

	GESTIÓN SERVICIOS BIBLIOTECARIOS					  	
	CARTA DE AUTORIZACIÓN						
CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 2

Neiva, 18 de Noviembre de 2016

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

Los suscritos:

Ramiro Alberto Dussán Tierradentro, con C.C. No. 1.075.269.014, Cristhian Andrés Rodríguez Polanía, con C.C. No. 1.075.270.674, autores de la tesis y/o trabajo de grado titulado Evaluación de la Calidad de Video 3D en Transmisiones Inalámbricas con el Nuevo Estándar HEVC presentado y aprobado en el año 2016 como requisito para optar al título de Ingeniero Electrónico; autorizamos al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.

- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.

- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores” , los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

	GESTIÓN SERVICIOS BIBLIOTECARIOS						  
	CARTA DE AUTORIZACIÓN						
CÓDIGO	AP-BIB-FO-06	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 2

EL AUTOR/ESTUDIANTE:

Firma: *Ramiro Alberto Dussán T.*

EL AUTOR/ESTUDIANTE:

Firma: *Erwin Andrés Rodríguez Polanco*

	GESTIÓN SERVICIOS BIBLIOTECARIOS						  
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	1 de 3

TÍTULO COMPLETO DEL TRABAJO: Evaluación de la Calidad de Video 3D en Transmisiones Inalámbricas con el Nuevo Estándar HEVC

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Dussán Tierradentro	Ramiro Alberto
Rodríguez Polanía	Cristhian Andrés

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre
Bravo Obando	Martín Diomedes

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre

PARA OPTAR AL TÍTULO DE: Ingeniero Electrónico

FACULTAD: Ingeniería

PROGRAMA O POSGRADO: Ingeniería Electrónica

CIUDAD: Neiva

AÑO DE PRESENTACIÓN: 2016

NÚMERO DE PÁGINAS: 83

TIPO DE ILUSTRACIONES (Marcar con una X):

Diagramas Fotografías Grabaciones en discos Ilustraciones en general Grabados Láminas
 Litografías Mapas Música impresa Planos Retratos Sin ilustraciones Tablas o Cuadros

	GESTIÓN SERVICIOS BIBLIOTECARIOS						  
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	2 de 3

SOFTWARE requerido y/o especializado para la lectura del documento:

MATERIAL ANEXO:

PREMIO O DISTINCIÓN (*En caso de ser LAUREADAS o Meritoria*):

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

<u>Español</u>	<u>Inglés</u>
1. MIMO	MIMO
2. Calidad de video	Video Quality
3. Video 3D	3D Video
4. Trasmisión de Video	Video
5. HEVC	HEVC

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

En este trabajo se realizó la evaluación de la calidad de video 3D con el nuevo estándar de codificación HEVC (High Efficiency Video Coding) en transmisiones inalámbricas sobre canales de frecuencia plana. Para esto, se configuraron los estándares de codificación para video 3D, 3D Extension (3D-HEVC) y Multiview Extension (MV-HEVC) que siguen las recomendaciones de la ITU-T con la ISO/IEC; se simularon esquemas MIMO con codificación de canal para la transmisión inalámbrica; se evaluó la correlación de movimiento de secuencias de textura y profundidad con Motion Vector (MV); y se implementaron métodos de evaluación objetiva y subjetiva de calidad de video 3D para ver el efecto que tiene la pérdida de paquetes en la información transmitida y la percepción visual del usuario, respectivamente.

	GESTIÓN SERVICIOS BIBLIOTECARIOS					   	
	DESCRIPCIÓN DE LA TESIS Y/O TRABAJOS DE GRADO						
CÓDIGO	AP-BIB-FO-07	VERSIÓN	1	VIGENCIA	2014	PÁGINA	3 de 3

ABSTRACT: (Máximo 250 palabras)

In this work evaluating the quality of 3D video was performed with the new coding standard HEVC (High Efficiency Video Coding) in wireless transmissions on at frequency channels. For this, the coding standards for 3D video, 3D Extension (3D-HEVC) and Multiview Extension (MV-HEVC) following the recommendations of the ITU-T with ISO/IEC were configured; MIMO schemes were simulated with channel coding for wireless transmission; the correlation of movement sequences texture and depth with Motion Vector (MV) was evaluated; and methods of objective and subjective evaluation of quality of 3D video were implemented, to view the effect of packet loss on the information transmitted and the visual perception of the user, respectively.

APROBACION DE LA TESIS

Nombre Jurado: CARLOS ALBERTO PEREZ CAMACHO



Firma:

Nombre Jurado: ALBEIRO CORTÉS CABEZAS



ALBEIRO CORTÉS CABEZAS

Firma:

**EVALUACIÓN DE LA CALIDAD DE VÍDEO 3D EN TRANSMISIONES
INALÁMBRICAS CON EL NUEVO ESTÁNDAR HEVC**

**CRISTHIAN ANDRÉS RODRÍGUEZ POLANÍA
RAMIRO ALBERTO DUSSÁN TIERRADENTRO**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2016**

**EVALUACIÓN DE LA CALIDAD DE VÍDEO 3D EN TRANSMISIONES
INALÁMBRICAS CON EL NUEVO ESTÁNDAR HEVC**

**CRISTHIAN ANDRÉS RODRÍGUEZ POLANÍA
RAMIRO ALBERTO DUSSÁN TIERRADENTRO**

Trabajo de grado para optar al título de Ingeniero Electrónico

**Director
Martín Diomedes Bravo Obando
Candidato PhD. en Telecomunicaciones**

**UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2016**

Nota de aceptación:

Firma del presidente del Jurado

Firma del Jurado

Firma del Jurado

Neiva, Noviembre 17 de 2016

“ Si piensas que vales lo que sabes, estás muy equivocado. Tus conocimientos de hoy no tienen mucho valor más allá de un par de años. Lo que vales es lo que puedes llegar a aprender, la facilidad con la que te adaptas a los cambios que esta profesión nos regala tan frecuentemente ”

José María Aguilar - Sevilla (España)

AGRADECIMIENTOS

Este proyecto de grado no habría sido posible sin el apoyo del **Prof.Dr-Ing. Martín Bravo** por haber sido el promotor del proyecto, quien en su gran y total dedicación académica e investigativa nos dió la oportunidad de disfrutar de su conocimiento y brindarnos asesoría académica y profesional en el desarrollo del proyecto.

Me gustaría agradecer también al **Dr. Ruhan Conceição** de la Universidad Federal de Pelotas (UFPel), por su contribución en el proyecto de grado, ya que su buena influencia a nivel de investigación nos introdujo al mundo de la codificación, decodificación y empaquetamiento de video en telecomunicaciones.

Mil gracias a Dios por todas la bendiciones y alegrías de la vida. Por rodearme de personas maravillosas como lo son mis padres **Viryeni Polanía Vera** y **Gregorio Rodríguez González**, mis hermanos **Daniel Stiven Rodríguez** y **José David Rodríguez**, mis tíos, primos, y mis grandiosos abuelos, que sin su ayuda y apoyo durante todos estos años no habría sido posible ser lo que soy ahora.

Finalmente, a **Ramiro Alberto Dussán Tierradentro**, **Daniel Fernado Murcia**, **Joseph Moreno**, **Laura Daniela Méndez** y todo mi grupo de compañeros por influenciarme positivamente, ayudarme a ser una mejor persona y por soportarme. Gracias Dios por permitirme compartir el final de esta maravillosa etapa de mi vida con una persona tan especial como lo es **Estefanía Salazar**.

Cristhian Andrés Rodríguez Polanía

Dios es aquel ser a quien más estoy agradecido. Me dio a mis padres, **Ramiro Dussán Chaux** y **Gloria Amparo Tierradentro Álvarez**, a quienes agradezco por todo lo que han hecho por mi desde que me trajeron al mundo, pasando por cuidarme, educarme e inculcarme los valores y principios que hasta el día de hoy mantengo. Me dio a mis hermanas, **Lina Marcela** y **Gloria Isabel Dussán Tierradentro**, hermosas compañías a quienes agradezco por el apoyo y amor incondicional que me han dado desde que soy uno de sus modelos a seguir en la vida. Me acompañó en los obstáculos de esta hermosa carrera, donde crucé caminos con **Cristhian Andrés Rodríguez Polanía**, mi compañero de tesis, y **Martín Diomedes Bravo Obando**, mi director de tesis, personas que creyeron en mí y me acompañaron en el transcurso de mi vida universitaria, aportando sus conocimientos, virtudes y experiencias, no sólo en este trabajo, sino también en mi vida, y por lo cual estoy muy agradecido. Y como no, me presentó tíos, primos, abuelos, amigos y conocidos que ya pasaron por mi vida, que de una u otra manera aportan su granito de arena en mi camino y me recuerdan lo especial que soy para ellos. A todos y cada uno de los seres que Dios ha puesto en mi camino, sepan que gracias a sus aportes, valores, esfuerzos y sacrificios, estoy logrando terminar este ciclo de mi vida, dando como último paso la culminación de este trabajo de grado.

Ramiro Alberto Dussán Tierradentro

CONTENIDO

	Pág.
INTRODUCCIÓN	17
1. PLANTEAMIENTO DEL PROBLEMA	18
2. JUSTIFICACIÓN	19
3. OBJETIVOS	20
3.1 OBJETIVO GENERAL	20
3.2 OBJETIVOS ESPECÍFICOS	20
4. MARCO TEÓRICO	21
4.1 SENSACIÓN DE PROFUNDIDAD	21
4.1.1 Técnicas de visualización 3D	21
4.1.2 Compresión	23
4.1.3 Estándar de codificación H.265 (HEVC)	24
4.2 TRANSMISIÓN INALÁMBRICA	27
4.2.1 Ganancias en Diversidad y Multiplexación	27
4.2.2 Modelo del canal MIMO	28
4.2.3 Codificación de bloque espacio - temporal	30
4.2.4 Codificación de canal	30
4.2.5 Método de Montecarlo	32
4.3 EVALUACIÓN OBJETIVA	33
4.3.1 Métodos de evaluación objetiva	33
4.3.2 Modelos existentes de estimación de calidad de video	35
4.4 EVALUACIÓN SUBJETIVA	35
4.4.1 MOS (Mean Opinion Score)	35
4.4.2 Métodos subjetivos para medir la calidad percibida de video	35
4.4.3 Agudeza visual - Test Snellen	36
5. SISTEMA DE SIMULACIÓN	38
5.1 Configuración del esquema HEVC	38
5.2 Codificación de Video 3D	38
5.3 Transporte y empaquetamiento de datos	41
5.4 Simulación de la transmisión inalámbrica	43
5.4.1 Verificación de los esquemas de transmisión	43
5.4.2 Implementación del codificador de canal	46
5.5 Pérdida de paquetes	47
5.6 Evaluación objetiva	47
5.7 Evaluación subjetiva	49
5.7.1 Prueba Snellen	49
5.7.2 Condiciones de Observación	50
5.7.3 Realización de la evaluación subjetiva	50

6.	ANÁLISIS Y RESULTADOS	53
6.1	Análisis y resultados de los esquemas de transmisión	53
6.2	Análisis y resultados del estándar HEVC	53
6.2.1	Video 3D de 2 vistas	56
6.2.2	Video 3D de 2 vistas más profundidad y renderización	59
6.2.3	Calidad de codificación del estándar HEVC	59
6.3	Análisis y resultados de la evaluación subjetiva	60
7.	CONCLUSIONES	63
8.	RECOMENDACIONES Y TRABAJO FUTURO	64
9.	BIBLIOGRAFÍA	65
	ANEXOS	68

LISTA DE TABLAS

	Pág.
Tabla 1. Formatos YUV y calidades asignadas	24
Tabla 2. Videos de test de verificación	39
Tabla 3. Videos de prueba en formato .YUV y especificaciones originales	40
Tabla 4. Parámetros de cámara para configuración de codificación	40
Tabla 5. Relación entre los valores de PSNR y MOS.	48
Tabla 6. Esquema de valores para DSIS	52
Tabla 7. Valores PSNR de las secuencias de video 3D	61
Tabla 8. Directorios según manual de HM Software	69

LISTA DE FIGURAS

		Pág.
Figura 1.	Representación de la Técnica de Anáglifo	22
Figura 2.	Representación de la Técnica de gafas pasivas polarizadas	22
Figura 3.	Modelo de color YUV. Imagen junto con sus componentes Y, U y V	24
Figura 4.	Esquema general del canal MIMO	29
Figura 5.	Diagrama de bloques del sistema MIMO	29
Figura 6.	Esquema de modelos de FR, RR y NR	34
Figura 7.	Prueba de Snellen	37
Figura 8.	Vista principal de la secuencia de video “Newspaper”	39
Figura 9.	Casos de pruebas de codificación	40
Figura 10.	Una estructura GOP	41
Figura 11.	Tipos de NALs de H.265	42
Figura 12.	cabecera de unidades NAL H.265	43
Figura 13.	Análisis del empaquetamiento de datos. Identificación de NALs	43
Figura 14.	Verificación de los esquemas MIMO	45
Figura 15.	Implementación de la prueba Snellen	49
Figura 16.	Escenario para la evaluación subjetiva	51
Figura 17.	Estructura de presentación del material de prueba	51
Figura 18.	Resultados esquemas MIMO con Turbo Códec	54
Figura 19.	Estructura IPPP	55
Figura 20.	Primera estructura GOP de 3 vistas	55
Figura 21.	Segunda estructura GOP de 3 vistas	55
Figura 22.	Estructura IPPP del archivo cfg de video 2D	55
Figura 23.	Estructura All-Intra	56
Figura 24.	Gráficas de la relación de SNR contra PSNR Video 3D de 2 vistas	57
Figura 25.	Muestra de datos de la secuencia <i>Undo_Dancer</i>	58
Figura 26.	Gráfica PlotBox 8X8 <i>Undo_Dancer</i>	58
Figura 27.	Relación de SNR contra PSNR Video 3D de 2 vistas con profundidad	59
Figura 28.	Renderización	60
Figura 29.	PSNR <i>Newspaper</i>	60
Figura 30.	Medida MOS ideal	61
Figura 31.	Medida MOS Prueba Subjetiva	62
Figura 32.	Comparación entre valor ideal de MOS y MOS promedio de la evaluación subjetiva	62
Figura 33.	Descarga de los componentes del esquema 3D-Extension del SVN Server	68
Figura 34.	Compilación del esquema 3D-Extension	69

LISTA DE ANEXOS

Anexo A: Configuración de las recomendaciones del estándar HEVC	Pág. 68
Anexo B: Códigos Referenciados de Python	70
Anexo C: Códigos Referenciados de C++	79
Anexo D: Formato calificación prueba subjetiva	83

GLOSARIO

3D-HEVC: Software/Extensión de codificación de video 3D del estándar HEVC.

Alamouti: Propuesta de un esquema de diversidad en transmisión. Este esquema se trabaja usando dos antenas transmisoras para un sistema receptor de una o más antenas. Alamouti es una STBC muy especial, puesto que es la única STBC ortogonal que puede lograr su plena diversidad de ganancia sin necesidad de sacrificar su velocidad de transmisión de datos.

AVC: Advanced Video Coding. Es una norma que define un códec de video de alta compresión, desarrollada conjuntamente por el ITU-T Video Coding Experts Group (VCEG) y el ISO/IEC Moving Picture Experts Group (MPEG). La intención del proyecto fue la de crear un estándar capaz de proporcionar una buena calidad de imagen con tasas binarias notablemente inferiores a los estándares previos, además de no incrementar la complejidad de su diseño. También conocido como H.264 o MPEG-4.

BER: Tasa de error de bits. Se define como el número de bits recibidos de forma incorrecta respecto al total de bits enviados durante un intervalo especificado de tiempo.

Compresión de datos: Es la reducción del volumen de datos tratables para representar una determinada información empleando una menor cantidad de espacio. Al acto de compresión de datos se denomina compresión, y al contrario descompresión.

Contraste: Diferencia de intensidad de iluminación en la gama de blancos y negros o en la de colores de una imagen fotográfica o cinematográfica.

Crominancia: Es la componente de la señal de video que contiene las informaciones del color.

DIBR: Depth Image Based Rendering. Tratamiento de renderización basado en las capas de profundidad de las vistas del video 3D.

DSIS: Escala de degradación con doble estímulo. Método propuesto en la recomendación ITU-R BT.500-11 para la realización de la prueba subjetiva.

Frame: Fotograma. Un fotograma es cada una de las imágenes impresionadas químicamente en la tira de celuloide del cinematógrafo o bien en la película fotográfica. Cuando una secuencia de fotogramas es visualizada de acuerdo a una determinada frecuencia de imágenes por segundo se logra generar la sensación de movimiento en el espectador.

GOP: Group Of Pictures. En codificación de video, un grupo de imágenes, o estructura GOP, especifica el orden en el que las imágenes tipo intra e inter son ordenadas. El GOP es la unidad de acceso aleatorio, que se utiliza para declarar si las primeras imágenes tipo B después de unas de tipo I pueden ser decodificadas correctamente en caso de un acceso aleatorio. Es un grupo de imágenes sucesivas dentro de un stream codificado de video. Cada stream de video codificado consiste en sucesivos GOP, y es desde las imágenes que conforman estos GOP desde donde se generan las imágenes visibles.

HEVC: High Efficiency Video Coding. Es una norma que define un formato de compresión de video, sucesor de H.264/MPEG-4 AVC, desarrollado conjuntamente por la ISO/IEC Moving Picture Experts Group (MPEG) y ITU-T Video Coding Experts Group (VCEG) como ISO/IEC CD 23008-2 High Efficiency Video Coding. También conocido como H.265 o MPEG-H.

IT++: Es una biblioteca de C++ de procesamiento de la señal matemática y clases de comunicación y funciones. Su uso principal es en la simulación de sistemas de comunicación y para la realización de investigaciones en el área de las comunicaciones. El núcleo de la biblioteca consta de vector genérico y clases de matriz, y un conjunto de rutinas que acompañan. Tal núcleo hace IT++ similar a MATLAB, o GNU Octave SciPy.

ITU: Union Internacional de Telecomunicaciones. Es el organismo encargado de regular las telecomunicaciones a nivel internacional entre las distintas administraciones y empresas operadoras.

JCT-VC: Joint Collaborative Team on Video Coding. Grupo de contactos de expertos en codificación de video.

JMVC: Software/Extensión de codificación de video del estándar AVC.

Luminancia: Densidad angular y superficial de flujo luminoso que incide, atraviesa o emerge de una superficie siguiendo una dirección determinada.

MIMO: Multiple-input, Multiple-output. Esquemas de transmisión inalámbricos cuyo modelado se basa en la diversidad espacial y en el arreglo de 2 o más antenas transmisoras y receptoras. En el formato de transmisión inalámbrica tradicional la señal se ve afectada por reflexiones, lo que ocasiona degradación o corrupción de la misma y por lo tanto pérdida de datos. MIMO aprovecha fenómenos físicos como la propagación multicamino para incrementar la tasa de transmisión y reducir la tasa de error. En breves palabras MIMO aumenta la eficiencia espectral de un sistema de comunicación inalámbrica por medio de la utilización del dominio espacial.

MOS: Puntaje de Opinión Media. Mide la calidad de video a través de una evaluación subjetiva que realiza un grupo amplio de personas que opinan acerca de su percepción.

MPEG: Moving Picture Experts Group. Es un grupo de trabajo de expertos que se formó por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC) para establecer estándares para el audio y la transmisión de video.

MSE: Error Medio Cuadrático. Cuantifica el error entre la señal distorsionada y la señal original.

MV-HEVC: Multiview Extension. Software/Extensión de codificación de video del estándar HEVC.

NAL: Network Abstraction Layer. Los datos de video codificado se organizan en unidades NAL, cada una de las cuales es efectivamente un paquete que contiene un número entero de bytes. Los primeros bytes de cada unidad NAL son bytes de cabecera que contiene una indicación del tipo de datos que contiene dicha unidad NAL, y el resto de bytes contienen la carga útil de datos del tipo indicado en la cabecera. La definición de la estructura de la unidad NAL especifica un formato genérico para su uso tanto en sistemas orientados a la transmisión de paquetes como sistemas orientados al streaming de bits. Una serie de unidades NAL generadas por un codificador se conoce como un stream de unidades NAL.

