una

respuesta predecible y una cantidad predeterminada de ancho de banda. Por tal motivo este parámetro es muy útil en aplicaciones como telefonía, videoconferencias, etc, debido a que proporciona una latencia muy baja y por lo tanto se puede predecir muchos datos.



Imagen 28. Velocidad binaria de 1M con buffer de 10K

De las anteriores imágenes se puede concluir que el buffer de almacenamiento de la codificación no influye para nada siempre y cuando se respeten unas mínimas en el tamaño del buffer, solamente se tiene en cuenta la velocidad constante que se determine, teniendo en cuenta que es necesario indicar un mínimo y un máximo, como se muestra en la siguiente imagen:

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i Video.mp4 -c:v libx265 -x265-params "nal-hrd=cbr:force-cbr:force-cfr=1" -b:v 1M -minrate 1M -maxrate 1M -bufsize 2M binaria.mp4
```

Imagen 29. Comando para ajustar velocidad constante

Este método es bastante útil siempre y cuando la fuente original sea fácil de codificar, es decir no tenga tanto parámetros que dificulten la codificación, ya que con CBR se garantiza una velocidad de transmisión constante, por lo cual es muy

utilizado en streaming, donde generalmente se desea permitir que los flujos en ocasiones utilicen una velocidad de bits menor y estable.

4.1.4. Calidad constante (CQ) / Factor de frecuencia constante (CRF)

CRF oscila como la constante QP entre 0 y 51, teniendo en cuenta que un valor predeterminado para x265 es 28. Cualquier resultado inferior la probabilidad es que solo desperdicie el tamaño del archivo mostrando como resultado la mitad o el doble de la velocidad de bits del archivo original, teniendo como problema que no se tendrá claro cual será el tamaño del archivo resultante.

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i Video.mp4 -c:v libx265 -crf 28 frecuencia.mp4
```

Imagen 30. Comando para modificar la frecuencia constante CRF

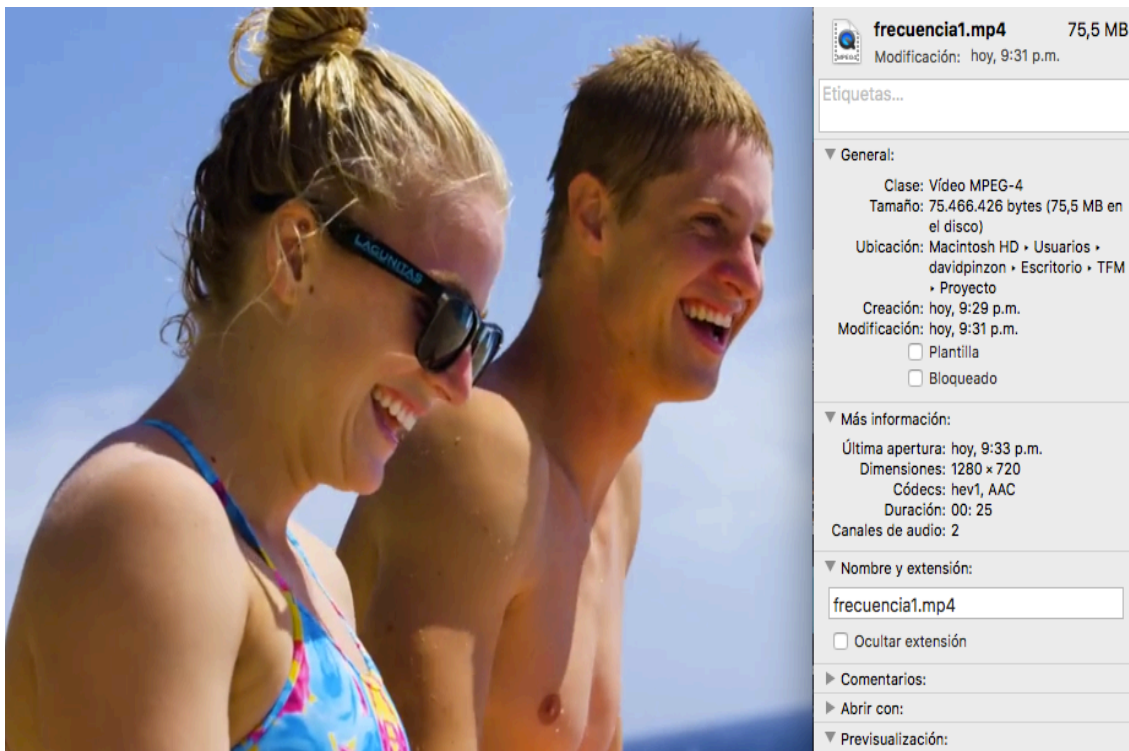


Imagen 31. Frecuencia constante CRF=1

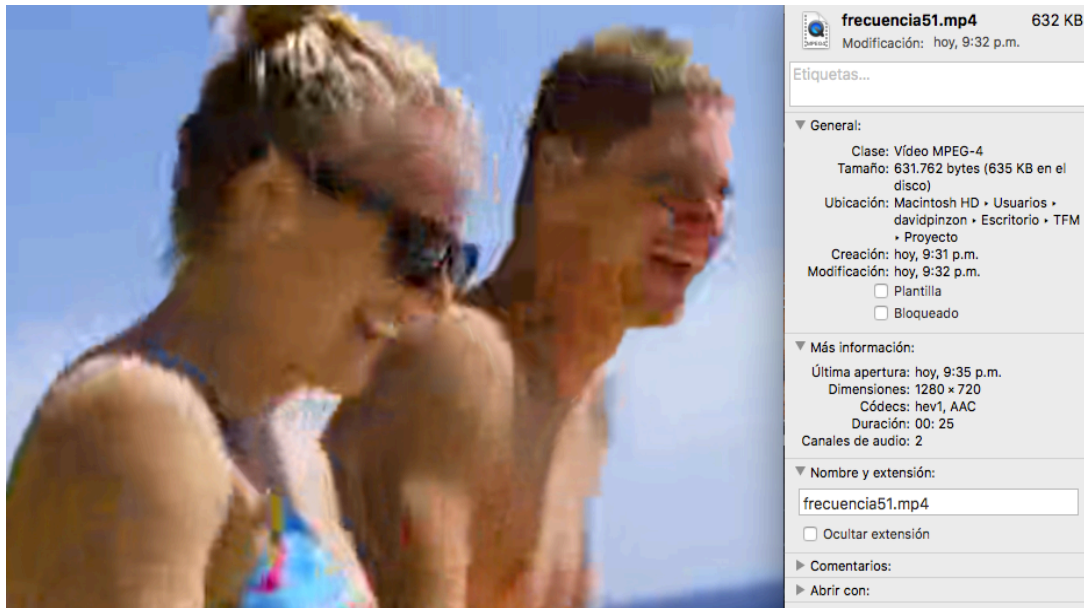


Imagen 32. Frecuencia constante CRF=51

Teniendo en cuenta las dos pruebas realizadas, una con una frecuencia de 1 y la otra de 51, se denota con claridad la calidad de la imagen. Por supuesto se había nombrado anteriormente que el tamaño del archivo no es posible controlarse con este método, entre menos frecuencia, por la calidad de su imagen, el peso del archivo es mucho mayor al de menor frecuencia, por esta razón, se puede concluir que CRF solo se tiene en cuenta para especificar la calidad que se desea sin importar el peso del archivo resultante.

4.1.5. Codificación restringida (VBV)

El Video Buffering Verifier proporciona la manera en la cual se puede asegurar que la velocidad de bits se encuentra limitada a un máximo. De esta manera es muy útil para la transmisión, por lo que en la transmisión se puede garantizar que no se enviarán mas bits de lo necesario en un determinado tiempo, superando el ancho de banda disponible del canal. Este método se puede combinar con otros para hacerlo mas efectivo garantizando la velocidad y una frecuencia adecuada como las presentadas anteriormente.

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i Video.mp4 -crf 28 -x265-params vbv-maxrate=1000:vbv-bufsize=2000 maximo.mp4
```

Imagen 33. Comando para transmitir el número máximo de bits



Imagen 34. Máximo número de bits 1000 con buffer de 2000

Con las opción vbv que se muestra en la imagen 33, se puede manejar dos características importantes en este método el “-maxrate” y el “-bufsize”, que se utilizan principalmente para establecer la velocidad de bits máxima y el tamaño del buffer del cliente esperado en cuanto la transmisión. Un tamaño que se consideraría adecuado sería que el buffer sea el doble a la velocidad máxima, sin embargo, estas sugerencia varían dependiendo de la transmisión y de la capacidad de la transmisión.

4.2. TRANSMISION DE FLUJOS DE VIDEO (STREAMING)

FFmpeg brinda dos maneras para transmitir video: El primero por medio de un servidor (con el cual ffmpeg cuenta) y de esta manera sirve como multidifusión para que la cantidad de receptores sea mayor; y el segundo es por vía UDP / TCP con un único receptor o directamente a un destinatario que brinde la posibilidad de multidifusión. Este servidor incorporado brinda posibilidades de transmisión muy buenas las cuales serán utilizadas en esta parte del desarrollo del proyecto.

El servidor de FFmpeg es una posibilidad adecuada para la transmisión de archivos, incluso el reproductor VLC tiene la manera de obtener el flujo de datos de ffmpeg y de esta manera distribuirlo porque brinda la posibilidad de trabajar como servidor. Sin embargo, ffmpeg es mas eficiente que el reproductor de multimedia debido a que el software realiza transcodificación y de esta manera puede comparar los datos transmitidos para valorar la carga útil de las unidades NAL.

Las primeras pruebas realizadas constan principalmente del envío de un archivo llamado Original.mp4 el cual será enviado por medio del protocolo RTP a la dirección IP 127.0.0.1 y puerto 5600. Junto con la información anterior se utilizan algunos otros parámetros como latencia, escala de la visualización y codificador a utilizar, entre otros. A continuación se muestra el comando utilizado para la transmisión del archivo:

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -re -f lavfi -i smptebars -vcodec libx265 -tune zerolatency -vf scale=1920:1080 -b 900k -sdp_file Original.mp4 -f rtp "rtp://127.0.0.1:5600"
```