OpenCV: Es una biblioteca informática de código abierto desarrollado originalmente por la visión de Intel. Es gratuito para uso comercial y la investigación bajo licencia BSD. La biblioteca es multiplataforma y funciona en Mac OSX, Windows y Linux. Se centra principalmente hacia procesamiento de imagen en tiempo real.

OSTBC: Orthogonal Space-time block code. El OSTBC está diseñado de tal forma que los vectores que representan cualquier par de columnas tomadas de la codificación de la matriz sea ortogonal. El resultado de esto es simple, lineal, óptima decodificación en el receptor. Su desventaja más grave es que todos, excepto uno de los códigos que satisfacen este criterio deben sacrificar algunas de sus tasas de datos.

Píxel: Unidad básica de una imagen digitalizada en pantalla a base de puntos de color o en escala de grises.

PlotBox: También conocido como diagrama de caja, es un gráfico que está basado en cuartiles y mediante el cual se visualiza la distribución de un conjunto de datos. Está compuesto por un rectángulo, la “caja”, y dos brazos, los “bigotes”. Es un gráfico que suministra información sobre los valores mínimo y máximo, los cuartiles Q1, Q2 o mediana y Q3, y sobre la existencia de valores atípicos y la simetría de la distribución. Primero es necesario encontrar la mediana para luego encontrar los 2 cuartiles restantes.

Prueba Snellen: Prueba diseñada para evaluar la agudeza visual.

PSNR: Relación Señal a Ruido Pico. Define la relación entre la máxima energía posible de una señal y el ruido que afecta a su representación fidedigna. Debido a que muchas señales tienen un gran rango dinámico, el PSNR se expresa generalmente en escala logarítmica, utilizando como unidad el decibelio (dB).

Python: Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

QAM Digital: Quadrature Amplitude Modulation. QAM Es una técnica que transporta dos señales independientes, mediante la modulación de una señal portadora, tanto en amplitud como en fase. para QAM Digital, se tiene como entrada un flujo de datos binarios, el cual es dividido en grupos de tantos bits como se requieran para generar N estados de modulación, de allí que se hable de N-QAM.

Renderización: Es un término usado en jerga informática para referirse al proceso de generar una imagen o video mediante el cálculo de iluminación GI partiendo de un modelo en 3D. Este término técnico es utilizado por los animadores o productores audiovisuales (CG) y en programas de diseño en 3D como por ejemplo Cinema 4D, 3DS Max, Maya, Blender, SolidWorks, Source Filmmaker, etc.

RGB: Rojo, Verde, Azul. Es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios.

SNR: La relación señal/ruido, se define como la proporción existente entre la potencia de la señal que se transmite y la potencia del ruido que la corrompe. Este margen es medido en decibelios.

STBC: Siglas de codificación espacio-temporal por bloques (Space-time block coding en inglés), es una técnica utilizada en las comunicaciones inalámbricas para transmitir múltiples copias de un flujo de datos a través de una serie de antenas, y para aprovechar versiones recibidas para mejorar la fiabilidad de los datos.

SVH: Sistema visual humano.

Tasa de bits: En informática y telecomunicaciones, el término tasa de bits define el número de bits que se transmiten por unidad de tiempo a través de un sistema de transmisión digital o entre dos dispositivos digitales. Así pues, es la velocidad de transferencia de datos.

Tasa de bloque: Relación entre el número de símbolos y los instantes de tiempo en que éstos símbolos son transmitidos.

Turbo Códec: Ejemplo de código de corrección de errores. Su importancia es que permiten una comunicación fiable y su eficiencia energética está muy cerca del límite teórico predicho por Shannon

YUV: Es un espacio de color típicamente usado como parte de un sistema de procesamiento de imagen en color. Una imagen o video en color se codifica en este espacio de color teniendo en cuenta la percepción humana, lo que permite un ancho de banda reducido para los componentes de diferencia de color o crominancia, de esta forma, hace que los errores de transmisión o las imperfecciones de compresión se oculten más eficientemente a la percepción humana que usando una representación RGB directa.

RESUMEN

En este trabajo se realizó la evaluación de la calidad de video 3D con el nuevo estándar de codificación HEVC (High Efficiency Video Coding) en transmisiones inalámbricas sobre canales de frecuencia plana. Para esto, se configuraron los estándares de codificación para video 3D, 3D Extension (3D-HEVC) y Multiview Extension (MV-HEVC) que siguen las recomendaciones de la ITU-T con la ISO/IEC; se simuló esquemas MIMO con codificación de canal para la transmisión inalámbrica; se evaluó la correlación de movimiento de secuencias de textura y profundidad con Motion Vector (MV); y se implementaron métodos de evaluación objetiva y subjetiva de calidad de video 3D para ver el efecto que tiene la pérdida de paquetes en la información transmitida y la percepción visual del usuario, respectivamente.

PALABRAS CLAVE:

HEVC, Video 3D, Calidad de Video, MIMO.

ABSTRACT

In this work evaluating the quality of 3D video was performed with the new coding standard HEVC (High Efficiency Video Coding) in wireless transmissions on flat frequency channels. For this, the coding standards for 3D video, 3D Extension (3D-HEVC) and Multiview Extension (MV-HEVC) following the recommendations of the ITU-T with ISO/IEC were configured; MIMO schemes were simulated with channel coding for wireless transmission; the correlation of movement sequences texture and depth with Motion Vector (MV) was evaluated; and methods of objective and subjective evaluation of quality of 3D video were implemented, to view the effect of packet loss on the information transmitted and the visual perception of the user, respectively.

KEYWORDS:

HEVC, 3D Video, Video Quality, MIMO.

INTRODUCCIÓN

La tecnología de video ha evolucionado desde sus principios y con ello la necesidad de la transmisión multimedia de modo alámbrico e inalámbrico. En las últimas dos décadas hemos observado un gran desarrollo en la calidad de la imagen prestada al usuario, llegando a tener en la actualidad calidad HD y UltraHD, dando mayor exigencia a los sistemas de codificación y de transmisión para disminuir la pérdida de paquetes o distorsión con un menor ancho de banda y comprimir las imágenes de modo más eficiente.

Con el surgimiento y evolución de la tecnología de video, la aparición de las aplicaciones de video en tres dimensiones genera en el usuario mayores expectativas en cuanto a calidad logrando mayor exigencia en el estándar de codificación y transmisión de video. Es por ello que con el más reciente estándar de codificación HEVC (“High Efficiency Video Coding”) y su extensión 3D-ExtensionHEVC, enfocada en la codificación de video MVC y DIBR (“Depth Image Based Rendering”) en constante desarrollo por expertos de todo el mundo, nace éste proyecto de investigación con el fin de evaluar la calidad de la codificación, decodificación y transmisión en sistemas inalámbricos MIMO de éstos tipos de video 3D, orientado a la pérdida de paquetes.

Por estas razones, el objetivo principal del proyecto es evaluar la calidad de video sobre canales inalámbricos de frecuencia plana en la transmisión de aplicaciones 3D usando el nuevo estándar HEVC.

Este proyecto contiene ocho capítulos, en los cuales se describe el proceso para el diseño, implementación, adquisición de datos y verificación de los mismos en el método de evaluación trabajado.

Los primeros tres capítulos tratan los motivos fundamentales que llevaron a elaborar el proyecto; el planteamiento del problema, justificación y los objetivos propuestos para su desarrollo.

En el cuarto capítulo se realiza una descripción de la tecnología 3D existente, la teoría de codificación de video utilizada y el método de transmisión a implementar; también se hace una descripción completa de los métodos de evaluación objetivas y subjetivas trabajados en el proyecto.

El quinto capítulo detalla paso a paso el proceso de implementación de los sistemas de codificación a trabajar, el diseño de los programas de adquisición de datos de la codificación, transmisión y decodificación, y la implementación de las pruebas objetivas y subjetivas con los cuales se realizó la evaluación de los videos 3D.

En el sexto capítulo se detallan resultados y análisis de los sistemas de codificación, transmisión y decodificación, y de las evaluaciones objetivas y subjetivas implementadas.

En los capítulos siete y ocho se especifican las conclusiones, recomendaciones y se exponen algunas ideas sobre trabajos futuros.

1. PLANTEAMIENTO DEL PROBLEMA

La tecnología 3D se fundamenta en la forma de trabajo conjunto que tiene nuestro cerebro y ojos. Cada uno de nuestros ojos ve la misma escena desde un ángulo ligeramente diferente, cada elemento se ve desde una perspectiva distinta. Esta pequeña diferencia en el ángulo de visión le permite al cerebro distinguir la profundidad. Cuando el cerebro genera una sola imagen es producto de la fusión de dos imágenes para crear una perspectiva en 3D. La tecnología de video 3D tiene como objetivo enviar una imagen distinta a cada ojo, por lo tanto es posible percibir y experimentar una profundidad de perspectiva, el mismo efecto que tenemos al observar las cosas en la vida real¹.

En la actualidad, las aplicaciones de video 3D se han popularizado debido a su capacidad para dar a los espectadores una mejor experiencia multimedia en comparación con el tradicional video 2D, sin embargo, durante su transmisión algo de ésta información se pierde debido a errores de malla que podrían conducir a una disminución en la calidad del video 3D en el receptor. Estos efectos de pérdida de información pueden ser peores que en las convencionales aplicaciones de video 2D, por lo que la transmisión de video 3D es un gran desafío².

El surgimiento de las aplicaciones de video en tres dimensiones (3D) basadas en la técnica DIBR (Depth Image Based Rendering) ha traído un gran reto para la transmisión de video, debido a que, a la información transmitida se le adiciona la información del mapa de profundidad de la imagen, por lo que si explotamos de manera adecuada la correlación que existe entre el color y su correspondiente mapa de profundidad podríamos llegar a reducir la redundancia de información transmitida en dicha aplicación de video 3D³.

El DIBR en 3D es mucho más eficiente comparado con el método de almacenamiento y transmisión de las vistas derecha e izquierda de baja tasa de bits, y de otros métodos de transmisión de video 3D que brinda la mejor experiencia multimedia para el usuario, que por su menor uso de infraestructura para su visualización, se ha impuesto en los últimos años como la opción más eficiente y viable para brindar una experiencia multimedia de video 3D al usuario en el futuro cercano.

¹Red Gráfica Latinoamérica, Fundamentos de la tecnología 3D. En: <http://redgrafica.com/Fundamentos-de-la-tecnologia-3D>.

²S.Grewatsch and E.Muller, Sharing of Motion Vectors in 3D Video Coding, in Proceedings of IEEE International Conference on Image Processing (ICIP '04), vol. 5, pp. 3271-3274, Singapore, October 2004.

³Christoph Fehn , "Depth-Image-Based Rendering (DIBR), Compression and transmission for a new Approach on 3D-TV", Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut (HHI) Einsteinufer 37,10587 Berlin, Germany.

2. JUSTIFICACIÓN

Los errores que presentan las técnicas usadas en el pasado para la disminución de redundancia es que ninguna de estas técnicas fue directamente dirigida a la codificación de video 3D basado en DIBR. Recientemente la correlación entre los vectores de movimiento MV con sus tramas de profundidad y color en secuencias de DIBR fue usado para la corrección de errores y la codificación en H.264. El análisis de la correlación de movimiento de secuencias de color y profundidad con vectores de movimiento MV basada en MPEG-2 y H.264/AVC fue desarrollado principalmente enfocada en reducir el ancho de banda, complejidad y tiempo de codificación⁴.

H.265 o MPEG-H Parte2 llamado de forma común como “High Efficiency Video Coding (HEVC)” es actualmente la norma de codificación de video más reciente de “ITU-T Video Coding Experts Groups” e “ISO/IEC Moving Picture Experts Group”. El principal objetivo de la estandarización HEVC es mejorar significativamente el rendimiento de la compresión en relación con las normas existentes en una tasa del 50% de la tasa binaria para la misma calidad de video perceptual⁵.

⁴C.T.E.R. Hewage, S.T. Worrall, S. Dogan, A.M. Kondoz, “Frame concealment algorithm for stereoscopic video using motion vector sharing,” Multimedia and Expo, 2008 IEEE International Conference on, pp.485-488, June 23 2008-April 26, 2008.

⁵Gary j. Sullivan, Fellow, Jens-Rainer OHM, Member, IEEE, Woo-jin Han, Member, IEEE, and Thomas Wiegand, Fellow, “Overview of the high Efficiency Video Coding (HEVC) Standard”.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Evaluar la calidad de video sobre canales inalámbricos de frecuencia plana en la transmisión de aplicaciones 3D usando el nuevo estándar HEVC (“High Efficiency Video Coding”)

3.2 OBJETIVOS ESPECÍFICOS

- Comprender, implementar y evaluar estructuras básicas de aplicación de video 3D basadas en DIBR en H.264 y H.265
- Desarrollar un componente de software que mediante el uso de DIBR en aplicaciones de video 3D evalúe la correlación que existe entre las tramas de profundidad y color reduciendo la redundancia de información.
- Estudiar modelos utilizados en la implementación de comunicaciones inalámbricas donde se consideran canales de frecuencia plana.
- Evaluar el componente de software desarrollado en el proyecto de grado y comparar resultados con experiencias similares obtenidas en H.264, evaluando características como lo son relación PSNR, calidad de secuencias de profundidad y color transmitidas.
- Elaborar un documento mostrando el resultado de la investigación y la elaboración de un paper.

4. MARCO TEÓRICO

4.1 SENSACIÓN DE PROFUNDIDAD

Todo proceso que permite obtener o crear imágenes en 3D (3 dimensiones) se conoce con el nombre de estereoscopia, y fundamentalmente se basa en el principio natural de la visión humana, en donde cada uno de nuestros ojos capta en un mismo instante una imagen ligeramente diferente a la del otro ojo, debido a la distancia que los separa. Ambas imágenes son procesadas por nuestro cerebro permitiéndonos obtener una sensación de profundidad⁶.

La forma más sencilla de crear en el cerebro la percepción de profundidad es proporcionando a los ojos del espectador dos imágenes diferentes, que representan dos perspectivas del mismo objeto, con una pequeña desviación similar a las perspectivas que de forma natural reciben los ojos.

Los dos ojos, al estar separados entre sí captan cada uno una imagen ligeramente distinta (ángulo). Las diferencias se procesan en el cerebro para calcular la distancia a la que se encuentran los objetos, obteniendo una sensación de profundidad o volumen.

Las variaciones verticales son indiferentes para la percepción de sensación de volumen (a no ser que esta diferencia sea demasiado grande, en cuyo caso crearían visión doble). Solo las variaciones horizontales, producidas por la diferente ubicación de los ojos, producen una sensación de profundidad⁷.

4.1.1 Técnicas de visualización 3D

Existen muchas técnicas distintas para lograr hacer llegar cada imagen al ojo que le corresponde:

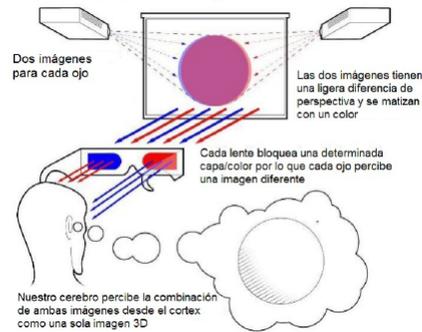
Técnica de Anáglifo

Son imágenes tratadas con filtros de distintos colores (normalmente rojo y cian) sobrepuestas en una sola imagen. Se utilizan unas gafas llamadas gafas anáglifo, que tienen un filtro de diferente color para cada ojo. De esta forma se consigue filtrar las imágenes y conseguir el efecto deseado para que el cerebro pueda interpretar profundidad al captar una imagen diferente en cada ojo. Su representación se muestra en la Figura 1.

⁶Sergio López, Memoria TFC - "Transmisión de vídeo 3D sobre redes P2P". Tesis de grado Ingeniería Técnica en Telecomunicaciones, especialidad en Telemática, (2012).

⁷Sergio López, Memoria TFC - "Transmisión de vídeo 3D sobre redes P2P". Tesis de grado Ingeniería Técnica en Telecomunicaciones, especialidad en Telemática, (2012).

Figura 1. Representación de la Técnica de Anáglifo



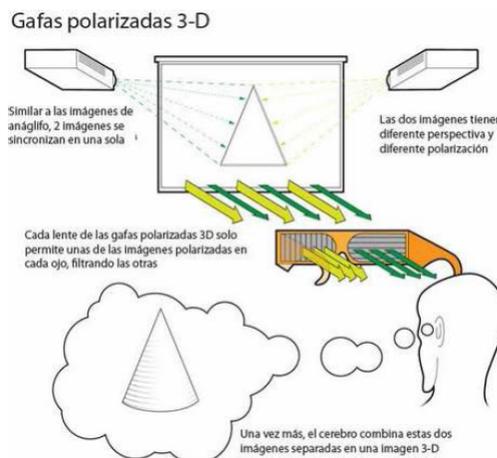
Fuente: Técnicas de visualización de 3D. En línea:
<http://www.nocturnar.com/forum/ciencia/424121-3d-tres-dimensiones.html>
Última sesión: 15-Marzo-2016.

Técnica de gafas pasivas polarizadas

Estas gafas crean la ilusión de imágenes en tres dimensiones mediante la restricción de la luz que llega a cada ojo (estereoscopia basada en la polarización de la luz).

Dos imágenes son proyectadas superpuestas en la misma pantalla a través de diferentes filtros polarizadores. El espectador lleva gafas 3D pasivas polarizadas que también contienen un par de diferentes filtros polarizadores. Como cada filtro pasa sólo la luz que está igual polarizada y bloquea la luz polarizada en la dirección opuesta, cada ojo ve una imagen diferente. Esto se utiliza para producir un efecto tridimensional mediante la proyección de la misma escena en ambos ojos, pero utilizando perspectivas ligeramente diferentes. Su representación se muestra en la Figura 2.

Figura 2. Representación de la Técnica de gafas pasivas polarizadas



Fuente: Técnicas de visualización de 3D. En línea:
<http://www.nocturnar.com/forum/ciencia/424121-3d-tres-dimensiones.html>
Última sesión: 17-Marzo-2016.

Técnica de frames alternados

Gafas activas capaces de cerrar y abrir las lentes de forma sincronizada consiguiendo hacer llegar a cada ojo sólo la imagen que le corresponde. La pantalla de visualización deberá estar sincronizada con las gafas.

Autoestereoscopia

Es el método para reproducir imágenes en 3D que puedan ser visualizadas sin que el usuario tenga que utilizar ningún dispositivo especial (como por ejemplo gafas) ni necesite condiciones especiales de luz.

4.1.2 Compresión

Uno de los grandes problemas para la transmisión o almacenamiento de video digital es el alto ancho de banda necesario. La mayoría de las aplicaciones multimedia requieren volúmenes importantes de almacenamiento y transmisión.

Por esta razón se hace imprescindible el uso de técnicas de compresión de las señales audiovisuales que permitan reducir considerablemente el volumen de información necesario para representarlas⁸.

Modelo de color YUV

El ojo humano es más sensible a la luminancia que al color. YUV es un modelo de color que aprovecha la insensibilidad humana a la crominancia permitiendo tomar menos muestras de crominancia que de luminancia.

Este modelo permite utilizar un ancho de banda más reducido para las componentes de crominancia, de esta forma, consigue que los errores de transmisión o las imperfecciones de compresión se oculten más eficientemente a la percepción humana que usando por ejemplo el modelo de color RGB⁹.

YUV permite la representación de imágenes en tres capas, de manera que juntando las tres capas se obtiene la imagen en color. Usa una representación del tipo Y:Cb:Cr y expresa la proporción entre muestras de luminancia (Y), crominancia azul (Cb) y crominancia roja (Cr).