Imagen 35. Comando para enviar por RTP un archivo de video

Al ejecutar el anterior comando, se tiene la garantía de que el archivo fue enviado por RTP, sin embargo como se utiliza la funcionalidad del servidor este queda enviando el archivo continuamente hasta ser detenido manualmente. Este envío continuo es con el fin de garantizar que cada uno de los paquetes llegue de forma correcta por medio de la comprobación de frames y haciendo un barrido de cada uno de las unidades NAL que se envían.

Durante la transmisión del archivo se puede observar información de este, de tal manera que se tenga con claridad los parámetros con los cuales se está realizando el streaming. En la imagen 36 se observa los datos que se describen durante el proceso:

```

Stream #0:0 -> #0:0 (rawvideo (native) -> hevc (libx265))
Press [q] to stop, [?] for help
x265 [info]: HEVC encoder version 2.4+14-bc0e9bd7c08f
x265 [info]: build info [Mac OS X][clang 4.2.0][64 bit] 8bit
x265 [info]: using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2 AVX
x265 [info]: Main profile, Level-4 (Main tier)
x265 [info]: Thread pool created using 4 threads
x265 [info]: Slices                : 1
x265 [info]: frame threads / pool features      : 2 / wpp(17 rows)
x265 [info]: Coding QT: max CU size, min CU size : 64 / 8
x265 [info]: Residual QT: max TU size, max depth : 32 / 1 inter / 1 intra
x265 [info]: ME / range / subpel / merge      : hex / 57 / 2 / 2
x265 [info]: Keyframe min / max / scenecut / bias: 25 / 250 / 0 / 5.00
x265 [info]: Lookahead / bframes / badapt     : 0 / 0 / 0
x265 [info]: b-pyramid / weightp / weightb      : 0 / 1 / 0
x265 [info]: References / ref-limit cu / depth : 3 / on / on
x265 [info]: AQ: mode / str / qg-size / cu-tree  : 1 / 1.0 / 32 / 0
x265 [info]: Rate Control / qCompress            : ABR-900 kbps / 0.60
x265 [info]: tools: rd=3 psy-rd=2.00 rskip signhide tmpv strong-intra-smoothing
x265 [info]: tools: lsllices=6 deblock sao
Output #0, rtp, to 'rtp://127.0.0.1:5600':
  Metadata:
    encoder      : Lavf57.71.100
    Stream #0:0: Video: hevc (libx265), yuv420p, 1920x1080 [SAR 3:4 DAR 4:3], q=2-31, 900 kb/s, 25 fps, 90k tbn, 25 tbc
  Metadata:
    encoder      : Lavc57.89.100 libx265
frame= 716 fps= 16 q=-0.0 size= 209kB time=00:00:28.52 bitrate= 60.1kbits/s speed=0.652x

```

Imagen 36. Características del archivo enviado por RTP

Para que el archivo de multimedia pueda ser reproducido, se utilizó VLC el cual tiene la funcionalidad de ser utilizado como servidor y como reproductor, de tal manera que para abrir el archivo transmitido solo se debe utilizar la función de “abrir archivo de red” y seleccionar los parámetros que se encuentran descritos en la imagen 37.

Protocolo

UDP

RTP

Modo

Unidifusión

Multidifusión

Dirección

Puerto

Dirección IP Puerto

Si quiere abrir una emisión multidifusión («multicast»), introduzca la dirección IP proporcionada por el proveedor. En un modo unidifusión («unicast»), VLC usa la dirección IP de su máquina automáticamente.

Para abrir una emisión usando un protocolo diferente, presione Cancelar para cerrar esta hoja.

Imagen 37. Asociación de la dirección IP y puerto del streaming

Con estos parámetros estamos garantizando la dirección IP proporcionada por el proveedor, en este caso quien la proporcionó es el ffserver; de esta manera se

garantiza una multidifusión del archivo. Se utiliza el puerto 5600/TCP y de esta manera se tiene un control entre los “dispositivos” que se encuentran dentro de la red para que pueda proporcionar la fiabilidad del streaming y de esta manera la calidad del video se encuentre en su máxima expresión.

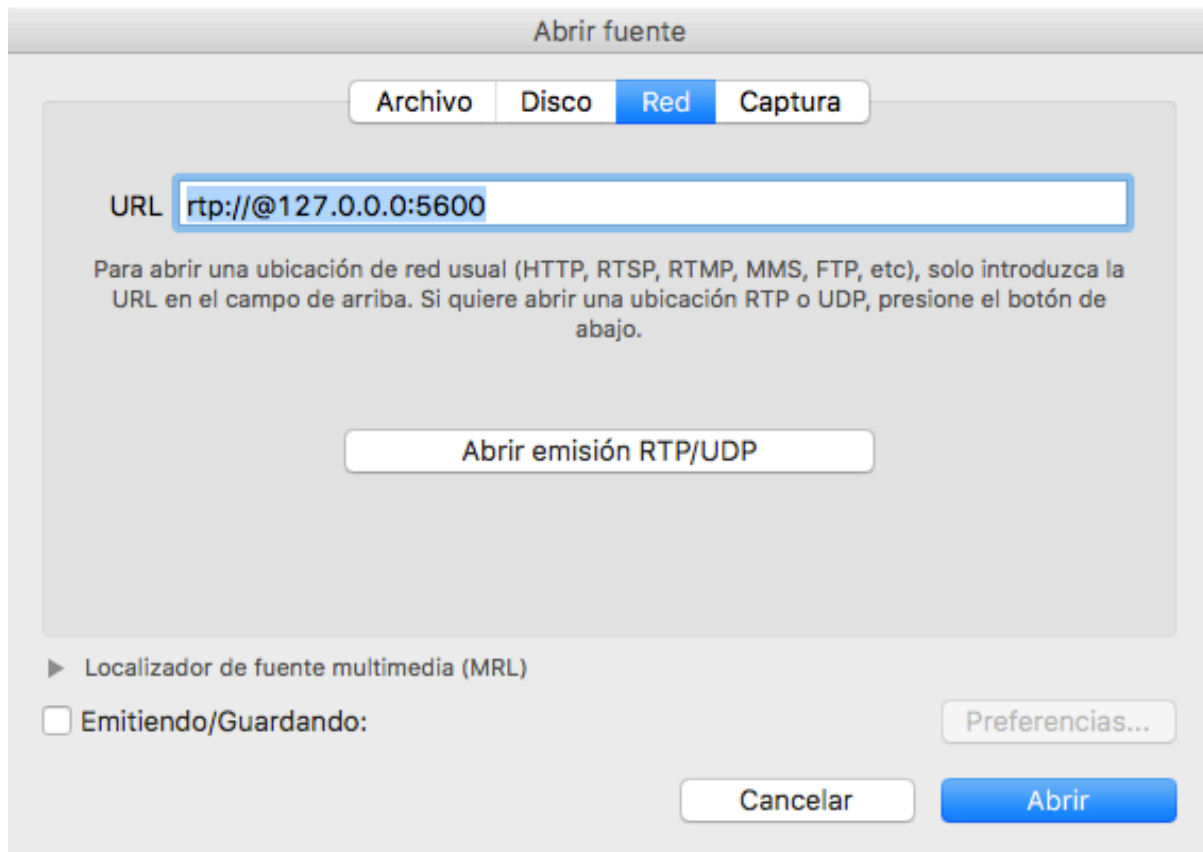


Imagen 38. URL del archivo enviado por streaming

Una vez se le de click en abrir, se realizará la emisión del archivo de tal manera que queda en la lista de reproducción, por lo tanto, puede ser visualizado con una calidad en alta resolución y de esta manera se garantiza que todo lo descrito anteriormente funciona de forma correcta y la emisión de un archivo por medio del comando de ffmpeg utilizando el protocolo RTP se puede realizar brindando cualidades de codificación, espacio de almacenamiento y otros factores que hacen que H.265/HEVC sea una buena opción para la transmisión de archivos para streaming.

Título	Autor	Duración	
Original.mp4		00:00	
Original.mp4		00:40	
Original.mp4		00:40	
rtp://127.0.0.1:5600		00:40	

Imagen 39. Lista de reproducción en reproductor VLC

4.3. PERDIDA DE UNIDADES NAL

Las unidades NAL como se nombró anteriormente es la parte estándar de la codificación H.264/AVC que fue acogida por H.265/HEVC con el objetivo de parametrizar datos y tener un determinado orden en la composición de las Unidades transmitidas. En esta parte del proyecto se simula estas unidades como paquetes de red, de tal manera que para poder robustizar el codificador fue necesario eliminar algunas de estas unidades paulatinamente para observar los cambios al decodificar con cierta cantidad de unidades eliminadas. Fue un proceso que se realizó manualmente utilizando una herramienta llamada hevcbrowser, la cual nos sirve para mirar la composición y estado de cada una de las unidades.