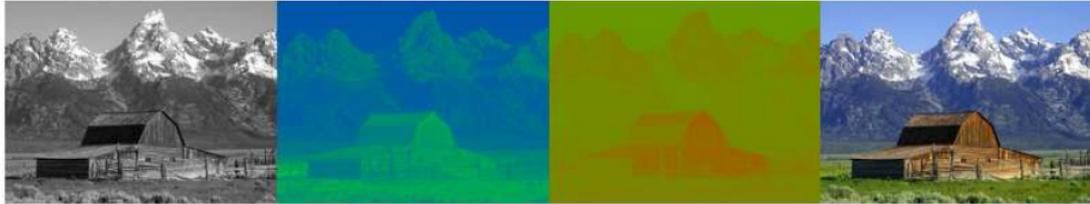
El parámetro Y representa la luminancia (información de brillo), es decir, imagen en escala de grises, mientras que las componentes U (Cb) y V (Cr) representan la crominancia (información de color). En la Figura 3 podemos ver las componentes Y,

⁸Sergio López, Memoria TFC - "Transmisión de vídeo 3D sobre redes P2P". Tesis de grado Ingeniería Técnica en Telecomunicaciones, especialidad en Telemática, (2012).

⁹Sergio López, Memoria TFC - "Transmisión de vídeo 3D sobre redes P2P". Tesis de grado Ingeniería Técnica en Telecomunicaciones, especialidad en Telemática, (2012).

Cb, Cr e imagen resultante.

Figura 3. Modelo de color YUV. Imagen junto con sus componentes Y, U y V



Fuente: Modelo de color YUV. Transmisión de video 3D sobre redes P2P, Sergio López. Pág 58.

Por tanto, es posible reducir la información del color respecto a la de brillo, utilizando varios métodos de submuestreo como por ejemplo 4:4:4, es decir, por cada 4 muestras de luminancia hay 4 muestras de cada una de las crominancias. También puede reducirse la información cromática a la mitad, 4:2:2 (reducir en un factor de 2 en dirección horizontal), con lo que el color tiene la mitad de resolución (en horizontal), y el brillo sigue intacto. Otro método, muy usado, es reducir el color a la cuarta parte, 4:2:0, en el que el color se reduce en un factor de 2 en ambas direcciones, horizontal y vertical. Puede eliminarse por completo la información de color, quedando como 4:0:0 (escala de grises).

De esta manera, en la Tabla 1 podemos distinguir diferentes calidades de imagen en función de esta relación.

Tabla 1. Formatos YUV y calidades asignadas

4:1:1/4:2:0	Calidad Broadcast/Calidad PAL
4:2:2	Calidad de estudio de televisión
4:4:4	Calidad de cine digital y televisión HD

Fuente: Sergio López, "Transmisión de vdeo 3D sobre redes P2P" pág 58.

4.1.3 Estándar de codificación H.265 (HEVC)

"High Efficiency Video Coding" (HEVC) es el más reciente proyecto de video conjunta de las organizaciones de normalización "UIT-T Video Coding Experts Group" (VCEG) y la "ISO/IEC Moving Picture Experts Group" (MPEG), trabajando juntos en una asociación conocida como el equipo de colaboración conjunta sobre la codificación de video (JCT-VC)¹⁰. La primera edición de la norma HEVC se finalizó en enero de 2013, dando como resultado un texto publicado por la UIT-T e ISO/IEC. Se planea seguir trabajando para ampliar la norma para apoyar a varios escenarios de aplicaciones adicionales, incluyendo autonomía extendida utiliza con mayor precisión y soporte de formatos de color, la codificación de video escalable y 3-D/estéreo/multiview codificación de video. En la norma ISO/IEC, la norma HEVC se convertirá en MPEG-H

¹⁰Generic Coding of Moving Pictures and Associated Audio Information? Part2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG 2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994

Parte 2 (ISO/IEC 23008-2) y en la Recomendación UIT-T que es probable que se convierta la Recomendación UIT-T H.265¹¹.

Las normas de codificación de video han evolucionado principalmente mediante el desarrollo de las conocidas normas del UIT-T e ISO/IEC. El UIT-T que produjo H.261¹² y H.263¹³, la norma ISO/IEC produjo MPEG-1¹⁴ y MPEG-4 Visual, y las dos organizaciones produjeron conjuntamente las normas H.262/MPEG-2 Video¹⁵ y H.264/MPEG-4 Advanced Video Coding (AVC). Las dos normas que se produjeron en forma conjunta han tenido un impacto particularmente fuerte y han encontrado su camino en una amplia variedad de productos que son cada vez más frecuentes. A lo largo de esta evolución, se han hecho esfuerzos continuos para maximizar la capacidad de compresión y mejorar otras características como la robustez de pérdida de datos, teniendo en cuenta los recursos computacionales que eran prácticos para su uso en los productos en el momento del esperado despliegue de cada norma.

El estándar de codificación de video principal que precede directamente el proyecto HEVC es el H.264/MPEG-4 AVC, que fue desarrollado inicialmente en el período comprendido entre 1999 y 2003, y luego se extendió en varios aspectos importantes de 2003 a 2009. H.264/MPEG-4 AVC ha sido una tecnología habilitadora para el video digital en casi todas las áreas que no estaba cubierto previamente por H.262/MPEG-2 Video y ha desplazado sustancialmente el estándar más antiguo dentro de sus ámbitos de aplicación existentes. Es ampliamente utilizado para muchas aplicaciones, incluyendo la emisión de alta definición (HD) de señales de televisión por satélite, por cable y sistemas de transmisión terrestres, sistemas de adquisición de contenido de video y edición, cámaras de video, aplicaciones de seguridad, Internet y video a la red móvil, discos Blu-ray y en tiempo real las aplicaciones conversacionales como el chat de video, videoconferencia, y los sistemas de telepresencia¹⁶.

Sin embargo, una creciente diversidad de los servicios, la creciente popularidad del video de alta definición, y la aparición de los formatos más allá-HD (por ejemplo, 4k x 2k o 8k x 4k de resolución) están creando necesidades aún más fuertes para una eficiencia superior a la codificación H.264 / MPEG-4 en capacidad AVC. La necesidad es aún más fuerte cuando una resolución más alta se acompaña de música o captura multivisión y la pantalla. Por otra parte, el tráfico causado por las aplicaciones de video dirigidas a dispositivos móviles y Tablet PC, así como las necesidades de transmisión para servicios de video bajo demanda, están imponiendo severos desafíos en las redes actuales. Un

¹¹Gary j. Sullivan, Fellow, Jens-Rainer OHM, Member, IEEE, Woo-jin Han, Member, IEEE, and Thomas Wiegand, Fellow, "Overview of the high Efficiency Video Coding (HEVC) Standard".

¹²Coding of Audio-Visual Objects? Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual version 1), ISO/IEC JTC 1, Apr. 1999 (and subsequent editions).

¹³Advanced Video Coding for Generic Audio-Visual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003 (and subsequent editions).

¹⁴S.Grewatsch and E.Muller, Sharing of Motion Vectors in 3D Video Coding, in Proceedings of IEEE International Conference on Image Processing (ICIP '04), vol. 5, pp. 3271-3274, Singapore, October 2004.

¹⁵Generic Coding of Moving Pictures and Associated Audio Information? Part2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG 2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994

¹⁶B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, High Efficiency Video Coding (HEVC) Text Specification Draft 9, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct 2012.

aumento del deseo de una mayor calidad y resoluciones también está surgiendo en aplicaciones móviles.

HEVC ha sido diseñado para abordar esencialmente todas las aplicaciones existentes de H.264/MPEG-4 AVC y prestar especial atención a dos cuestiones fundamentales: el aumento de la resolución de video y un mayor uso de las arquitecturas de procesamiento paralelo. La sintaxis de HEVC es genérico y también debe poder ser general para otras aplicaciones que no se mencionan específicamente arriba¹⁷.

Como ha sido el caso para todos los anteriores estándares de codificación de video de la UIT-T e ISO/IEC, en HEVC sólo la estructura de flujo de bits y la sintaxis es estándar, así como restricciones en el flujo de bits y su asignación para la generación de imágenes decodificadas. El mapeo está dado por definir el significado semántico de los elementos de sintaxis y un proceso de decodificación de tal manera que cada decodificador conforme a la norma producirá el mismo resultado cuando se administra un flujo de bits que se ajusta a las limitaciones de la norma. Esta limitación del alcance de la norma permite la máxima libertad para optimizar las implementaciones de una manera apropiada a las aplicaciones específicas (equilibrio entre la calidad de compresión, de costos de implementación, el tiempo de comercialización, y otras consideraciones). Sin embargo, no proporciona garantías de extremo a extremo de la calidad de reproducción¹⁸.

Para ayudar a la comunidad de la industria en el aprendizaje de cómo utilizar la norma, el esfuerzo de estandarización no sólo incluye el desarrollo de un documento de especificación de texto, sino también hacer referencia al código fuente del software como un ejemplo de cómo el video HEVC¹⁹ puede ser codificado y decodificado. El software de proyecto de referencia se ha utilizado como una herramienta de investigación para el trabajo interno de la comisión durante el diseño de la norma, y también puede ser utilizado como una herramienta general de investigación y como la base de productos. Un conjunto de datos de prueba estándar también se está desarrollando para el ensayo de conformidad con el estándar.

En el estándar HEVC podemos encontrar diferentes tipos de imágenes como :

Imágenes I (Intracadro): Las imágenes I son imágenes que utilizan sólo compresión intracadro. Cada cuadro es comprimido y procesado de forma independiente de los demás, y contiene por sí solo toda la información necesaria para su reconstrucción. Las imágenes I suelen ser codificadas de manera similar a la codificación JPEG y son las que más información contienen, y por tanto las que menos compresión aportan. Siempre inician una secuencia y sirven de referencia a las imágenes P y B siguientes.

En secuencias largas, es necesario disponer de imágenes I intercaladas, ya que facilitan el

¹⁷B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, High Efficiency Video Coding (HEVC) Text Specification Draft 9, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct 2012.

¹⁸Advanced Video Coding for Generic Audio-Visual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003 (and subsequent editions).

¹⁹Gary j. Sullivan, Fellow, Jens-Rainer OHM, Member, IEEE, Woo-jin Han, Member, IEEE, and Thomas wiegand, Fellow, "Overview of the high Efficiency Video Coding (HEVC) Standard".

acceso aleatorio a un fotograma dentro de la secuencia. Si todas las imágenes dependen de la anterior podemos llegar a la situación en que para recuperar una imagen, hemos de leer todas las anteriores desde el principio, y eso, en ocasiones, no es productivo.

Imágenes P (Predicción): Las imágenes P se generan a partir de la imagen I o P anterior más próxima. El codificador compara la imagen actual con la anterior I o P, y codifica únicamente los vectores de movimiento y el error de predicción. Se utiliza en este caso una predicción hacia adelante. Estas aportan un grado importante de compresión. Se necesita la decodificación de la imagen de referencia antes de poder ser descodificadas.

Imágenes B (Bidireccionales): Las imágenes B se generan a partir de imágenes previa y futura de los tipos I o P. Mientras que las imágenes I y P pueden propagar errores, ya que de ellas dependen otras, las imágenes B no lo hacen, ya que no intervienen en otras predicciones posteriores. Es necesario que el decodificador tenga las dos referencias, anterior y posterior, para poder descodificarlas correctamente²⁰.

4.2 TRANSMISIÓN INALÁMBRICA

La tecnología MIMO consiste en el empleo de múltiples antenas tanto en el extremo transmisor como en el receptor para comunicaciones inalámbricas.

Los enlaces inalámbricos se ven altamente afectados por efectos multitrayecto. Con los sistemas MIMO conseguimos afrontar de lleno los desvanecimientos que afectan al canal inalámbrico, así como la limitación de potencia transmitida que debe existir en las estaciones base, gracias a que además de introducir mejoras en cuanto a la dimensión frecuencial y temporal, MIMO aumenta la eficiencia espectral de un sistema de comunicación inalámbrica por medio de la utilización del dominio espacial²¹.

Los mecanismos de transmisión con múltiples antenas que ayudan a alcanzar una alta eficiencia espectral se basan en los principios fundamentales de diversidad espacial y multiplexación espacial.

4.2.1 Ganancias en Diversidad y Multiplexación

Ganancia de Diversidad Espacial

Utiliza la diversidad espacial que proveen las múltiples antenas para mejorar la robustez de la comunicación inalámbrica frente al desvanecimiento multitrayecto. Cada enlace producido por una antena transmisora y otra receptora se ve afectado por

²⁰Sergio López, “Transmisión de vídeo 3D sobre redes P2P”, Trabajo Final de Carrera en Ingeniería Técnica de Telecomunicaciones, Universitat Politècnica de Catalunya, Julio 2012.

²¹Alexis Alfredo Dowhuszko, “Teoría de la Transmisión de Información sobre Canales MIMO Inalámbricos con Múltiples Usuarios”, Tesis Doctoral en Ciencias de la Ingeniería, Universidad Nacional de Córdoba, Argentina, Marzo 2010.

un desvanecimiento provocado por obstáculos en el entorno y múltiples reflexiones. Cuantos más enlaces existan entre transmisor y receptor mayor será la probabilidad de que un enlace se vea afectado en menor medida por los desvanecimientos, con lo cual su objetivo final es reducir la probabilidad de corte de servicio en el sistema consiguiendo una mejora en la relación señal a ruido (SNR). A este concepto se le denomina ganancia de diversidad y viene expresada por el producto, entre el número de antenas receptoras y el número de antenas transmisoras²².

Ganancia de Multiplexación Espacial

Consiste en transmitir varios flujos de información por medio de las múltiples antenas (destinados a un mismo usuario) a través de trayectos independientes, los cuales se obtienen a partir de un procesamiento de señales conjunto tanto en transmisión como en recepción. El empleo de múltiples antenas en ambos extremos del enlace posibilita que la eficiencia espectral del sistema crezca de forma lineal con el mínimo número de antenas en transmisión y recepción (siempre y cuando el entorno inalámbrico sea lo suficientemente dispersivo).

4.2.2 Modelo del canal MIMO

En un sistema MIMO, donde N_{tx} es el número de antenas transmisoras y N_{rx} el número de antenas receptoras, se genera un canal de $N_{tx} \times N_{rx}$ subcanales entre el array transmisor y el array receptor. La respuesta al impulso de un canal MIMO se expresa en forma matricial como:

$$H(t) = \begin{bmatrix} h_{11}(t) & \cdots & h_{t1}(t) \\ \vdots & \ddots & \vdots \\ h_{1r}(t) & \cdots & h_{tr}(t) \end{bmatrix} \quad (1)$$

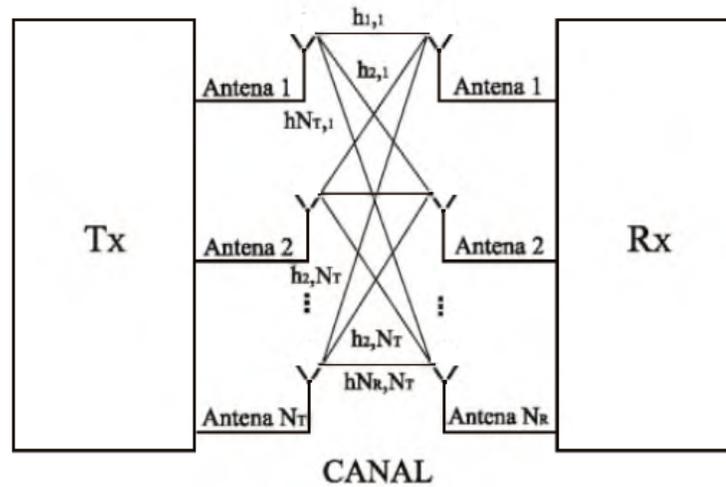
Donde cada uno de los elementos $h_{ij}(t)$ representa el canal generado entre la antena transmisora i , con $i = 1 \dots N_{tx}$, y la antena receptora j , con $j = 1 \dots N_{rx}$. El esquema general del canal MIMO descrito por la ecuación 1 queda representado por la Figura 4.

De esta forma, el diagrama de bloques correspondiente a un sistema MIMO queda definido en la Figura 5.

El diagrama de bloques representado en la Figura 5 se representa matricialmente a través de la ecuación:

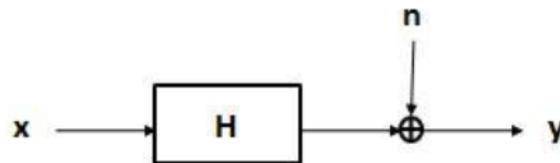
²²Javier Muñoz Asensio, “Comparativa de técnicas de precodificación ZF y MMSE en un sistema celular coordinado”, Proyecto Fin de Carrera en Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Enero 2011. Pag 34.

Figura 4. Esquema general del canal MIMO



Fuente: Bravo. M, Trujillo. D, Barrera. A; Simulación y evaluación de arreglo de antenas MIMO para sistemas de comunicaciones inalámbricas mediante el uso del paradigma de tensores; Revista Ingeniería y Región No. 12 Año 2014; pág 12

Figura 5. Diagrama de bloques del sistema MIMO



Fuente: Muñoz. J; Comparativa de técnicas de precodificación ZF y MMSE en un sistema celular coordinado; Tesis de grado Universidad Carlos III de Madrid 2011; pág 35.

$$\begin{bmatrix} y_1 \\ \vdots \\ y_r \end{bmatrix} = \begin{bmatrix} h_{11}(t) & \cdots & h_{t1}(t) \\ \vdots & \ddots & \vdots \\ h_{1r}(t) & \cdots & h_{tr}(t) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_r \end{bmatrix} + \begin{bmatrix} n_1 \\ \vdots \\ n_r \end{bmatrix} \quad (2)$$

Donde los valores de la matriz y corresponden a las señales obtenidas de la transmisión inalámbrica, siendo éste proceso la adición entre el producto de la respuesta al impulso H y la señal de entrada x , con el ruido e interferencias n .

4.2.3 Codificación de bloque espacio - temporal

Una de las metodologías para la explotación de la capacidad en sistemas MIMO consiste en usar la diversidad espacial para contrarrestar el desvanecimiento de canal²³. Esto se puede lograr mediante la transmisión de varias réplicas de la misma información a través de cada antena; al hacerlo, la probabilidad de perder la información disminuye exponencialmente. Un sistema MIMO con antenas de transmisión, N_{tx} , antenas de recepción, N_{rx} , tiene potencialmente diversidad completa (i.e., máxima diversidad) ganancia igual a $N_{tx} \times N_{rx}$. Las diferentes réplicas enviadas para explotar la diversidad son generadas por un codificador de espacio-tiempo, que codifica una única transmisión a través del espacio utilizando todas las antenas de transmisión y a través del tiempo mediante el envío de cada símbolo en diferentes momentos. Esta forma de codificación se denomina Codificación en Espacio-Tiempo [Space-Time Code, STC]. Debido a su sencillez de decodificación, la forma más dominante de STC son Códigos de Bloque Espacio-Temporal (Space-Time Block Codes).

Esquema de Alamouti

Alamouti propuso un esquema de diversidad en transmisión para complementar los estudios que ya existían sobre diversidad espacial, pero que sobre todo se centraban en la recepción, por ejemplo aplicadas a las estaciones base de sistemas de telefonía móvil para mejorar la calidad en la recepción. El esquema Alamouti es parte de una clase general de STBC conocida como Códigos de Bloque Espacio-Temporal Ortogonales [OSTBC]²⁴.

4.2.4 Codificación de canal

El codificador de canal introduce redundancia controlada de tal forma que si se produce algún error en el canal de comunicaciones se pueda detectar y corregir. Hay distintos sistemas de protección contra errores: códigos bloque, códigos convolucionales, códigos de rejilla (“trellis”), etc.

Turbo Códec

Los turbo códigos son una clase de códigos de corrección de errores, que se introdujeron, junto con un algoritmo de decodificación. La importancia de los turbo códigos es que permiten una comunicación fiable y su eficiencia energética está muy cerca del límite teórico predicho por Shannon²⁵. Desde su introducción, los turbo códigos se han utilizado en aplicaciones de baja potencia, como las comunicaciones por satélite, así

²³Cortés-Peña, L.M. (2009). MIMO Space-Time Block Coding (STBC): Simulations and Results [Design Project: Personal and Mobile Communications, Georgia Tech (ECE6604)]. En línea: http://users.ece.gatech.edu/cortes/STBC_Matlab.html

²⁴Capítulo 3: Esquema de Alamouti. En Línea: <http://bibing.us.es> Última conexión: 30-Sept-2016

²⁵Shannon, C. E. (1948), A Mathematical Theory of Communication. Bell System Technical Journal, 27: 379?423. doi: 10.1002/j.1538-7305.1948.tb01338.x

como para aplicaciones de interferencia limitada.