La interfaz de este software es muy sencilla y amigable, cuando se ejecuta observamos una pantalla como la de la imagen 45, en ella solamente se pueden realizar acciones como abrir un archivo de video el cual debe estar en un formato en específico, ejecutando el comando de la imagen 45 podemos realizar el análisis del video en cualquier tipo de contenedor debido a que la función de este es desmultiplexar el archivo y podemos observar cada parámetro de raíz.

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i Original.mp4 -vcodec copy -an video.h265
```

Imagen 40. Comando para pasar a modo lectura de hevcbrowser

Una vez tengamos el archivo en el formato necesario para su lectura simplemente se da click en “OPEN” para buscarlo y de esta manera se pueda ver su composición.

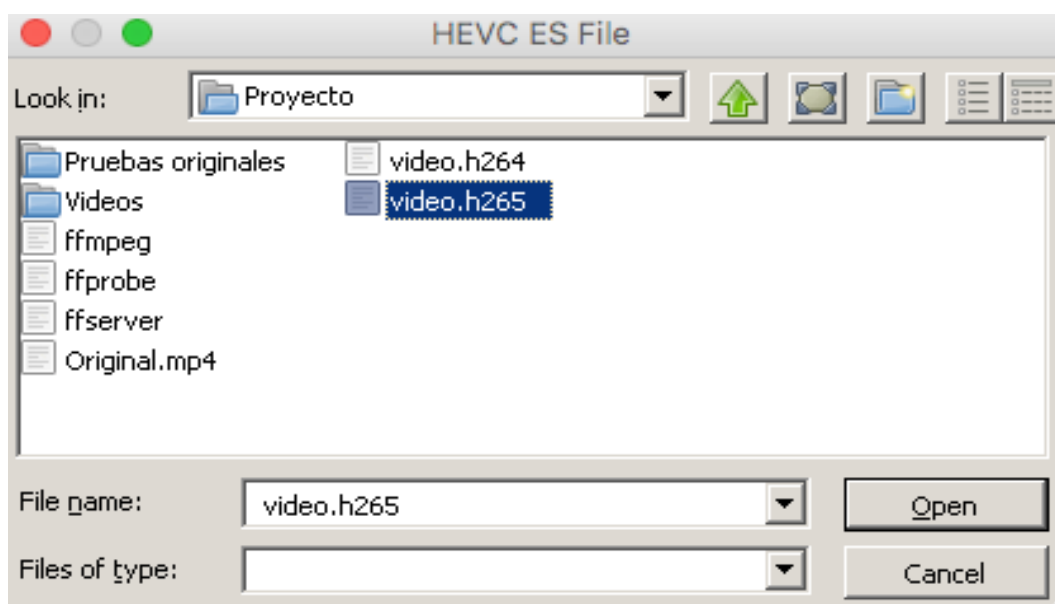


Imagen 41. Archivo en formato lectura hevcbrowser

Cuando se abre el archivo podemos observar 3 pantallas, como se muestra en la imagen 47:

- La pantalla de la parte superior izquierda, determina el nombre de la unidad en ella se especifica su longitud, el tipo de la unidad y la información que nos explica que va contenida en cada una de ellas.
- En la parte inferior izquierda tenemos las unidades especificadas en bits para poder tener una descripción más específica para quienes quieran modificarlas directamente manejando bits.
- En la parte izquierda tenemos la pantalla donde nos especifica cada unidad NAL en parámetros más específicos los cuales nos sirven para

determinar la composición y los valores exactos con los que contamos en cada unidad.

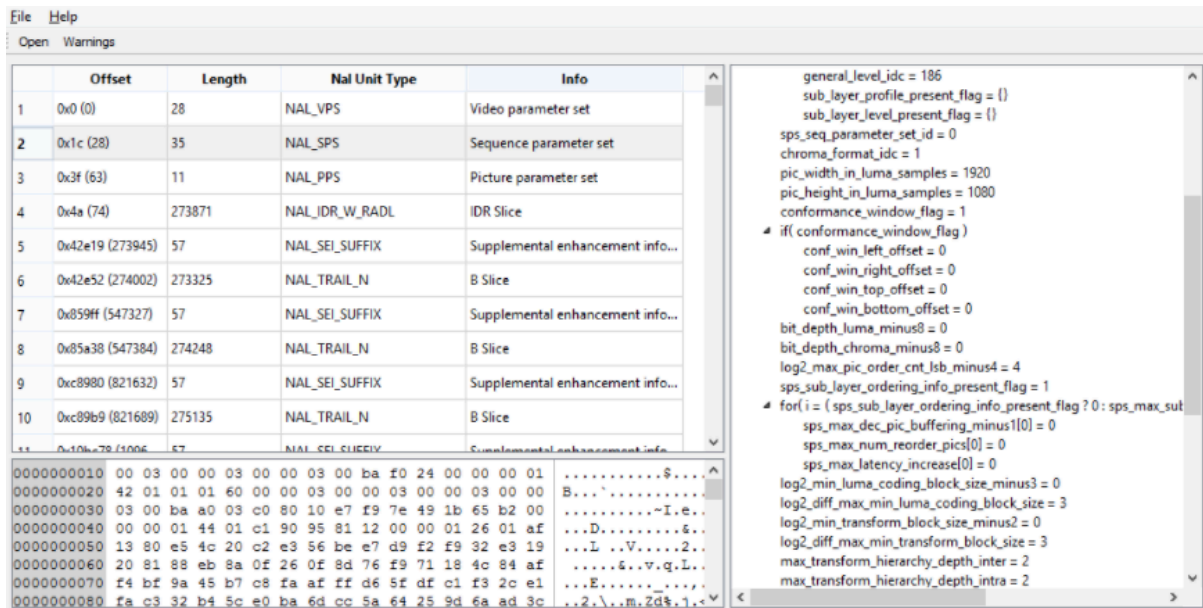


Imagen 42. Interfaz de cada archivo abierto por hevcbrowser

Para nuestra facilidad el software permite eliminar las unidades manualmente sin necesidad de utilizar un tipo de código u otras aplicaciones, para ello simplemente se selecciona la unidad que queremos y darle suprimir se eliminara la unidad del archivo. Luego de ello se guarda el archivo y se procederá a ejecutar el comando para recuperar el archivo en el formato inicial.