Codificador

Los turbo códigos son una familia de códigos concatenados o unidos en paralelo. La creación de los mismos parte de un conjunto concreto de códigos básicos denominados códigos constituyentes, formándose una palabra código de salida a partir de los bits de redundancia de cada uno de los códigos constituyentes. Un ejemplo de lo que significa la concatenación en paralelo puede ser el de imaginar que se está transmitiendo la misma información en dos lenguas distintas simultáneamente. La transmisión de un mismo mensaje con distintas codificaciones de forma simultánea puede facilitar el proceso de detección y corrección de errores en el receptor de manera decisiva²⁶.

Los códigos constituyentes de los turbo códigos son códigos convolucionales que proporcionan mejores prestaciones, llamados “Códigos convolucionales Recursivos Sistemáticos (RSC)”. Los RSC son códigos convolucionales realimentados, es decir, la entrada al registro de desplazamiento del codificador no solo depende del bit de entrada, sino que, también es función del contenido de dicho registro.

Los códigos constituyentes en los turbo códigos se concatenan en paralelo por medio de *entrelazadores*. Esto significa que los bits se toman en grupos de una cierta longitud, proporcionando a cada codificador constituyente una permutación distinta de cada grupo. Barajando la información a la entrada de cada codificador, se pretende crear dependencia entre bits no consecutivos, dando a cada bit capacidad de corrección sobre otros bits alejados del mismo.

Decodificador

La decodificación de los turbo códigos no responde a un único algoritmo de decodificación. En el proceso de decodificación de códigos concatenados, dos o más decodificadores constituyentes suelen pretender tomar una decisión sobre una misma información recibida. La decisión final debe tomarse a partir de las decisiones de los decodificadores constituyentes y alguna información adicional que permita discriminar entre ellas. En este sentido, algunos de los algoritmos propuestos se basan en modificaciones sobre el algoritmo de Viterbi²⁷, con la intención de obtener a la salida del decodificador algún tipo de información de fiabilidad además del símbolo decodificado.

Esta idea motivó la aparición del SOYA (Soft Output Viterbi Algorithm) a finales de los ochenta²⁸. Este tipo de algoritmos suponen una clase subóptima de decodificación de los turbo códigos, dada su incapacidad de obtener una distribución de probabilidades a posteriori (APP) para cada símbolo recibido, aunque la degradación que introducen no es muy significativa si las operaciones se realizan bit a bit.

Otra posibilidad es la de los algoritmos de decodificación que hacen uso de la máxima

²⁶Los “Turbo Códigos”, nuevos subsistemas de codificación de canal para mejorar la transmisión digital de gran capacidad, Enrie Armengol Roig, J.A. Delgado-Penín, BURAN No.8 DICIEMBRE 1996 Pag 66, 67, 68, 69

²⁷En línea: http://materias.fi.uba.ar/6624/index_files/outline_archivos/viterbi

²⁸J. HAGENAUER, P. HOEHER, A Viterbi Algorithm with Soft-Decision Outputs and its Applications, Proceedings of GLOBECOM'89, Dallas, Texas, noviembre 1989, pp. 47.1.1-47.1.7

probabilidad a posteriori (MAP). Estos algoritmos, conocidos desde principios de los setenta y menos populares que el algoritmo de Viterbi se han visto reavivados recientemente con la aparición de los turbo códigos. En ellos, el decodificador proporciona a la salida la secuencia de distribuciones APP de los símbolos de información, es decir, sin forzar decisión alguna facilita la información más completa posible sobre el conjunto de palabras código para cada palabra recibida, después de haber recibido toda la secuencia. La decodificación por medio de algoritmos basados en la MAP es óptima si se conoce la probabilidad a priori asociada a cada palabra código.

LDPC (Low Density Parity Check)

Comprobación de Paridad de Baja Densidad. Son una clase de códigos de corrección de error lineal que permiten transmitir un mensaje por un canal de comunicaciones ruidoso (canal de transmisión con errores).

Los códigos LDPC²⁹ son códigos lineales cuya propiedad esencial es la de tener por lo menos una matriz de paridad de baja densidad, es decir con pocos elementos distintos de cero. Formalmente, decimos que una secuencia (n) de códigos es LDPC si cada código tiene por lo menos una matriz de paridad en la cual la cantidad de elementos distintos de cero es $O(n)$.

El gran interés por los sistemas de codificación basados en códigos LDPC se debe a que permiten comunicar con eficiencia muy cercana al límite establecido por Shannon, con confiabilidad arbitrariamente grande y con muy baja complejidad para una gran variedad de medios de comunicación.

Los códigos LDPC se están implementando en aplicaciones donde la transferencia de información a través del ancho de banda o de canal de retorno está limitado por la presencia de ruido. Aunque la aplicación de turbo códigos se ha impuesto en Estados Unidos como código para los transpondedores de satélite, LDPC tiene una ventaja, la ausencia de patente.

4.2.5 Método de Montecarlo

El método Montecarlo³⁰ es un método numérico que permite resolver problemas físicos y matemáticos mediante la simulación de variables aleatorias. La importancia actual del método Montecarlo se basa en la existencia de problemas que tienen difícil solución por métodos exclusivamente analíticos o numéricos, pero que dependen de factores aleatorios o se pueden asociar a un modelo probabilístico artificial (resolución de integrales de muchas variables, minimización de funciones, etc.)³¹.

El método de Montecarlo proporciona soluciones aproximadas a una gran variedad de

²⁹En línea: <http://circuit.ucsd.edu/yhk/ece154c-spr16/pdfs/ErrorCorrectionIII.pdf>

³⁰Peña Sánchez de Rivera, Daniel (2001). "Deducción de distribuciones: el método de Monte Carlo", en Fundamentos de Estadística. Madrid: Alianza Editorial.

³¹Método de Montecarlo. En https://www.uam.es/personal_pdi/ciencias/carlosp/html/pid/montecarlo.html. Última sesión: 28-sept-16

problemas matemáticos posibilitando la realización de experimentos con muestreos de números pseudoaleatorios en una computadora. A diferencia de los métodos numéricos que se basan en evaluaciones en N puntos en un espacio M -dimensional para producir una solución aproximada, el método de Montecarlo tiene un error absoluto de la estimación que decrece a razón de $\frac{1}{\sqrt{N}}$.

4.3 EVALUACIÓN OBJETIVA

Los métodos objetivos se basan en analizar el contenido y las características del material a evaluar mediante algoritmos y fórmulas matemáticas, para lograr como resultado una estimación de la calidad equivalente a la que evalúa el observador, siendo esta el mayor desafío.

En general, los métodos objetivos para medir la calidad de video pueden categorizarse según el análisis de los parámetros de codificación y transmisión o el análisis del video.

4.3.1 Métodos de evaluación objetiva

Hasta la fecha no hay métodos objetivos que evalúen la calidad del video que aplique para cualquier caso, en donde se incluyan diferentes codificadores, formatos de pantalla, tasas de bits, imágenes por segundo, ambientes de transmisión, etc. No obstante, existen diversas propuestas de métodos o modelos de medida, adaptados únicamente a ciertas condiciones³².

Varios de estos modelos han sido estandarizados por la ITU (Unión Internacional de Telecomunicaciones). Además, muchos otros modelos han sido expuestos en distintas publicaciones y congresos³³.

Modelos basados en el análisis de contenido

Estos modelos basan su funcionamiento especialmente, en el análisis detallado del contenido del video. Los modelos a su vez se clasifican de acuerdo a la disponibilidad total, parcial o nula de la señal del video original.

Métodos con disponibilidad total de la señal original (FR - Referencia completa)

Básicamente, la señal de video original se contrasta con la señal de video degradado cuadro por cuadro. Estos métodos pueden ser utilizados para clasificar en forma objetiva un codificador, un sistema de transmisión, el efecto que puede producir un ancho de

³²Suarez. J, Valencia. O; Diseño de un modelo objetivo para evaluar la calidad de video 3D anaglifo; Tesis de grado Ingeniería Electrónica, 2016.

³³Kai Zeng, Abdul Rehman, Jiheng Wang, and Zhou Wang. "From H. 264 to HEVC: Coding gain predicted by objective video quality assessment models". En: International Workshop on Video Processing and Quality Metrics for Consumer Electronics],(Feb 2013). 2013

banda pequeño o para medir factores que degraden las señales mediante ambientes controlados. Sin embargo, su principal desventaja es la imposibilidad de trabajar en aplicaciones de tiempo real ya que no es posible contar con las señales originales.

Métodos con disponibilidad parcial de la señal original (RR - Referencia reducida)

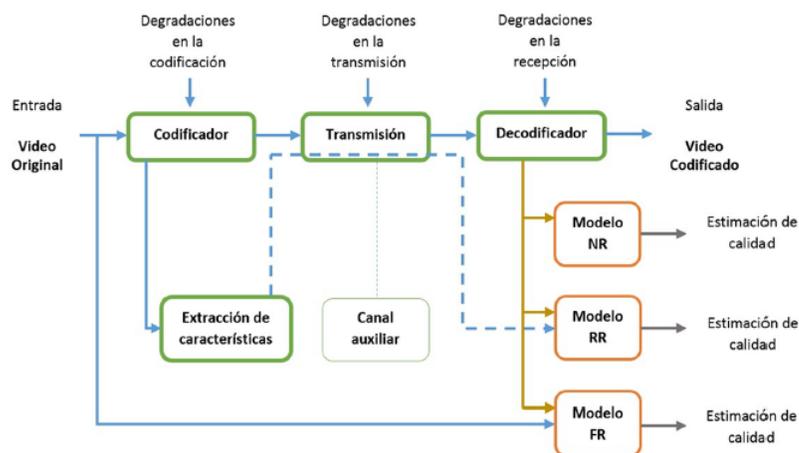
A través del canal de transmisión se envían junto con el video codificado ciertos parámetros que caracterizan a la señal de video original. Se selecciona un ancho de banda específico para el envío de este tipo de información adicional. Así el tamaño del archivo sería un poco más grande, a cambio de tener una referencia en el receptor para poder medir la calidad percibida.

Métodos sin disponibilidad de la señal original - NR (Sin referencia)

Estos métodos estiman la calidad de video percibida basándose exclusivamente en el análisis de la señal que llega al receptor, luego de la codificación y la transmisión. Son los más complejos al momento de implementar, pero su ventaja radica en que no se requiere de otra información diferente a la del propio video recibido y degradado. De cierta manera es análogo a la evaluación subjetiva que realizan las personas, ya que al momento de calificar un servicio de video el cliente no cuenta con la señal original, ni con información adicional para juzgar la calidad de una señal de video, solo se basa en experiencias anteriores y en las expectativas que tenga respecto al sistema.

El sistema de cualquiera de los tres métodos descritos anteriormente se referencia en la Figura 6. Se puede observar que la estimación del canal para cualquiera de los modelos requiere los resultados del decodificador, donde se comparan con el video original en el caso del modelo FR, para el caso del modelo RR se comparan con ciertas características del video original transmitidas por un ancho de banda diferente, y no hay comparación alguna para el modelo NR.

Figura 6. Esquema de modelos de FR, RR y NR



Fuente: José Joskowicz and Rafael Sotelo Bovino. "Modelo de estimación de calidad de video: video quality experts groups". En: Memoria de Trabajos de Difusión Científica y Técnica 10 (2012), pp. 7-17.

4.3.2 Modelos existentes de estimación de calidad de video

Muchos modelos han sido desarrollados para diferentes aplicaciones específicas (SDTV, HDTV, Multimedia) y publicados en organismos internacionales dedicados exclusivamente a la estandarización de métricas de calidad de video.

Estas medidas objetivas de la calidad del video requieren de la secuencia original para poder ser calculadas y están basadas en obtener las diferencias, píxel a píxel, entre el video original que se encuentran previo a la compresión y el video degradado que se obtienen luego del proceso de reconstrucción. Las medidas más simples de este tipo son el MSE (Error medio cuadrático), RMSE (Raíz cuadrada del error medio cuadrático) y el PSNR (Relación señal a ruido pico). Es necesario determinar la relación de aspecto, el rango dinámico que pueden tomar los valores por cada píxel y es fundamental que los videos original y degradado estén alineadas espacial y temporalmente³⁴.

4.4 EVALUACIÓN SUBJETIVA

4.4.1 MOS (Mean Opinion Score)

La forma más confiable para medir la calidad de un video es a través de la evaluación subjetiva realizada por un conjunto amplio de personas que opinan acerca de su percepción. Con estas opiniones se puede calcular la métrica MOS (Puntaje de Opinión Media)³⁵, la cual es aceptada como medida de calidad.

Esta evaluación subjetiva debe ser ejecutada en ambientes controlados, de tal forma que todos los asistentes a la evaluación observen los videos bajo las mismas condiciones; lo que conlleva a la necesidad de estandarizar la forma en que se realizan estas evaluaciones a través de recomendaciones que permitan luego comparar los resultados de una misma evaluación entre sí, y a su vez, comparar evaluaciones realizadas en diferentes lugares y por diferentes personas.

4.4.2 Métodos subjetivos para medir la calidad percibida de video

Los métodos subjetivos son los más confiables, sin embargo son difíciles de implementar. Se requiere bastante tiempo para preparar las condiciones bajo las que se realiza la prueba, involucra a un conjunto de personas “no expertas” que dispongan de la voluntad y el tiempo para realizar las evaluaciones; además son necesarios períodos de procesamiento antes y después de realizar la prueba.

Diversos métodos subjetivos de evaluación de la calidad de video son reconocidos,

³⁴Suarez. J, Valencia. O; Diseño de un modelo objetivo para evaluar la calidad de video 3D anaglifo; Tesis de grado Ingeniería Electrónica, 2016.

³⁵ITU Union. “ITU-T Recommendation P.800.1: Mean Opinion Score (MOS) terminology”. En: International Telecommunication Union, Tech. Rep (2006).

y están estandarizados en las recomendaciones ITU-R BT.500-11³⁶, especialmente desarrollada para aplicaciones de televisión donde se detallan los métodos DSIS, DSCQS, SSCQE y SDSCE, e ITU-T P.910³⁷ para aplicaciones multimedia la cual propone los métodos ACR, DCR y PR. A continuación se describe el método implementado en el proyecto:

DSIS (Escala de degradación con doble estímulo)

Este método consiste en comparar dos secuencias de videos, una dada por la señal degradada y la otra secuencia por la señal original. Proporciona una evaluación precisa de la transparencia o fidelidad del sistema que procesa las secuencias, puesto que el observador compara directamente la secuencia original con la degradada. Además evalúa métricas de tipo FR (Referencia Completa), en las que se dispone de la señal original para realizar la predicción; las señales pueden ser videos o imágenes. En el desarrollo de este método se deben tener en cuenta las condiciones de visualización, iluminación del ambiente, ubicación de las personas con respecto al monitor o televisor, entre otras; las cuales están controladas y detalladas en la recomendación³⁸.

4.4.3 Agudeza visual - Test Snellen

La agudeza visual es la principal característica evaluada para determinar que tan bien ve una persona. En la actualidad, una de las pruebas más utilizadas para medir la agudeza es el test de Snellen³⁹ que se ilustra en la figura 7.

Con esta prueba es posible estudiar la capacidad que tiene el ojo humano de diferenciar objetos a distancias considerables. La agudeza visual alcanza varias decenas de metros en las personas con una vista sana, pero esta capacidad se puede ver disminuida por alteraciones oculares como la miopía (visión borrosa para los objetos lejanos) o la hipermetropía (visión borrosa para los objetos cercanos) en las que el ojo tiene dañado su mecanismo de refracción y enfoca de una manera errónea.

³⁶ITU-R BT Recommendation. "500-11: Methodology for the Subjective Assessment of the Quality of Television Pictures. 500-11". En: Standardization Sector of International Telecommunication Union, ITU Telecom (2002).

³⁷ITUT Recommendation. "P.910: Subjective video quality assessment methods for multimedia applications". En: Standardization Sector of International Telecommunication Union, Geneva (2008).

³⁸Suarez. J, Valencia. O; Diseño de un modelo objetivo para evaluar la calidad de video 3D anaglifo; Tesis de grado Ingeniería Electrónica, 2016.

³⁹Alejandro León Álvarez and Jorge Mario Estrada. "Prueba piloto para determinar la reproducibilidad de la carta de Snellen al evaluar la agudeza visual en sujetos jóvenes". En: Ciencia Tecnología para la Salud Visual y Ocular 7.1 (2009), pp. 13-24.

Figura 7. Prueba de Snellen

E	1	20/200
F P	2	20/100
T O Z	3	20/70
L P E D	4	20/50
P E C F D	5	20/40
E D F C Z P	6	20/30
F E L O P Z D	7	20/25
D E F P O T E C	8	20/20
L E F O D P C T	9	
F D P L T C E O	10	
P E Z O L C F T D	11	

Fuente: fondation pro visu; documentos fidedignos de Internet en enfermedades visuales y en la prevención de la ceguera; En línea: https://www.provisu.ch/Age/Snellenchart_e.s.pdf

5. SISTEMA DE SIMULACIÓN

Este capítulo describe la forma en cómo se implementaron los diferentes tipos de esquemas para la codificación, transmisión y decodificación de los videos 3D. Así mismo, se habla del diseño e implementación de códigos para la identificación, el análisis y la evaluación de parámetros objetivos y subjetivos requeridos para el proyecto.

5.1 CONFIGURACIÓN DEL ESQUEMA HEVC

En este primer apartado, se realiza la configuración del esquema HEVC que codifica video 3D aprovechando la correlación que existe entre las diferentes vistas. Para la labor investigativa se estudió la extensión JMVC de las recomendaciones de H.264 para configurar 3D-Extension de H.265. Estas recomendaciones permiten una eficiente codificación de secuencias capturadas simultáneamente desde varias cámaras de video, para poder predecir imágenes que no solo se basan en la relación temporal entre frames de la misma secuencia sino que se basa en los frames de secuencias de imágenes de cámaras adyacentes.

Para la configuración del esquema HEVC, hubo colaboración de la lista de correos del “Joint Collaborative Team on Video Coding (JCT-VC)”, que reúne y contacta expertos de codificación de video en todo el mundo. Los componentes que siguen éstas recomendaciones pueden ser descargadas de SVN Server [21][22]. El proceso de descarga de los componentes, configuración y compilación de las recomendaciones usando el terminal del sistema operativo GNU/Linux con distribución Ubuntu se puede detallar en la Anexo A.

5.2 CODIFICACIÓN DE VIDEO 3D

Para realizar los experimentos y/o pruebas de codificación se usaron videos desarrollados en laboratorios bajo parámetros controlados que son aptos para codificar tanto en Multiview Extension (MV-HEVC) como en 3D-Extension (3D-HEVC) con su mapa de profundidad y su posterior sintetización de vistas (renderización) para displays o pantallas 3D.

Los videos de prueba de secuencias de múltiples vistas junto con sus asociados datos de profundidad y correspondientes vistas de entrada para los experimentos se especifican en la Tabla 2.

La Tabla 2 muestra en la primera columna el número de secuencia de video , en la segunda el nombre del video, en la tercera el nombre de la vista que tiene asignado para cada video, debido a que las secuencias cuentan hasta nueve vistas tomadas en un laboratorio con cámaras en ángulos ligeramente diferentes entre ellas, en la tercera columna el orden en que aparecen, primero la vista izquierda, luego central y por último vista derecha. En casos similares se presentan la cuarta y quinta columna pero para

Tabla 2. Videos de test de verificación

Seq. ID	Secuencia de video	Izq-Cent-Der	Izq-Der	Vista Principal
S01	Poznan_Hall2	7-6-5	7-5	5
S02	Poznan_Street	5-4-3	5-3	3
S03	Undo_Dancer	1-5-9	1-9	9
S04	GT_Fly	9-5-1	9-1	1
S05	Kendo	1-3-5	1-5	5
S06	Balloons	1-3-5	1-5	5
S08	Newspaper	2-4-6	2-6	6
S10	Shark	1-5-9	1-9	9

Fuente: “MV-HEVC / 3D-HEVC Verification Test Plan”.

secuencias en las que se trabajan al tiempo solo dos o una vista respectivamente.

También se observa en la Figura 8 la vista principal de uno de los videos de prueba con sus datos de profundidad y textura.

Figura 8. Vista principal de la secuencia de video “Newspaper”



Datos de profundidad (a) y de textura (b) de la secuencia de video.

Se realizaron pruebas de codificación con las secuencias de video de la Tabla 3.

En la Tabla 3 se especifica el número de frames que tiene cada video, su resolución y el número de frames que se reproducen por segundo en formato YUV 4:2:0. La secuencia *Newspaper* es de resolución estándar, la secuencia *Poznan_Street* es de resolución UltraHD con baja movilidad, y la secuencia *Undo_Dancer* es de resolución UltraHD con alta movilidad.

Para hacer la codificación de los videos se necesita un archivo de configuración y el ejecutable `./TAppEncoderStatic`, y dos archivos de configuración de extensión `cfg`, en uno de los cuales se denotan las especificaciones originales de la secuencia de video a trabajar de la Tabla 3, y en el otro archivo se realiza la configuración de los parámetros más importantes para las pruebas de eficiencia de la codificación, siendo éstos el tamaño del GOP, la estructura de codificación, la cuantización, el rango de búsqueda de correlación, entre otros parámetros.

Tabla 3. Videos de prueba en formato .YUV y especificaciones originales

Seq. ID	Secuencia de video	Frames a codificar	Resolución	Frames por segundos	YUV
S08	Newspaper	300	1024 X 768	30	4:2:0
S02	Poznan _Street	250	1920 X 1088	25	4:2:0
S03	Undo _Dancer	250	1920 X 1088	25	4:2:0

Cuando se codifican los videos con su mapa de profundidad se requieren los parámetros de las cámaras para el codificador. Un ejemplo de parámetros de cámara del video *Undo_Dancer* es dado en la Tabla 4. Estos parámetros son suministrados de acuerdo al ambiente en el que se creó el video de prueba.

Tabla 4. Parámetros de cámara para configuración de codificación

# No. Vista	Frame inicial	Frame final	Longitud focal	Posición	Turno C	Z Cerca	Z Lejos
1	0	249	2.302.852.541.609.160	-80.0	960.0	2289	213500
2	0	249	2.302.852.541.609.160	-60.0	960.0	2289	213500
5	0	249	2.302.852.541.609.160	0.0	960.0	2289	213500
9	0	249	2.302.852.541.609.160	80.0	960.0	2289	213500

Se plantearon diferentes casos de codificación, los cuales se encuentran representados en la Figura 9.

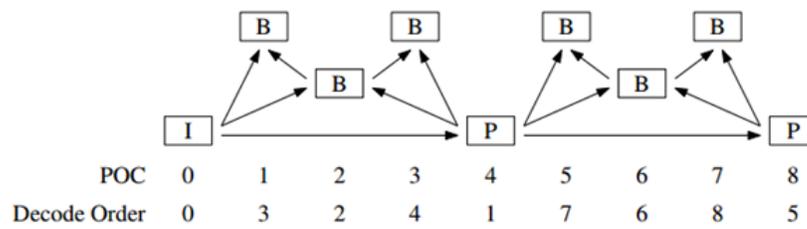
Figura 9. Casos de pruebas de codificación

2 View	IPBB	GOP Side
	IIII	
	IPPP	
2 View + Depth	IPBB	
	IIII	
	IPPP	
3 View	IPBB	
	IIII	
	IPPP	
3 View + Depth	IPBB	
	IIII	
	IPPP	

En los casos de codificación que se plantearon para los diferentes grupos de vistas

por video, se enfocó en la modificación y eficiencia de las GOP (Group Of Pictures) para la transmisión con pérdida de paquetes. Cada GOP debe iniciar con una imagen I, que son imágenes que sólo utilizan compresión intracuadro en donde cada imagen es procesada de forma independiente de las demás y contienen por si sola toda la información necesaria para su reconstrucción. Las imágenes P o imágenes de predicción se generan a partir de las imágenes I más próxima que la anteceda. Las imágenes B se generan a partir de imágenes previas tipo I o P. En la Figura 9 se referencia los diferentes tipos de GOP de tamaño 8 y rango de búsqueda y coincidencia 64 que se trabajaron para las estructuras IPPP e IPBB, mientras que las estructuras GOP IIII (AllIntra) son de tamaño 1 y rango de búsqueda y coincidencia 64. Un ejemplo de una estructura de codificación de imagen 4 y cómo se crea se encuentra en la Figura 10.

Figura 10. Una estructura GOP



Fuente: Title: HM Software Manual; Document: JCTVC-Software Manual.

5.3 TRANSPORTE Y EMPAQUETAMIENTO DE DATOS

El empaquetamiento de la codificación es una parte muy importante del estándar HEVC, ya que, con ella se obtiene una mejora significativa en la tasa de transmisión y distorsión. Los datos codificados con HEVC se encapsulan en NAL (Network Abstraction Layer) que contienen un número entero de bits. Para entender la forma en que se encapsulan las NAL en H.265 se estudian y analizan las recomendaciones ITU-T del estándar de telecomunicaciones⁴⁰. El formato de bitstream de HEVC puede catalogarse en formato de stream de unidades NAL o de stream de bytes. El formato de unidades NAL es conceptualmente el más básico, en donde las secuencias de NALs se organizan en orden de decodificación, teniendo impuestas ciertas limitaciones. El formato de stream de bytes se puede construir desde el formato de stream de unidades NALs en orden de decodificación y al prefiar cada unidad NAL con un prefijo de código de comienzo con bytes de valor 0 para formar una corriente de bytes.

En la Figura 11 se referencia los diferentes tipos de unidades NAL y su respectivo tipo de clase.

Existen 63 tipos de NAL en H.265 divididas en clase VLC y non_VLC. Las unidades VLC contienen datos que representan los valores de las imágenes del video y las unidades non_VLC contienen información adicional asociada con conjuntos de parámetros.

⁴⁰SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services-Coding of moving video

Figura 11. Tipos de NALs de H.265

Unidades NAL Clase tipo VCL		Unidades NAL Clase tipo non_VCL	
0	TRAIL_N	32	VPS_NUT
1	TRAIL_R		
2	TSA_N	33	SPS_NUT
3	TSA_R		
4	STSA_N	34	PPS_NUT
5	STSA_R		
6	RADL_N	35	AUD_NUT
7	RADL_R		
8	RASL_N	36	EOS_NUT
9	RASL_R		
10	RSV_VCL_N10	37	EOB_NUT
12	RSV_VCL_N12		
14	RSV_VCL_N14		
11	RSV_VCL_R11	38	FD_NUT
13	RSV_VCL_R13		
15	RSV_VCL_R15	39	PREFIX_SEI_NUT
16	BLA_W_LP		
17	BLA_W_RADL		
18	BLA_N_LP	40	SUFFIX_SEI_NUT
19	IDR_W_RADL		
20	IDR_N_LP		
21	CRA_NUT		
22	RSV_IRAP_VCL22	41..47	RSV_NVCL41...
23	RSV_IRAP_VCL23		
24..31	RSV_VCL24... RSV_VCL31	48..63	UNSPEC48...
			UNSPEC63

Fuente: ITU-T Rec. H.265 (10/2014) High efficiency video coding.

Las NAL son importantes para el proceso de decodificación ya que éste inicia su proceso al principio del flujo de bytes. De esta forma, según establece las recomendaciones de la ITU-T, cada NAL inicia con un patrón binario de 31 bits consecutivos iguales a 0 seguido de un bit igual a 1.

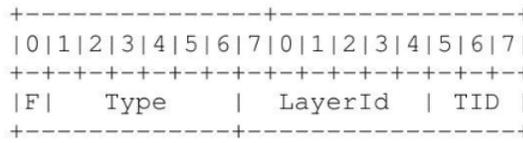
Las NAL son importantes en el proceso de empaquetamiento y transmisión, como por ejemplo en el “RTP Payload Format for High Efficiency Video Coding (HEVC)⁴¹” de la “Internet Engineering Task Force (IETF)” según el cual las cabeceras de las unidades NAL están formadas por dos bytes como se muestra en la figura 12, donde:

F: Es un parámetro usado para detectar errores de transmisión. Casi siempre es 0. El valor de 1 puede ser usado para indicar una violación a la sintaxis. Cantidad de bits: 1.

Type: Este parámetro indica el tipo de NAL al que pertenece. Cantidad de bits: 6.

⁴¹RFC 7798 RTP para el estándar de codificación ITU-T de la recomendación e h.265 SO/IEC International Standard 23008-2

Figura 12. cabecera de unidades NAL H.265



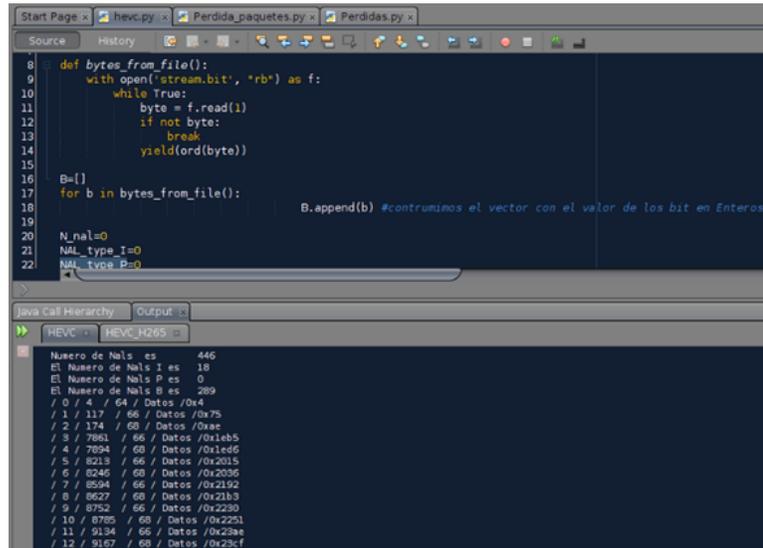
Fuente: RTP Payload Format for High Efficiency Video Coding

LayerId: Este parámetro es normalmente 0 en codificación de video 2D, pero para la codificación de video 3D, este elemento de sintaxis es usado para identificar las capas adicionales que se presentan en las diferentes vistas. Cantidad de bits: 6.

TID: Este campo es especial para la identificación temporal de la unidad NAL. Cantidad de bits: 3.

Se desarrolló un componente de software en lenguaje *Python3* que analiza la correlación que existe entre la capa de profundidad y textura planteado como objetivo, identificando la cantidad de NALs, el tamaño de los paquetes, el tipo de NAL y la vista a la cual pertenece. Evidencias del análisis del empaquetamiento se encuentran en la Figura 13.

Figura 13. Análisis del empaquetamiento de datos. Identificación de NALs



5.4 SIMULACIÓN DE LA TRANSMISIÓN INALÁMBRICA

5.4.1 Verificación de los esquemas de transmisión

Se implementaron 3 diferentes esquemas MIMO, los cuales deben ser verificados primero. El primer esquema es el tradicional Alamouti 2x2, con un máximo orden

de diversidad de $2N_{rx}$, donde N_{rx} es el número de antenas receptoras, y tasa de bloque de 1, ya que transmite 2 símbolos cada dos intervalos de tiempo. El código de bloque de este esquema se expresa:

$$G2 = \begin{pmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{pmatrix} \quad (3)$$

El segundo esquema es un MIMO 4x4 con tasa de bloque de 1/2 con máximo orden de diversidad de $4N_{rx}$, permitiendo la transmisión de 4 símbolos cada 8 instantes de tiempo. Su código de bloque se define de la siguiente manera:

$$G2 = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 \\ -s_2 & s_1 & -s_4 & s_3 \\ -s_3 & s_4 & s_1 & -s_2 \\ -s_4 & -s_3 & s_2 & s_1 \\ s_1^* & s_2^* & s_3^* & s_4^* \\ -s_2^* & s_1^* & -s_4^* & s_3^* \\ -s_3^* & s_4^* & s_1^* & -s_2^* \\ -s_4^* & -s_3^* & s_2^* & s_1^* \end{pmatrix} \quad (4)$$

El tercer esquema es un MIMO 8x8 con tasa de bloque de 1/2 con máximo orden de diversidad de $8N_{rx}$, transmitiendo 8 símbolos cada 16 instantes de tiempo. Su representación de bloque es la siguiente:

$$G2 = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ -s_2 & s_1 & s_4 & -s_3 & s_6 & -s_5 & -s_8 & s_7 \\ -s_3 & -s_4 & s_1 & s_2 & s_7 & s_8 & -s_5 & -s_6 \\ -s_4 & s_3 & -s_2 & s_1 & s_8 & -s_7 & s_6 & -s_5 \\ -s_5 & -s_6 & -s_7 & -s_8 & s_1 & s_2 & s_3 & s_4 \\ -s_6 & s_5 & -s_8 & s_7 & -s_2 & s_1 & -s_4 & s_3 \\ -s_7 & s_8 & s_5 & -s_6 & -s_3 & s_4 & s_1 & -s_2 \\ -s_8 & -s_7 & s_6 & s_5 & -s_4 & -s_3 & s_2 & s_1 \\ s_1^* & s_2^* & s_3^* & s_4^* & s_5^* & s_6^* & s_7^* & s_8^* \\ -s_2^* & s_1^* & s_4^* & -s_3^* & s_6^* & -s_5^* & -s_8^* & s_7^* \\ -s_3^* & -s_4^* & s_1^* & s_2^* & s_7^* & s_8^* & -s_5^* & -s_6^* \\ -s_4^* & s_3^* & -s_2^* & s_1^* & s_8^* & -s_7^* & s_6^* & -s_5^* \\ -s_5^* & -s_6^* & -s_7^* & -s_8^* & s_1^* & s_2^* & s_3^* & s_4^* \\ -s_6^* & s_5^* & -s_8^* & s_7^* & -s_2^* & s_1^* & -s_4^* & s_3^* \\ -s_7^* & s_8^* & s_5^* & -s_6^* & -s_3^* & s_4^* & s_1^* & -s_2^* \\ -s_8^* & -s_7^* & s_6^* & s_5^* & -s_4^* & -s_3^* & s_2^* & s_1^* \end{pmatrix} \quad (5)$$

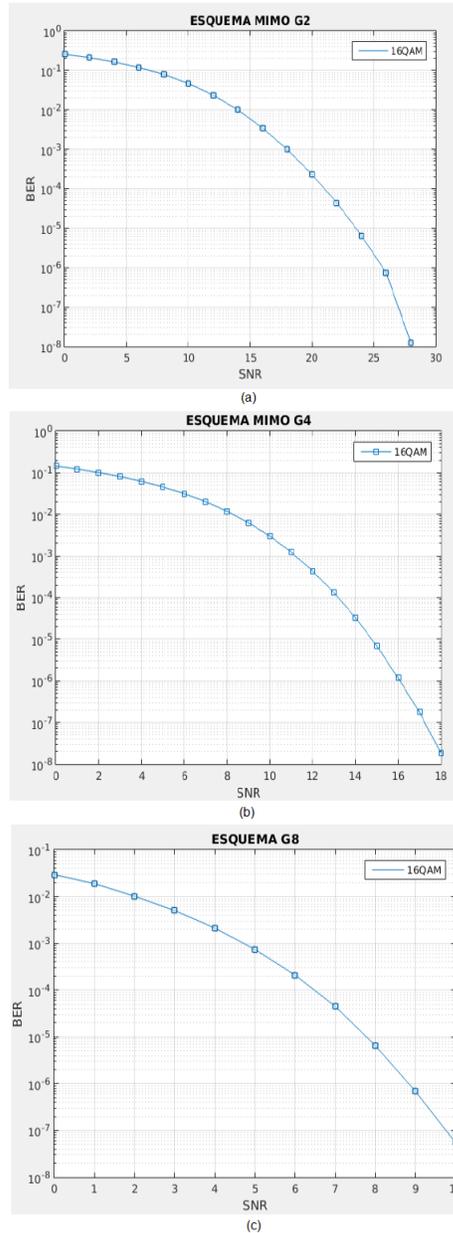
Los diagramas de bloques se encuentran conformados por los símbolos s que se transmiten y sus respectivas conjugadas complejas s^* . Para el esquema MIMO 2X2, la distribución de los símbolos es de naturaleza OSTBC, y los esquemas MIMO 4x4 y 8x8 son de naturaleza STBC.

Los 3 esquemas fueron implementados en lenguaje C++ utilizando librerías de comunicaciones IT++⁴², realizando la modulación QAM de orden 16 a diferentes valores de relación Señal/Ruido (SNR) y utilizando el método de Montecarlo, tal como se

⁴²IT++ Documentation. En línea: <http://itpp.sourceforge.net/4.3.1/> Última sesión: 13-Abril.2016

plantea en otros trabajos⁴³, para tener como valor final la tasa de error de bits (BER), parámetro necesario para el análisis de la pérdida de paquetes para un caso real de transmisión de video 3D. Los resultados de la verificación de los esquemas se muestran en la Figura 14.

Figura 14. Verificación de los esquemas MIMO



(a) Esquema 2X2. (b) Esquema 4X4. (c) Esquema 8X8.

⁴³Dorado, S., Santacruz, J. Pino, F. (2014). Propuesta de estructuras con tensores para la implementación de esquemas MIMO. Revista ST, 12(29), 85-102

Se puede observar en la figura que la tasa de error de bits disminuye a medida que aumenta la complejidad del esquema. Esto se debe gracias a la ganancia de diversidad de cada esquema, que aumenta acorde el número de antenas aumenta.

5.4.2 Implementación del codificador de canal

Una vez los esquemas de transmisión hayan sido verificados, se procede a implementar el codificador de canal. En este trabajo se configuró el Turbo Códec, ya que es el indicado para obtener resultados favorables en base a la relación señal a ruido.

A la hora de implementar el Turbo Códec en los esquemas de transmisión con la librería IT++⁴⁴, se establecen ciertos parámetros como lo son:

gen1: Un vector con elementos *n1* que contienen los polinomios generadores para el primer codificador constituyente.

gen2: Un vector con *n2* elementos que contienen los polinomios generadores para el segundo codificador constituyente.

constraint_length: la longitud de restricción de los dos codificadores constituyentes.

interleaver_sequence: Un vector imaginario definiendo el entrelazador turbo interno.

in_iterations: El número de iteraciones de decodificación. El valor por defecto es 8.

in_metric: Determina la métrica decodificador: "MAP", "LOGMAP", "LOGMAX", o "TABLE". El valor por defecto es "LOGMAX".

in_logmax_scale_factor: La información extrínseca de cada decodificador constituyente es optimista cuando se utiliza la decodificación LOGMAX. Este parámetro permite una reducción a escala de la información extrínseca que se pasará al siguiente decodificador. El valor por defecto es 1,0. Este parámetro se ignora para otras métricas diferentes a "LOGMAX".

in_adaptive_stop: Si este parámetro es cierto, entonces las iteraciones se detendrá si los resultados de la decodificación después de una iteración completa es igual a la iteración anterior. El valor por defecto es *false*.

lcalc: Este parámetro se puede utilizar para proporcionar una *LLR_calc_unit* específica que define la resolución en la *table-lookup* si se utiliza la decodificación con la métrica "TABLE".

⁴⁴Turbo Codec Documentation (2013, julio 6). Recuperado de http://itpp.sourceforge.net/4.3.1/classitpp-1-1Turbo_Codec.html

5.5 PÉRDIDA DE PAQUETES

Con los resultados de BER del apartado anterior, se formula la pérdida de paquetes en un entorno MIMO, PER_{MIMO} . Para la estimación de PER_{MIMO} cada bit dentro del paquete tiene el mismo BER y los errores de bits no están correlacionadas, entonces PER_{MIMO} puede estar relacionado con el BER a través de un paquete que contiene L bits⁴⁵:

$$PER_{MIMO} = 1 - (1 - BER)^L \quad (6)$$

Con este valor de PER_{MIMO} , la identificación de la cantidad de NALs de cada bitstream, que el tamaño de las unidades NAL es variable y tomando como referencia que el tamaño máximo de empaquetamiento TCP de un MSS (Maximum Segment Size) según la norma RFC 6691 es de 525 bytes⁴⁶, se estima el número de paquetes perdidos:

$$Packet_{Lost} = \frac{N_{NALs} * \frac{M_{NALs}}{525} * PER_{MIMO}}{100} \quad (7)$$

Donde N_{NALs} es la cantidad total de unidades NAL en el bitstream, M_{NALs} es el tamaño promedio de las NALs en el bitstream que se está analizando y PER_{MIMO} es la pérdida de paquetes en el entorno MIMO.