```
MacBook-Pro-de-David-2:Proyecto davidpinzon$ ./ffmpeg -i video.h265 -vcodec copy -an Original1.mp4
```

Imagen 43. Comando para restablecer al formato original

Las pruebas fueron realizadas con perdidas de hasta 5% aproximadamente de las unidades NAL, teniendo en cuenta que el archivo que se descompuso tenia 1250 unidades NAL y se eliminaron un máximo de 80 unidades aleatoriamente sin discriminar si era GOP o parámetros de imagen u otro tipo de unidad.

4.4. RECUPERACION DE LA INFORMACION PERDIDA

El método utilizado para la recuperación de la información perdida, es basada una herramienta llamada MEncoder que brinda y garantiza la sobre posición de imágenes para la recuperación visual de los errores. Se utilizó este método porque garantiza una calidad de salida del archivo bastante considerable sin modificar el tamaño de este, recuperando las unidades NAL que fueron eliminadas anteriormente. Por supuesto existen varios métodos a utilizar pero el que brindó una mejora de relación calidad/almacenamiento/procesamiento fue esta herramienta. A continuación se da una explicación del funcionamiento de la herramienta y la manera como se utilizó:

4.4.1. MEncoder

MEncoder es una herramienta gratuita de comandos para la transcodificación de archivos multimedia con el objetivo de convertir todos los formatos multimedia comprimidos y sin comprimir utilizando diferentes tipos de codecs. Tiene la capacidad de construirse a partir del mismo código codificado y de esta manera puede leer muchas fuentes de formatos conocidos, gracias a ser una librería abierta y en continua mejora.

Tiene la capacidad de brindar particularidades que sirven para la corrección de errores, debido a que permite analizar la salida del archivo y de esta manera filtrar varios parámetros para volverse a ejecutar. Por ejemplo si el sistema no se puede procesar en tiempo real, brinda la capacidad de desactivar el audio y de esta manera solo enfocar el análisis de la información en los frames de video.

Su funcionamiento es muy sencillo y útil debido a que utiliza un cambio de velocidad en los fotogramas y de esta manera salta cuadros, para crear efectos rápidos de velocidad como sucede en las películas. Por tal motivo, al ojo humano no es posible determinar el error. En caso de que el error sea visible se utiliza unas imágenes entrelazadas para crear dos marcos diferentes de cada fotograma de video y así sobreponer información y recuperar en el caso de que fuese necesario.

Gracias a este mecanismo de funcionamiento la reproducción en pantallas progresivas garantizan una resolución de alto nivel y convierte el flujo de video utilizable de fotogramas y crea escenas como si estuviera en cámara lenta a partir de las imágenes controladas con los cambios de velocidad. Por ejemplo, si la información se llegase a perder, los filtros adecúan las imágenes hasta el punto de obtener a la salida un movimiento de vibración creado por dos cuadros

sobrepuestos que se mostrarían en orden incorrecto para la recuperación visible de los errores.

Para realizar la recuperación de errores, MEncoder tiene alguna similitud con respecto a ffmpeg en cuanto a los comandos y la sintaxis. Se realizaron pruebas para recuperar errores y las cuales fueron satisfactorias, en la imagen 44 se observa la imagen corrupta después de haber eliminado las unidades NAL:



Imagen 44. Resultado con eliminación de unidades NAL

Para recuperar y mejorar las partes donde se ve corrupta la imagen (árboles), se utilizaron dos tipos de comando, idx y el forceidx los cuales trabajan de forma similar, en embargo tienen particularidades en cuanto a su manera de corrección.