5.6 EVALUACIÓN OBJETIVA

Para realizar la evaluación objetiva se decodifica los bitstream con los diferentes patrones de pérdidas de paquetes, proceso en el cual se obtiene el video en formato .YUV y se busca determinar la relación señal a ruido pico (PSNR) para los frames que forman el video original y el video degradado y así determinar el nivel de calidad al comparar los dos valores.

Para calcular el PSNR, primero tenemos que determinar el *Error Cuadrático Medio* (*MSE*) de cada fotograma mediante la ecuación:

$$MSE = \frac{1}{MNT} \sum_{i=1}^M \sum_{i=1}^N \sum_{i=1}^T [x(m, n, t) - y(m, n, t)]^2 \quad (8)$$

⁴⁵Obando. M, Freitas. W Jr. Cavalcanti. F; Switching Between Hybrid MIMO Structures for Video Transmission Based on Distortion Model; Brazil (2010)

⁴⁶Borman, D; TCP Options and Maximum Segment Size (MSS); EN línea: <https://tools.ietf.org/html/rfc6691> Última sesión: 23-Julio-2016

Donde la imagen tiene $M \times N$ píxeles y T frames; x , y corresponde a los píxeles de la señal original y la señal degradada respectivamente.

En este trabajo, al ya tenerse ésta medida para un par de frames (original - degradado) en un plano (R, G o B), solo es necesario realizar el mismo cálculo para los demás pares de frames y luego promediarlos. Luego se hace lo mismo para los otros dos planos.

$$MSE_v = \frac{1}{3F} \left(\sum_{j=1}^F MSE_R(j) + \sum_{j=1}^F MSE_G(j) + \sum_{j=1}^F MSE_B(j) \right) \quad (9)$$

$$PSNR = 10 \log_{10} \frac{L^2}{MSE_v} \quad (10)$$

Donde F es el número de frames y $MSE_R(j)$, $MSE_G(j)$, $MSE_B(j)$ es el valor del error cuadrático calculado para el frame j -ésimo en el plano R, G, B.⁴⁷

Finalmente, se expone la Tabla 5 que relaciona el PSNR con el MOS

Tabla 5. Relación entre los valores de PSNR y MOS.

PSNR (dB)	MOS
Mayor a 37	5
31 - 37	4
25 - 31	3
20 - 25	2
Menor a 20	1

Fuente: Suarez. J, Valencia. O; Diseño de un modelo objetivo para evaluar la calidad de video 3D anaglifo; Tesis de grado Ingeniería Electrónica, pág 46; 2016.

Durante el desarrollo del proyecto se trabajó con un componente de software programado en C++ con ayuda de la librería de tratamiento de video e imágenes *OpenCV* y se evaluaron el PSNR de los frames, pero por razones de agilidad en el proceso de evaluación, las pruebas se hicieron con el software JMVC de h.264⁴⁸.

⁴⁷Suarez. J, Valencia. O; Diseño de un modelo objetivo para evaluar la calidad de video 3D anaglifo; Tesis de grado Ingeniería Electrónica, 2016.

⁴⁸MVC Software Manual Version:JMVC 8.5 (CVS tag: JMVC₈₅)Últimasesión : 31 – Julio – 2016

5.7 EVALUACIÓN SUBJETIVA

En este apartado se detalla la implementación de la prueba subjetiva realizada durante la ejecución del proyecto. El objetivo de este experimento es obtener la valoración subjetiva de la calidad de un número determinado de secuencias de video por parte de espectadores, ya que siendo éstos los receptores finales, se consigue más seguridad y fiabilidad en la determinación de la calidad de las secuencias.

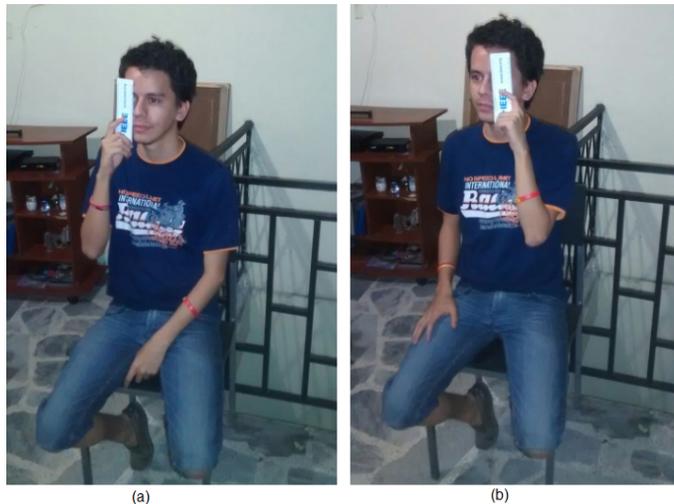
Antes de comenzar la evaluación se realiza una prueba Snellen tal como se sugiere en la recomendación ITUT 500-11 [18], luego se adecúa un escenario a ciertos requerimientos y se entrega un formato en físico para que el espectador diera su opinión de las secuencias proyectadas.

5.7.1 Prueba Snellen

Antes de dar inicio con las pruebas subjetivas, los espectadores realizaron la prueba inicial de agudeza visual normal (o corregida) conocida como prueba Snellen. Esta prueba es sugerida por la recomendación ITUT 500-11⁴⁹.

El panel Snellen fué impreso en papel de formato estándar A4 y los espectadores se ubicaron 2,8 metros frente a ella. El procedimiento de esta prueba inicial fué la siguiente: Se inició con la lectura de las letras del panel con el ojo izquierdo, cubriendo el derecho sin hacer presión, y después se realiza el mismo procedimiento con el ojo contrario. Ésta situación se representa en la Figura 15.

Figura 15. Implementación de la prueba Snellen



(a) Exámen del ojo Inquierdo. (b) Exámen del ojo Derecho.

⁴⁹ITU-R BT Recommendation. “500-11: Methodology for the Subjective Assessment of the Quality of Television Pictures. 500-11”. En: Standardization Sector of International Telecommunication Union, ITU Telecom (2002).

Si el espectador evaluado logra leer la línea correspondiente a 20/20 en el panel Snellen, su visión es óptima, y es ingresado a la sala adecuada para las sesiones de la prueba subjetiva.

5.7.2 Condiciones de Observación

Las condiciones empleadas para la adecuación del lugar donde se proyectan las secuencias se siguieron bajo la recomendación ITU-T P.910 [19]. Para el desarrollo de la evaluación se empleó un monitor LG de 42 pulgadas con una resolución de 1920x1080 píxeles. La distancia de observación fue de 5 veces la altura de la pantalla del monitor.

Las condiciones de iluminación de fondo menor a 20 lux (ésto es una unidad de medida de luminancia) requeridas se dieron gracias a la adecuación de una habitación donde no entraba luz solar y las ventanas se encontraban cubiertas de cortinas de 4 milímetros de espesor.

Las pruebas consistieron en la proyección de 5 sesiones, cada una con 2 secuencias de video 3D de tal manera de que no sobrepasaran 20 minutos. En cada sesión se proyectaba la secuencia original de video 3D *Undo_Dancer* en UltraHD el cual tiene una duración de 9 segundos, seguidamente un espacio de 5 segundos de relajación para finalmente proyectar la secuencia de video decodificada a evaluar. Terminada la proyección de ésta última secuencia, se daba un estimado de 40 segundos para que los espectadores calificaran la secuencia de video de acuerdo a las indicaciones dadas en la introducción por uno de los integrantes del proyecto.

Por lo tanto, cada sesión se estimó en:

Tiempo de la sesión = 2 secuencias de video 3D de 9 segundos + 5 segundos de relajación + 40 segundos para evaluar el video = $(18 + 5 + 40)$ segundos = 63 segundos.

Durante la prueba, hay un espacio de 60 segundos entre cada sesión ya que sus proyecciones se realizaban de forma manual, dando una breve explicación a los espectadores del tipo de secuencias que encontrarán en la nueva sesión. Al tener en total 5 sesiones de 63 segundos y 4 espacios de 60 segundos, un espacio introductorio de 3 minutos y otro de comentarios y conclusiones de los espectadores de 3 minutos, se tiene como estimación de la prueba:

Tiempo de la prueba = $((63 \times 5) + (60 \times 4) + 180 + 180)$ segundos = 915 segundos.

Por lo tanto, la prueba por grupo tuvo una duración de 15 minutos con 15 segundos.

El escenario donde se realizó la prueba subjetiva se contempla en la Figura 16. Se pueden apreciar las condiciones de iluminación.

5.7.3 Realización de la evaluación subjetiva

Para la ejecución de la prueba subjetiva se tomó el método DSIS (Escala de degradación con doble estímulo) debido a su poca complejidad y permite calificar el nivel de degradación entre el video original y el codificado. Al principio de la prueba se tomó

Figura 16. Escenario para la evaluación subjetiva

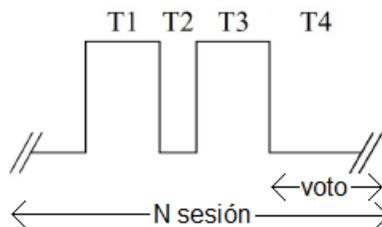


(a) Vista desde el espectador. (b) Vista desde el monitor.

un espacio de 3 minutos para explicar a los espectadores el tipo de evaluación, qué tipo de video 3D iban a visualizar, la escala de evaluación y la duración aproximada de la prueba. La muestra total al final de la prueba es de 18 usuarios.

Según las recomendaciones internacionales para la realización de éste tipo de evaluación, el número de observadores mínimo para realizar la prueba es de quince. Se dejó claro que el orden de las sesiones no necesariamente correspondía al orden de calidad de las secuencias de video 3D y que la nota a elegir en el formato en físico entregado al iniciar sea basada en la impresión global que perciben de los videos.

Figura 17. Estructura de presentación del material de prueba



Períodos de presentación.

La estructura de presentación de las sesiones de prueba subjetiva se muestra en la Figura 17. El tiempo T1 corresponde a la duración del video de referencia; T2 es el

período de relajación entre el video de referencia y el video a evaluar; T3 es la duración al video codificado y degradado el cual los espectadores darán su calificación subjetiva comparándolo con el video de referencia; y T4 corresponde al espacio asignado para calificar el video en el formato entregado en físico.

El método DSIS tiene un esquema de valores para ser interpretados en escala MOS. De esta manera, los espectadores escogen una de las cinco opciones dadas en el formato físico, tal como se estipula en la recomendación. Este esquema de valores se representa en la Tabla 6.

Tabla 6. Esquema de valores para DSIS

La degradación es imperceptible	5
La degradación es perceptible pero no molesta	4
La degradación es ligeramente molesta	3
La degradación es molesta	2
La degradación es muy molesta	1

Fuente: ITU-R BT Recommendation. "500-11: Methodology for the Subjective Assessment of the Quality of Television Pictures. 500-11".
 En: Standardization Sector of International Telecommunication Union, ITU Telecom (2002).

6. ANÁLISIS Y RESULTADOS

En este capítulo se analizan los resultados obtenidos de la implementación de los esquemas de transmisión y las evaluaciones objetivas y subjetivas de video 3D de múltiples vistas, para evaluar la calidad de video 3D cuando es transmitido utilizando esquemas MIMO de diversidad.

6.1 ANÁLISIS Y RESULTADOS DE LOS ESQUEMAS DE TRANSMISIÓN

En este apartado se explica el procedimiento realizado para obtener la tasa de error de bits a partir de valores de relación señal a ruido. Se examinaron los resultados obtenidos de los canales de transmisión inalámbricos que se diseñaron con la implementación del codificador de canal.

Ésta simulación se realizó con una señal de entrada generada con bits de forma aleatoria cuyo tamaño fuera igual o superior a 1.04MB (1.048.576 bytes), la modulación fue la misma planteada para la verificación de los sistemas (16QAM) y se obtuvo diez mil (10.000) iteraciones de BER por cada valor de SNR estimado. Éstas iteraciones obedecen al método de Montecarlo, con el fin de calcular un valor preciso de BER en cada proceso. Los resultados de ésta simulación se muestran en la Figura 18.

A partir de los resultados obtenidos en la Figura 18, podemos determinar que:

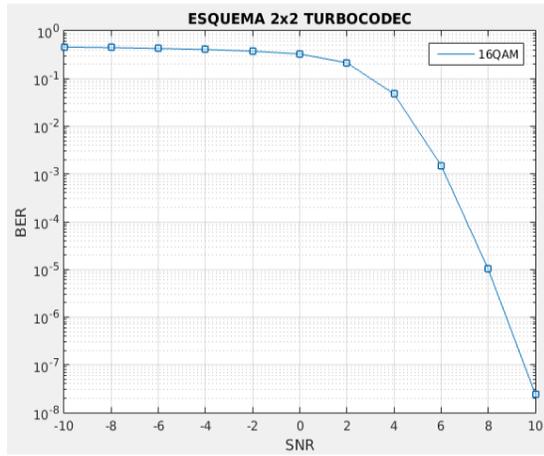
- La complejidad de los esquemas sigue jugando un papel importante, ya que, a medida que la ganancia de diversidad aumenta, la tasa de error de bits disminuye.
- Al implementar el codificador de canal en alguno de los esquemas, se mejora la tasa de error de bits al disminuir de forma considerable la relación señal/ruido (SNR) del sistema.
- La transmisión de video 3D requiere de codificación de canal, y se trabaja con una tasa de error de bits de 10^{-4} con relación señal/ruido relativamente baja, por lo que consideramos como los mejores canales para la transmisión los esquemas 4X4 y 8X8.

Una vez teniendo los valores de BER de cada esquema, se realiza el análisis de la pérdida de paquetes para un caso real de transmisión de video 3D

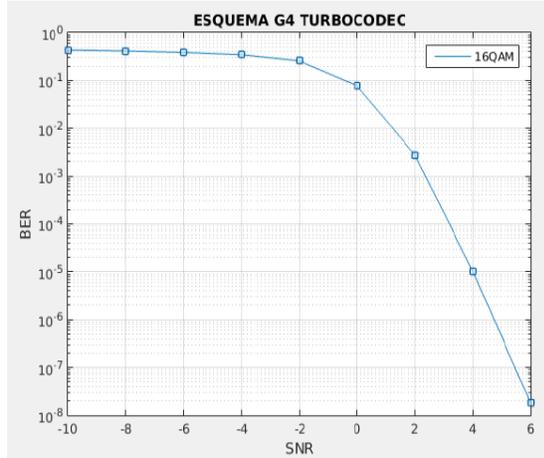
6.2 ANÁLISIS Y RESULTADOS DEL ESTÁNDAR HEVC

En este apartado se realizó pruebas de codificación enfocada en trabajar estructuras Group Of Pictures (GOP) que no fueran afectadas por la pérdida de paquetes. Teniendo en cuenta las experiencias con codificación de video 2D, se concluyó que la mejor estructura para codificar el video y minimizar las pérdidas de paquetes era la IPPP.

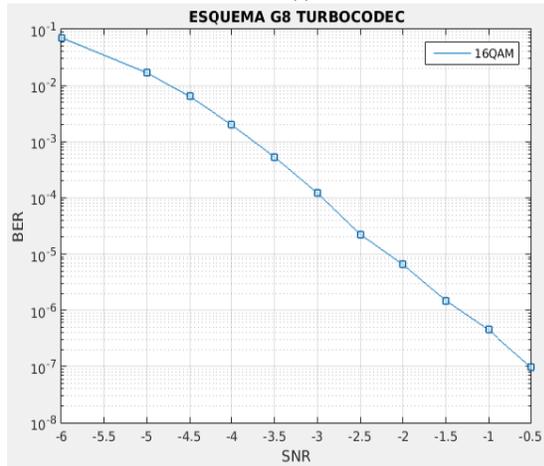
Figura 18. Resultados esquemas MIMO con Turbo C3dec



(a)



(b)

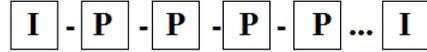


(c)

(a) Esquema 2X2. (b) Esquema 4X4. (c) Esquema 8X8.

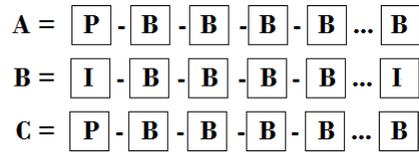
Este tipo de estructura se representa en la Figura 19.

Figura 19. Estructura IPPP



Debido que en el video 3D se trabajan dos o más vistas, teniendo siempre una vista base o principal, se hicieron pruebas de esquemas de codificación con estructuras GOP de 3 vistas como la que se expresa en la Figura 20.

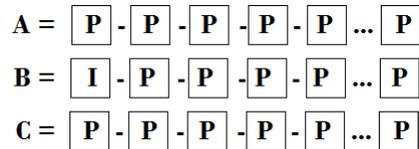
Figura 20. Primera estructura GOP de 3 vistas



En donde cada línea representa la GOP de cada vista, en la cual, la vista B es la vista base o principal. Durante las pruebas se pudo evidenciar que con este tipo de estructuras, no se puede perder ningún paquete o bit que se vaya a transmitir y la única forma de transmitir con este tipo de estructura GOP es orientada en el protocolo TCP.

También se probó la estructura GOP para las tres vistas representada en la figura 21.

Figura 21. Segunda estructura GOP de 3 vistas



Para la construcción y diseño de la anterior estructura GOP se tuvo como base la estructura IPPP del archivo cfg de video 2D, cuya información se muestra en la Figura 22.

Figura 22. Estructura IPPP del archivo cfg de video 2D

	I	II	III	IV	V	VI	VII	VIII	IX	X				XI	XII	XIII					
Frame1:	P	1	3	0.4624	0	0 0	4	4	4	-1	-5	-9	-13	0							
Frame2:	P	2	2	0.4624	0	0 0	4	4	4	-1	-2	-6	-10	1	-1	5	1	1	0	1	
Frame3:	P	3	3	0.4624	0	0 0	4	4	4	-1	-3	-7	-11	1	-1	5	0	1	1	1	1
Frame4:	P	4	1	0.578	0	0 0	4	4	4	-1	-4	-8	-12	1	-1	5	0	1	1	1	1

En donde la información más relevante para entender la estructura de codificación en el archivo cfg es la primera columna que define el número de frames; la segunda

columna que define el tipo de imagen como se va a codificar sea I, P o B; la tercera columna que define el orden de codificación; y la décima columna que define el rango de interdependencia para la codificación entre los diferentes frames del video teniendo como referencia el frame actual.

Se evidenció que si se usa la estructura GOP de la Figura 20 de jerarquía P en lugar de jerarquía B surgen una gran cantidad de pérdidas para video 3D ya que la distancia entre la imagen objetivo y la imagen de referencia se incrementa, y ese aumento no proporciona ningún beneficio de codificación por bi-predicción de imágenes codificadas posteriormente. Sin embargo, se recuerda que utilizar simplemente la estructura IPPP en video 2D genera pérdidas aceptables.

Para las pruebas de este trabajo orientado a la pérdida de paquetes se concluyó que la mejor estructura GOP a utilizar era All-Intra. Su estructura se representa en la Figura 23. Esta estructura es de un archivo cfg para una vista de video 3D.

Figura 23. Estructura All-Intra

	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII
FrameI:	I	0	0	0.442	0	0	0	0	0	0	0	0	1
Frame1:	B	1	1	0.442	0	0	0	4	0	0	0	0	1

Se codificaron dos secuencias de video de la Tabla 3 de alta y baja movilidad con estructura GOP All-Intra, y se aplicó la fórmula de pérdida de paquetes nombrada en la ecuación (7) para cada valor de tasa de error de bits, BER, de cada relación señal a ruido, SNR, resultantes del apartado anterior.

6.2.1 Video 3D de 2 vistas

Para cada valor de BER se hicieron 40 iteraciones de pérdida de paquetes y cada una de las cuales se decodificó y se realizó la medida del PSNR de cada vista.

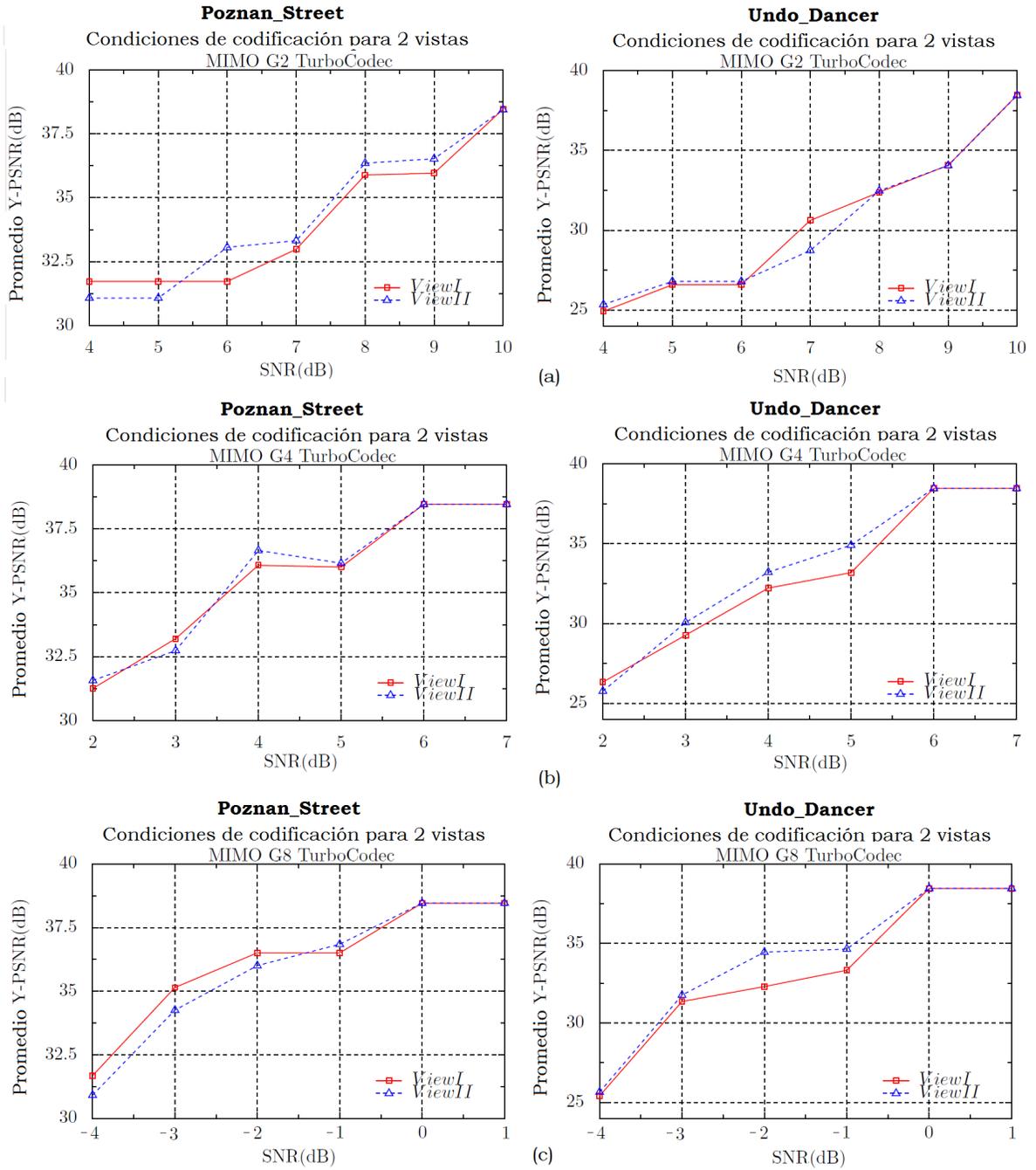
La relación de SNR contra PSNR se muestra en la Figura 24. Donde el valor de PSNR es la media de las 40 iteraciones con cada valor de BER de los tres esquemas de transmisión trabajados. Las gráficas de la derecha pertenecen a los resultados obtenidos con la secuencia de video *Poznan_Street*, y las de la izquierda con la secuencia de video *Undo_Dancer*.

Las gráficas demuestran que en un caso real de transmisión inalámbrica de video 3D de múltiples vistas, a mayor complejidad de los canales de transmisión, menor son las pérdidas de paquetes. Ésta pérdida al mismo error de BER afecta más la calidad del video de alta movilidad *Undo_Dancer* que al de baja movilidad *Poznan_Street*.

En los valores más altos de SNR no se presentan pérdidas de paquetes como lo es en el caso del esquema 8x8, donde a valores positivos de SNR la calidad es excelente.

Se pudo evidenciar que al presentarse pérdidas de paquetes en la transmisión de video 3D, la pérdida es más influyente sobre una de las vistas que sobre el resto, de modo que la calidad general del video no se ve afectada en gran medida, y teniendo en cuenta que

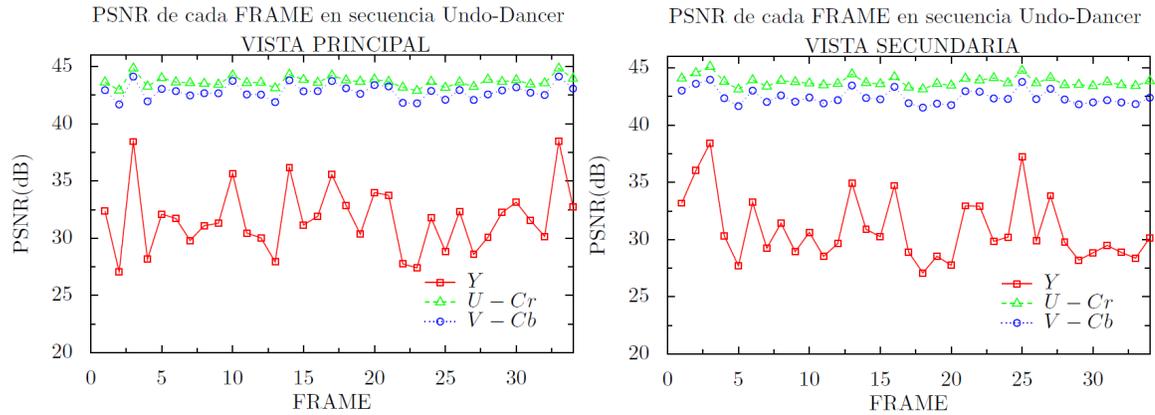
Figura 24. Gráficas de la relación de SNR contra PSNR Video 3D de 2 vistas



Relación de SNR contra PSNR para los casos de transmisión en los esquemas MIMO 2X2 (a), 4X4 (b) y 8X8 (c).

la estructura GOP trabajada es All-Intra, las pérdidas en una de las vistas no afecta en gran medida las vistas restantes, como lo evidencia la Figura 25.

Figura 25. Muestra de datos de la secuencia *Undo_Dancer*



Donde podemos ver que el componente de luminancia (Y) es el más afectado por las pérdidas y los componentes de crominancia azul y roja (Cb y Cr) son afectadas en menor medida, ya que la estructura .YUV de los videos trabajados es 4:2:0. Cada punto representa una iteración de los datos obtenidos en el esquema 8x8 con la secuencia *Undo_Dancer*.

En la Figura 26 se aprecia la representación de la distribución del rango de valores del esquema 8x8 en el cual se encuentra las 40 pruebas tomadas por cada SNR de la secuencia de video *Undo_Dancer*, donde la distribución de datos es muy amplia, por lo que en las gráficas de la Figura 24 se trabajó únicamente con la media.

Figura 26. Gráfica PlotBox 8X8 *Undo_Dancer*

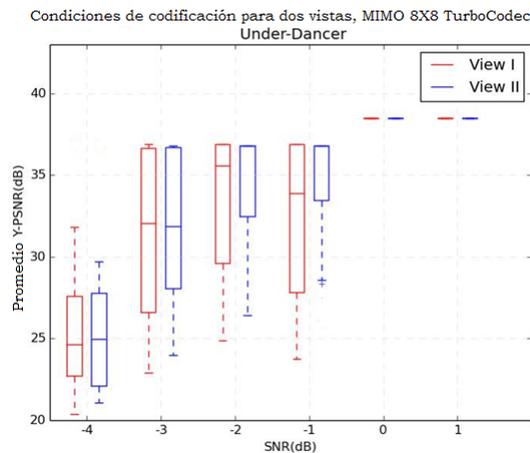
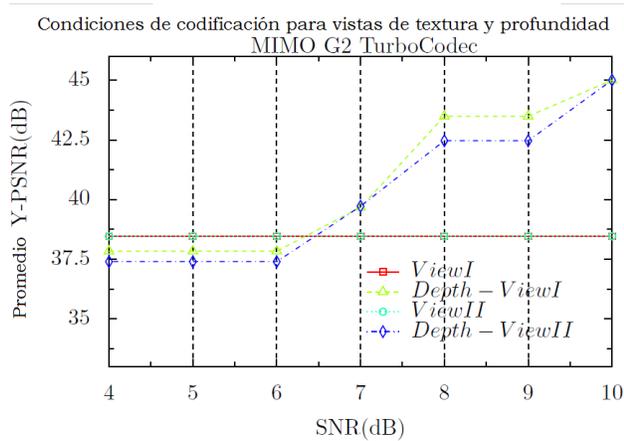


Diagrama PlotBox de cada vista en cada caso de SNR muestra los valores máximos y mínimos, los cuartiles de 75(%) (Q1), 50(%) (mediana) y 25(%) (Q3) y los valores atípicos.

6.2.2 Video 3D de 2 vistas más profundidad y renderización

Se codificó la secuencia de video *Poznan_Street* de dos vistas, cada una con su correspondiente capa de profundidad, resultando en 4 vistas para la transmisión con estructura GOP All-Intra. Durante las pruebas se pudo evidenciar que en este tipo de secuencias de video sólo se pueden perder datos o paquetes que correspondan a las vistas de profundidad, ya que al perderse información de las vistas de textura, el proceso de decodificación resulta fallido y se pierde toda la información. Los resultados de esta prueba se evidencian en la Figura 27.

Figura 27. Relación de SNR contra PSNR Video 3D de 2 vistas con profundidad



Esta gráfica es de la codificación usando los valores del esquema MIMO 2X2.

Las vistas de textura se mantienen constantes en el rango de calidad excelente mientras que las capas de profundidad pierden información. Después de que esta información sea decodificada, se realiza el proceso de renderización, mediante el cual se sintetizan 3 vistas de textura con la información de las capas de profundidad y del archivo cfg de las cámaras del video original. Los resultados de la evaluación del video renderizado con pérdidas después de la transmisión se visualizan en la Figura 28.

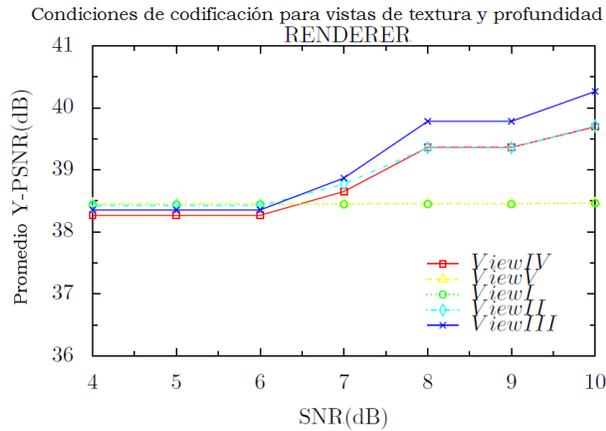
Podemos observar que las vistas I y V corresponden a las vistas de textura transmitidas sin pérdidas mientras que las vistas II, III y IV son las que resultaron de la renderización. Estas vistas tienen pérdidas ya que fueron creadas de la información de las capas de profundidad con pérdidas de paquetes. Vale aclarar que las pérdidas no son significativas para afectar la calidad del video.

6.2.3 Calidad de codificación del estándar HEVC

Para evaluar la calidad a diferente tasa de bit del estándar de codificación HEVC se varió el parámetro QP (Quantization Parameter), el cual, es inversamente proporcional a la calidad del video. El parámetro se puede modificar en enteros de intervalo de 21 a 54. Los resultados para la secuencia de video *Newspaper* con las especificaciones de la Tabla 4 con estructura GOP IBBB se muestran en la Figura 29.

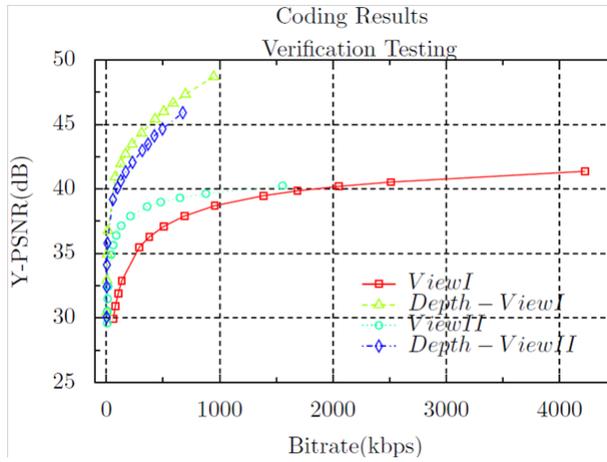
Teniendo en cuenta esta Figura, se puede decir que a mayor tasa de bits, mayor calidad

Figura 28. Renderización



Esta gráfica es de la renderización de la información de las vistas codificadas de la Figura 27.

Figura 29. PSNR Newspaper



de codificación.

6.3 ANÁLISIS Y RESULTADOS DE LA EVALUACIÓN SUBJETIVA

En este apartado se examinan los resultados obtenidos de la prueba subjetiva. Como primer punto a tratar, todos los datos e información obtenidos de los 18 usuarios que participaron en la prueba subjetiva y sus respectivas calificaciones en escala MOS fueron tabuladas y ordenadas en una hoja de cálculo.

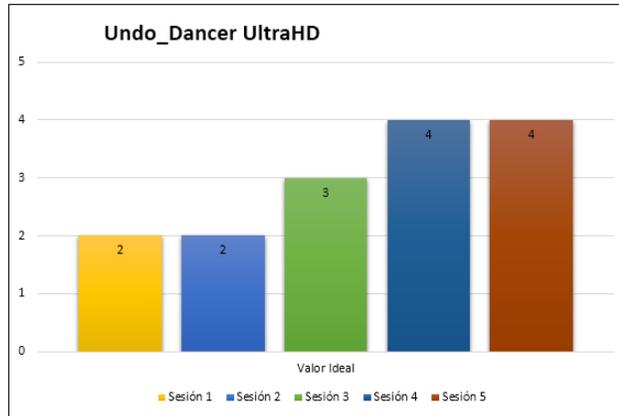
Cada secuencia de video tiene un valor diferente de PSNR. Estos valores se encuentran en la Tabla 7. Podemos observar que las sesiones fueron proyectadas de menor a mayor calidad, comparándolas con la calidad del video original.

Tabla 7. Valores PSNR de las secuencias de video 3D

Tipo de Vista	Sesión 1 (dB)	Sesión 2 (dB)	Sesión 3 (dB)	Sesión 4 (dB)	Sesión 5 (dB)	ORIGINAL (dB)
Principal	21,6872	24,2405	28,9508	33,8673	36,7061	38,4462
Secundaria	22,1322	24,1617	27,8781	34,1467	36,7978	38,4462

Tomando como referencia la Tabla 5 donde se relaciona los valores PSNR con la escala MOS y la Tabla 6 que habla del esquema de valores para la evaluación DSIS, se determinó que las secuencias de video 3D evaluadas en las sesiones 1 y 2 corresponden a escala MOS de orden 2 (“La degradación es molesta”), la secuencia en la sesión 3 corresponde a escala MOS de orden 3 (“La degradación es ligeramente molesta”), y las secuencias de las sesiones 4 y 5 a la escala MOS de orden 4 (“La degradación es perceptible pero no molesta”). El video 3D original usado como referencia para la calificación subjetiva corresponde a escala MOS de orden 5 (“La degradación es imperceptible”). Ésta información se visualiza en la Figura 30.

Figura 30. Medida MOS ideal



Luego de tener una medida MOS promedio para cada sesión calificada de forma subjetiva, se procedió a graficar los resultados como se observa en la Figura 31.

Con los resultados visualizados en las Figuras 30 y 31, se logra determinar que:

- La secuencia de video 3D evaluada en la quinta sesión fué la mejor valorada por los espectadores, lo cual es aceptable ya que el valor de PSNR de las vistas del video 3D proyectado en ésta sesión eran las más altas. De igual manera, se puede observar que los espectadores dieron las peores calificaciones para las secuencias de video proyectadas en las sesiones 1 y 2, lo cual también es aceptable al tener sus vistas, los valores de PSNR mas bajos trabajados. Éstas comparaciones se pueden visualizar en la Figura 32.
- Se puede apreciar en la gráfica que la escala de menor a mayor calidad coincide con el orden de las sesiones de las secuencias de video proyectadas.
- La calificación subjetiva de las sesiones 1 y 2 son similares aunque son aproximadas

Figura 31. Medida MOS Prueba Subjetiva

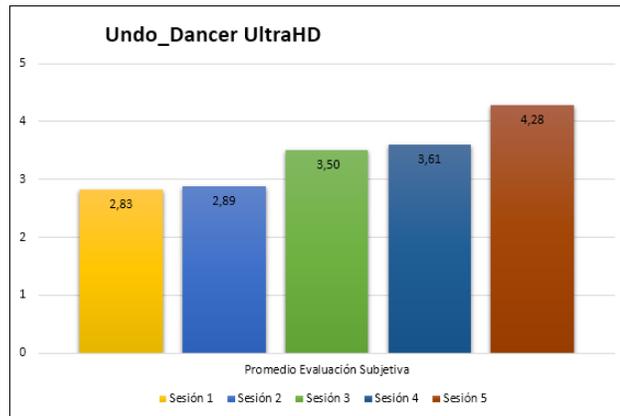
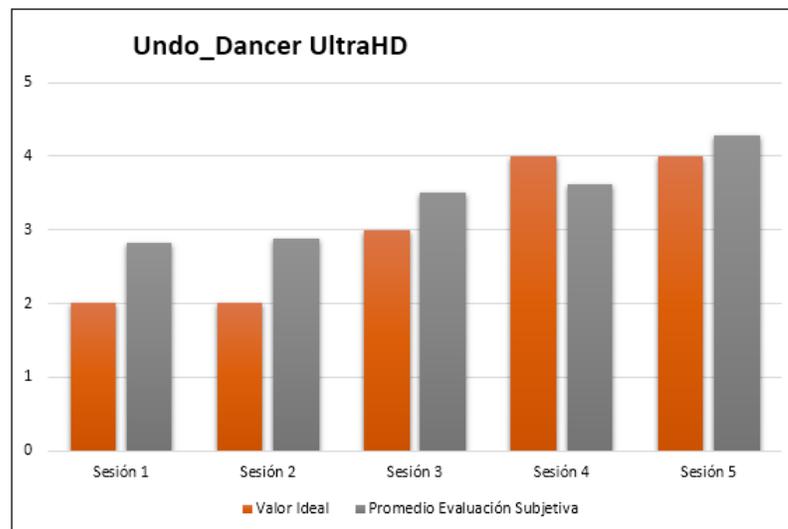


Figura 32. Comparación entre valor ideal de MOS y MOS promedio de la evaluación subjetiva



a la escala MOS de orden 3. Esto se debe a las calificaciones tan variantes de los espectadores por su percepción visual. Sin embargo, siguen siendo las secuencias con calificaciones más bajas de todas las evaluadas.

El objetivo de esta prueba subjetiva era demostrar cómo la pérdida de paquetes en base a la tasa de error de bits (BER) de un sistema de transmisión inalámbrica MIMO afecta la calidad del video 3D de dos vistas en la percepción de un usuario real.