```
MacBook-Pro-de-David-2:MenCoder davidpinzon$ ./mencoder Original1.mp4 -idx -oac copy -ovc copy -o recuperado.mp4
```

Imagen 45. Comando de MEncoder utilizando IDX

En el comando anterior se utiliza “idx” el cual reconstruye cada uno de los índices de los archivos en los que no encuentra similitud, esto lo logra debido a que las unidades NAL generan en su cabecera un índice de continuidad para su reconstrucción en la decodificación, por lo tanto, al no obtener esa secuencia busca patrones de similitud donde identifique que se encuentra el archivo corrupto o incorrecto y lo rellena teniendo en cuenta esos patrones que considere pertinente. El resultado final es el que se observa en la imagen 50.



Imagen 46. Resultado con la reconstrucción utilizando IDX

```
MacBook-Pro-de-David-2:MenCoder davidpinzon$ ./mencoder Original1.mp4 -forceidx -oac copy -ovc copy -o recuperado.mp4
```

Imagen 47. Comando de MEncoder utilizando FORCEIDX

La segunda manera de recuperar los errores utilizando MEncoder es con el comando “forceidx” el cual utiliza a la fuerza la reconstrucción de los índices. Es útil para los archivos que tienen mala sincronización debido a la eliminación de las unidades NAL. Tiene la particularidad que activa la posibilidad de búsqueda en

partes del archivo donde normalmente no es posible su manipulación debido a que en esas partes solo tiene interés para los programadores.



Imagen 48. Resultado de la reconstrucción utilizando FORCEIDX

En comparación con los dos métodos de recuperar errores (idx y forceidx) el segundo tiene la ventaja que explora lugares que para muchos es inaccesible debido a la parte de programación que utiliza diferentes parámetros de búsqueda. Por tal motivo, este método es un poco mejor que el primero, a comparación de resultado final como se observa en las imágenes 46 y 48.



Imagen 49. Enfoque en la calidad con pérdidas de unidades NAL

Al realizar un acercamiento a cualquier punto para observar la imágenes con pérdidas de unidades NAL, se puede determinar claramente la manera como se pierde calidad, pareciera como si estuviera distorsionada, eso es una de las ventajas que ofrece HEVC debido a que no aparece una línea negra o elimina el recuadro cuando se pierde información como lo hace AVC y versiones anteriores.

Esta información se debe a la manera como oculta los errores, sin embargo, como ya son aproximadamente el 5% de las unidades eliminadas no las puede ocultar, por tal motivo asimila colores cercanos alargando la información y de esta manera completando la información que no existe o se encuentra corrupta.

4.5. RECOMENDACIONES DE OTRA POSIBLES SOLUCIONES

Las recomendaciones que se dan a continuación para otras posibles soluciones a la pérdida de información, son basadas en pruebas realizadas por otros trabajos y se tienen en cuenta notificación y se da un explicación muy leve, con el objetivo de conocer en sí la solución pero no su funcionamiento como tal. Esto con el objetivo de dar a entender que existen otras opciones de solución, ya sea a un bajo nivel como a un alto nivel de programación.

4.5.1. BIT DE PARIDAD

Uno de los métodos mas comunes de bajo nivel que se utilizan para la detección de errores es el Bit de paridad, su mejor funcionalidad se encuentra cuando la probabilidad de que exista un error es muy baja. De esta manera, es el uso de un bit adicional de paridad por cada uno de los elementos de información transmitidos. Se añade un bit de paridad al bloque de datos, por ejemplo, si existe un numero de bits 1, se le añade un bit 0 de paridad y si son impares, se les añade un bit 1 de paridad

4.5.2. FORWARD CORRECTION

La corrección de errores es un método para obtener un control en la transmisión de datos y de esta manera tener conocimiento de los errores que fueron

transmitidos. De esta manera, el transmisor envía datos redundantes y así el receptor tiene la capacidad de identificar cada uno de los datos que no contiene ningún tipo de error. Es un método muy utilizado en multicast debido a que no necesita una conexión entre la fuente y el destino. En una forma sencilla, cada dato se envía dos veces y el receptor comprueba cada uno de los datos recibidos.

4.5.3. CODIGOS RAPTOR

Los códigos Raptor son la primera clase de código fuente lineal en el tiempo que codifica y decodifica. Su funcionamiento, en una explicación sencilla, consiste en codificar un mensaje que tiene cierto número de símbolos (k), con alguna secuencia que se encuentra delimitada entre símbolos, de tal manera que el conocimiento de cualquier símbolo o más, dentro de la codificación, permite recuperar el mensaje.

Capítulo 5

5. CONCLUSIONES

Todo codificador se puede mejorar en muchos aspectos, dentro de los cuales con determinadas herramientas se puede garantizar una mejora considerable, teniendo en cuenta los factores que afectan la calidad. Por esta razón, se realizaron múltiples pruebas donde se especificaba un patrón estándar para obtener un resultado ideal.

Se paquetizaron los datos y se simuló una transmisión típica como lo es rtp, de forma que se observó el comportamiento del estándar frente a pérdidas y se intentó mejorar el comportamiento con parámetros que mejoraban la calidad en varios aspectos.

En el proyecto se utilizó VLC como reproductor de multimedia debido a la facilidad que brinda en cuanto a la posibilidad de ser usado tanto de server como reproductor de archivos streaming. Pero este reproductor no es la única manera, se puede utilizar diferentes servidores o programas para la simulación de transmisión de los archivos y ver el resultado final.

Existen software como MEncoder, o simplemente programas realizados por desarrolladores, que pueden ser utilizados de una forma diferente. En este caso para la recuperación de las unidades NAL realizan interpolación, buscando patrones en los cuales se encuentre inconsistencias. De esta manera se puede mejorar considerablemente el codificador a parámetros como procesamiento, almacenamiento y calidad.

Se seleccionaron varios patrones de perdidas y se probó con un máximo del 5% de las unidades NAL eliminadas, con el objetivo de introducir técnicas de recuperación frente a errores. Sin embargo, no es la única técnica de recuperación recomendada, ya que existen otras sencillas técnicas como bit de paridad y a un alto nivel de complejidad como los códigos raptor.

6. BIBLIOGRAFIA

Sullivan, G., Ohm, J.-R., Han, W.-J., & Wiegand, T. (2012). Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE* .

ITU. (2013). H.265 : High efficiency video coding.

Vesga, A. (04 de 09 de 2015). Fotograma. *Escuela Iberoamericana de Bibliotecología* .

Martin, A. L. (s.f.). Formatos de audio digital. *Universidad de Valladolid* .

Arango, M. I. (2013). Tipos de formatos . *Universidad Autonoma de Mexico* .

APPLE. (2016). Introducción al formato AAC. *Creative commons* .

Profesorado., I. S. (2008). Conceptos basicos de sonido digital. *Ministerio de Educación, Política Social y Deporte*.

computacion, D. d. (2015). Procesamiento de audio digital. *Universidad de Malaga* .

QUEVEDO, A. F. (2003). FORMATO DE COMPRESION DE IMAGEN. *UNIVERSIDAD CATOLICA "NUESTRA SEÑORA DE LA ASUNCION"* .

tecnología, D. d. (2008). Master en ingeniería de la web. *IES la Cabrera* .

Izquierdo, A. Y. (2011). Conversión de formatos de vídeo .

II, T. d. (2015). VÍDEO DIGITAL Conceptos iniciales Formatos y compresión de vídeo. *Universidad de Buenos Aires* .

Concha, J. R. (2014). Contenedores y formatos de video. *Universidad Icesi* .

Dias, R. (2015). Video Converter Ultimate 4K. *Aiseesoft* .

Formatos de video. (s.f.). Obtenido de coolutils: <https://www.coolutils.com/>

Wang, Y. (2015). Survey of Objective Video Quality Measurements . *EMC Corporation Hopkinton*, .

Thomas Wiegand, G. J. (2003). Overview of the H.264/AVC Video Coding Standard. *IEEE* .

Marshall, D. (2001). Capas de vídeo MPEG.

FFMPEG. (s.f.). Obtenido de <https://ffmpeg.org/>

Project, C. (2015). *The Structure of HEVC Video*. Obtenido de <https://www.codeproject.com/Tips/896030/The-Structure-of-HEVC-Video>

Yuxia Wang, T.-L. L. (2010). PACKET DROPPING FOR H.264 VIDEOS CONSIDERING BOTH CODING AND PACKET-LOSS ARTIFACTS. *chool of Information Engineering, Communication University of China, Beijing, China* .

FFmpeg and H.265 Encoding Guide. (2017). Obtenido de <https://trac.ffmpeg.org/wiki/Encode/H.265>

Sullivan, G., Boyce, J., Chen, Y., Ohm, J.-R., & Segall, C. (2013). Extensiones Estandarizadas de Codificación de Video de Alta Eficiencia. *Revista IEEE* .