7. CONCLUSIONES

En la codificación de video 3D con múltiples vistas, siguiendo las recomendaciones del estandar HEVC con estructura de codificación GOP Allintra, se pueden perder paquetes de las diferentes vistas de textura; pero en la codificación de video 3D con DIBR y datos de profundidad, no se puede perder ningún paquete de las vistas de textura, pero sí de las vistas de profundidad. En caso que se pierda información de las vistas de textura, el video no se podra decodificar.

La codificación de video 3D con las recomendaciones del estandar HEVC con estructuras de codificación GOP de tipo IPBB y IPPP son muy susceptibles a las pérdidas de información, por lo que el protocolo de transmisión más recomendado para su transmisión es TCP. No se recomienda la implementación de éstos esquemas en transmisión en tiempo real.

El aumento en la diversidad en los esquemas de transmisión inalámbrica MIMO disminuye la pérdida en la transmisión aumentando la calidad en las mediciones objetivas de la transmisión y la percepción del usuario en la información transmitida.

La implemetación y estandarización del empaquetamiento de unidades NAL, en codificación y transmisión de video 3D con HEVC, aumenta y beneficia la transmisión y disminuye la pérdida de paquetes, debido a que por las recomendaciones del estandar HEVC, se sigue una clasificación de NALs con diferentes niveles de prioridad, de tal manera que se presentan de diferentes tamaños dependiendo de la sección del frame del video a la cual pertenezca y su nivel de importancia dentro del GOP de la codificación del video.

Las pérdidas de información en la transmisión de video 3D en esquemas inalámbricos afecta de forma más significativa una de las vistas, en comparación con la otras, de modo que si aprovechando la correlacion que hay entre las vistas codificadas, se puede recuperar la información de la vista con pérdida, al obtener y aprovechar los MV (Vectores de movimiento) creados durante la codificación.

La pérdida de información en lo videos transmitidos en formato YUV 4:2:0 se presentaron en mayor medida en los componentes de luminancia y poco en los componentes de crominancia, ya que la visión humana es más sensible a los componentes de luminancia.

Las pérdidas de paquetes en transmisiones inalámbricas con esquemas MIMO de video 3D codificados con la recomendación del estandar HEVC afecta más la calidad del video de alta movilidad que al video de baja movilidad.

8. RECOMENDACIONES Y TRABAJO FUTURO

- Se recomienda el diseño e implementación de una estructura de codificación GOP tipo IPPP más eficiente para prevenir la pérdida de paquetes en la implementación de la codificación en transmisiones inalámbricas en tiempo real de video 3D.
- Elaboración e implementación de pruebas subjetivas más elaboradas para videos 3D de más de 2 vistas de textura y DIBR con múltiples vistas autoestereoscópicas, implementado en un laboratorio y un ambiente más controlado.
- Se propone la evaluación de otro tipo de parámetros objetivos como lo son el UIQI (Índice de Calidad de Imagen Universal), junto al SSIM (Índice de Similitud Estructural) que miden si los cambios estructurales en la señal desenfocada y comprimida son relevantes.
- Por último se puede tener en cuenta la opción de incluir el sonido a la hora de evaluar la calidad de las secuencias audiovisuales. Existen estudios los cuales indican que la calidad de una secuencia tal como la perciben las personas se modifica por la presencia de una señal sonora.

9. BIBLIOGRAFÍA

Red Gráfica Latinoamérica, Fundamentos de la tecnología 3D. Disponible en: <http://redgrafica.com/Fundamentos-de-la-tecnologia-3D>.

S.Grewatsch and E.Muller, Sharing of Motion Vectors in 3D Video Coding, in Proceedings of IEEE International Conference on Image Processing (ICIP '04), vol. 5, pp. 3271-3274, Singapore, October 2004.

Christoph Fehn, "Depth-Image-Based Rendering (DIBR), Compression and transmission for a new Approach on 3D-TV", Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut (HHI) Einsteinufer 37, 10587 Berlin, Germany.

C.T.E.R. Hewage, S.T. Worrall, S. Dogan, A.M. Kondo, "Frame concealment algorithm for stereoscopic video using motion vector sharing," Multimedia and Expo, 2008 IEEE International Conference on, pp.485-488, June 23 2008-April 26, 2008.

Gary j. Sullivan, Fellow, Jens-Rainer OHM, Member, IEEE, Woo-jin Han, Member, IEEE, and Thomas Wiegand, Fellow, "Overview of the high Efficiency Video Coding (HEVC) Standard".

Sergio López, Memoria TFC - "Transmisión de vídeo 3D sobre redes P2P". Tesis de grado Ingeniería Técnica en Telecomunicaciones, especialidad en Telemática, (2012).

Javier Muñoz Asensio, "Comparativa de técnicas de precodificación ZF y MMSE en un sistema celular coordinado", Proyecto Fin de Carrera en Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Enero 2011. Pag 34.

González Fernández, S.; Compresión y transporte de televisión 3D sobre redes IP; Proyecto de fin de carrera - Universitat Politècnica de Catalunya. Escola Politècnica Superior de Castelldefels. Ingeniería técnica de Telecomunicación, especialidad Telemática, octubre 2010

Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG 2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994

Coding of Audio-Visual Objects - Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual version 1), ISO/IEC JTC 1, Apr. 1999 (and subsequent editions).

Advanced Video Coding for Generic Audio-Visual Services, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), ITU-T and ISO/IEC JTC 1, May 2003 (and subsequent editions).

B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, High Efficiency Video Coding (HEVC) Text Specification Draft 9, document JCTVC-K1003, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Oct 2012.

Grupalli, S.A., Cabrera, M.A., Bilbao, J.D., & Ferreyra, M.G. (2011). Sistemas de comunicación inalámbricos con tecnología MIMO. CET, (34), 7-13.

Alexis Alfredo Dowhuszko, “Teoría de la Transmisión de Información sobre Canales MIMO Inalámbricos con Múltiples Usuarios”, Tesis Doctoral en Ciencias de la Ingeniería, Universidad Nacional de Córdoba, Argentina, Marzo 2010.

Cortés-Peña, L.M. (2009). MIMO Space-Time Block Coding (STBC): Simulations and Results [Design Project: Personal and Mobile Communications, Georgia Tech (ECE6604)]. Recuperado de http://users.ece.gatech.edu/cortes/STBC_Matlab.html

Shannon, C. E. (1948), A Mathematical Theory of Communication. Bell System Technical Journal, 27: 379-423. doi: 10.1002/j.1538-7305.1948.tb01338.x

Los “Turbo Códigos”, nuevos subsistemas de codificación de canal para mejorar la transmisión digital de gran capacidad, Enrie Armengol Roig, J.A. Delgado-Penín, BURAN No.8 DICIEMBRE 1996 Pag 66, 67, 68, 69

J. HAGENAUER, P. HOEHER, A Viterbi Algorithm with Soft-Decision Outputs and its Applications, Proceedings of GLOBECOM’89, Dallas, Texas, noviembre 1989, pp. 47.1.1-47.1.7

Suarez. J, Valencia. O; Diseño de un modelo objetivo para evaluar la calidad de video 3D anaglifo; Tesis de grado Ingeniería Electrónica, 2016.

Kai Zeng, Abdul Rehman, Jiheng Wang, and Zhou Wang. “From H. 264 to HEVC: Coding gain predicted by objective video quality assessment models”. En: International Workshop on Video Processing and Quality Metrics for Consumer Electronics], (Feb 2013). 2013

ITU Union. “ITU-T Recommendation P.800.1: Mean Opinion Score (MOS) terminology”. En: International Telecommunication Union, Tech. Rep (2006).

ITU-R BT Recommendation. “500-11: Methodology for the Subjective Assessment of the Quality of Television Pictures. 500-11”. En: Standardization Sector of International Telecommunication Union, ITU Telecom (2002).

ITU Recommendation. “P.910: Subjective video quality assessment methods for multimedia applications”. En: Standardization Sector of International Telecommunication Union, Geneva (2008).

Alejandro León Álvarez and Jorge Mario Estrada. “Prueba piloto para determinar la reproducibilidad de la carta de Snellen al evaluar la agudeza visual en sujetos jóvenes”. En: Ciencia Tecnología para la Salud Visual y Ocular 7.1 (2009), pp. 13-24.

Documento jctvc-hm - Revision 4811 - recuperado de <http://hevc.kw.bbc.co.uk/svn/jctvc-hm/>

Documento [svn_HEVCSoftware](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/) - Revision 4811 - recuperado de https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services - Coding of moving video

RFC 7798 RTP para el estándar de codificación ITU-T de la recomendación e h.265
SO/IEC International Standard 23008-2

IT++ Documentation (2013, julio 6). Recuperado de
<http://itpp.sourceforge.net/4.3.1/>

Dorado, S., Santacruz, J. Pino, F. (2014). Propuesta de estructuras con tensores para
la implementación de esquemas MIMO. Revista ST, 12(29), 85-102

Turbo Codec Documentation (2013, julio 6). Recuperado de
http://itpp.sourceforge.net/4.3.1/classitpp_1_1Turbo_Codec.html

Obando. M, Freitas. W Jr. Cavalcanti. F; Switching Between Hybrid MIMO
Structures for Video Transmission Based on Distortion Model; Brazil (2010)

Borman, D; TCP Options and Maximum Segment Size (MSS); recuperado de
<https://tools.ietf.org/html/rfc6691>

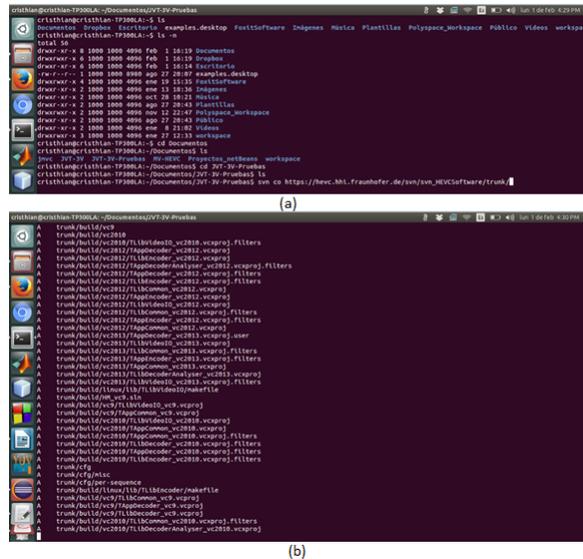
MVC Software Manual Version:JMVC 8.5 (CVS tag: JMVC₈₅)*Lastupdate* :
March26, 2011

BoxPlot. MathWorks documentation. Recuperado de:
<http://www.mathworks.com/help/stats/boxplot.html>

ANEXO A: CONFIGURACIÓN DE LAS RECOMENDACIONES DEL ESTÁNDAR HEVC

Los componentes que siguen éstas recomendaciones pueden ser descargadas de SVN Server^{50 51} El proceso de descarga de los componentes, usando el terminal del sistema operativo GNU/Linux con distribución Ubuntu se puede detallar en la Figura 33

Figura 33. Descarga de los componentes del esquema 3D-Extension del SVN Server



Una vez la versión de los componentes estén en nuestros directorios, se procede a hacer la compilación dependiendo del sistema operativo en que nos encontremos. La tabla 8 indica el directorio donde se encuentra el archivo de compilación según el entorno:

El ejemplo de la compilación de los componentes y las permisiones necesarias bajo el Sistema Operativo GNU/Linux con distribución Ubuntu se exponen en la Figura 34.

Para compilar correctamente la extensión del esquema 3D-Extension y no la versión para video de 2D, en el archivo *TypeDef.h* ubicado en el directorio *source\lib\TlibCommon* tiene que ser modificado el parámetro `#define HEVC_EXT` por el entero `2` para 3D-Extension, y en caso de que queramos la versión Multiview Extension (MV-HEVC), éste parámetro ha de cambiarse a `1`. Originalmente este parámetro se encuentra configurado en `0`, lo que significa que sólo codifica video 2D.

⁵⁰Documento jctvc-hm - Revision 4811 â En L: <http://hevc.kw.bbc.co.uk/svn/jctvc-hm/>

⁵¹Documento *svn_HEVCSoftware-Revision4811âEnLínea* :https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

Tabla 8. Directorios según manual de HM Software

Environment	Location of project file
MS Visual Studio 2008 (VC9)	build/HM _v c9.sln
MS Visual Studio 2010 (VC10)	build/HM _v c2010.sln
MS Visual Studio 2012 (VC11)	build/HM _v c2012.sln
MS Visual Studio 2013 (VC12)	build/HM _v c2013.sln
Xcode	HM.xcodeproj
Eclipse	.project .cproject
make/gcc(e.g. Linux)	build/linux/makefile

Fuente: Document JCTVC-Software Manual.

Figura 34. Compilación del esquema 3D-Extension

```

cristhian@cristhian-TP300LA: ~/Documentos/JVT-3V-Pruebas/trunk
cristhian@cristhian-TP300LA:~/Documentos/JVT-3V-Pruebas/trunk$ ls
build  cfg  compat  COPYING  doc  HM.xcodeproj  README  source
cristhian@cristhian-TP300LA:~/Documentos/JVT-3V-Pruebas/trunk$ ls -l
total 32
drwxrwxr-x 7 1000 1000 4096 Feb  1 16:30 build
drwxrwxr-x 4 1000 1000 4096 Feb  1 16:30 cfg
drwxrwxr-x 3 1000 1000 4096 Feb  1 16:30 compat
-rw-rw-r-- 1 1000 1000 1744 Feb  1 16:30 COPYING
drwxrwxr-x 2 1000 1000 4096 Feb  1 16:30 doc
drwxrwxr-x 2 1000 1000 4096 Feb  1 16:30 HM.xcodeproj
-rw-rw-r-- 1 1000 1000 927 Feb  1 16:30 README
drwxrwxr-x 4 1000 1000 4096 Feb  1 16:30 source
cristhian@cristhian-TP300LA:~/Documentos/JVT-3V-Pruebas/trunk$ chmod 777 -R build

```

(a)

```

cristhian@cristhian-TP300LA: ~/Documentos/JVT-3V-Pruebas/trunk/build/linux
cristhian@cristhian-TP300LA:~/Documentos/JVT-3V-Pruebas/trunk/build/linux$ ls -l
total 24
drwxrwxrwx 5 1000 1000 4096 Feb  1 16:30
drwxrwxrwx 2 1000 1000 4096 Feb  1 16:30
drwxrwxrwx 8 1000 1000 4096 Feb  1 16:30
-rwxrwxrwx 1 1000 1000 4785 Feb  1 16:30 makefile
drwxrwxrwx 4 1000 1000 4096 Feb  1 16:30
cristhian@cristhian-TP300LA:~/Documentos/JVT-3V-Pruebas/trunk/build/linux$ make

```

(b)

Proceso *build* (a) y *make* (b) para compilar el esquema 3D-Extension. Fuente: Autores.

ANEXO B: CÓDIGOS REFERENCIADOS DE PYTHON

El código presentado en este apartado corresponde al sistema de identificación y formulación de pérdidas de un stream binarios usando lenguaje de programación Python3. Basándose en éste programa se puede lograr la identificación de los stream binarios del video codificado en H.265 de modo que comienza ha identificar los paquetes o NALs que lo componen al identificar cada cabecera y el tipo de paquete entre los 63 tipos diferentes de NALs básicas que hay. También se adjunta el código en Python que después de identificar los paquetes toma el valor de la Tasa de Error de Bits (BER) de la simulación de la transmisión inalámbrica y procede ha sobrescribir determinado valor de NALs escogidas aleatoriamente, con el objetivo de emular la pérdida de paquetes en transmisiones inalámbricas de video 3D. Éstos códigos han sido diseñados e implementados usando como base trabajos de las recomendaciones de empaquetamiento y codificación del estándar de codificación HEVC

• Código para la formulación de pérdidas

```
'''
Created on 27/01/2016
@author: cristian
'''
class Identificacion():
    def __init__(stream, nombre_arch_txt):
        #####
    def binarizar(decimal):
        binario = ''
        while decimal // 2 != 0:
            binario = str(decimal % 2) + binario
            decimal = decimal // 2
        return str(decimal) + binario
        #####
    def concatenarLista(lista_a, lista_b):
        lista_nueva = []
        for d in range(0, len(lista_a)):
            lista_nueva.append(lista_a[d])
        for d in range(0, len(lista_b)):
            lista_nueva.append(lista_b[d])
        return lista_nueva
        #####
# IDENTIFICA LA LAYER A LA QUE PERTENECE LA NAL
def Identifica_Layer(LayerId_Deci, Layer1, Layer2,
                    Layer3, Layer4, Layer5, Layer6):
    if LayerId_Deci == 0:
        Layer1 += 1
        LAYER = str("LAYER_1")
    elif LayerId_Deci == 1:
        Layer2 += 1
        LAYER = str("LAYER_2")
    elif LayerId_Deci == 2:
        Layer3 += 1
        LAYER = str("LAYER_3")
    elif LayerId_Deci == 3:
        Layer4 += 1
        LAYER = str("LAYER_4")
    elif LayerId_Deci == 4:
        Layer5 += 1
        LAYER = str("LAYER_5")
    else:
```

```

        Layer6 += 1
        LAYER = str("LAYER_6")
    return (LAYER, Layer1, Layer2, Layer3,
            Layer4, Layer5, Layer6)
#####
def bytes_from_file(stream):
    with open(stream, "rb") as f:
        while True:
            byte = f.read(1)
            if not byte:
                break
            yield (ord(byte))
    return
B = []
for b in bytes_from_file(stream):
    B.append(b)
NAL_type_I = 0
NAL_type_P = 0
NAL_type_B = 0
# CLASES DE NALS
TRAIL_N = 0 # 0
TRAIL_R = 0 # 1
TSA_N = 0 # 2
TSA_R = 0 # 3
STSA_N = 0 # 4
STSA_R = 0 # 5
RADL_N = 0 # 6
RADL_R = 0 # 7
RASL_N = 0 # 8
RASL_R = 0 # 9
RSV_VCL_N10 = 0 # 10
RSV_VCL_N12 = 0 # 12
RSV_VCL_N14 = 0 # 14
RSV_VCL_R11 = 0 # 11
RSV_VCL_R13 = 0 # 13
RSV_VCL_R15 = 0 # 15
BLA_W_LP = 0 # 16
BLA_W_RADL = 0 # 17
BLA_N_LP = 0 # 18
IDR_W_RADL = 0 # 19
IDR_N_LP = 0 # 20
CRA_NUT = 0 # 21
RSV_IRAP_VCL22 = 0 # 22
RSV_IRAP_VCL23 = 0 # 23
RSV_VCL24__RSV_VCL31 = 0 # 24,25,26,27,
                        #28,29,30,31
VPS_NUT = 0 # 32
SPS_NUT = 0 # 33
PPS_NUT = 0 # 34
AUD_NUT = 0 # 35
EOS_NUT = 0 # 36
EOB_NUT = 0 # 37
FD_NUT = 0 # 38
PREFIX_SEI_NUT = 0 # 39
SUFFIX_SEI_NUT = 0 # 40
N_nal = 0 # Es el numero de NAL en el stream
HEADER = [] # Vector que almacena los 2 bytes de cabecera
Ubic_NAL = []

```

```

Ubic_NAL_HEX = []
h = len(B)

Type_array = []
LayerId_array = []

Type_Decimal_array = [] # Vector almacena tipos de NAL
LayerId_Decimal_array = [] # Vector almacena layers
#en la que aparece la NAL

Tipo_NAL = []
Tipo_LAYER = []
# CAPAS DE LA NAL
Layer1 = 0
Layer2 = 0
Layer3 = 0
Layer4 = 0
Layer5 = 0
Layer6 = 0

for k in range(h):
if B[k] == 0 and B[k + 1] == 0 and B[k + 2] == 0
and B[k + 3] == 1:
N_nal += 1
p = k + 4
H = hex(p)
Ubic_NAL_HEX.append(H) # Numero de NAL
Ubic_NAL.append(p) # Ubicacion del inicio cada NAL
Valor_Byte_Ini_en_Binario = binarizar(
B[k + 4])
Valor_Byte_Ini_en_Binario_2 = binarizar(B[k + 5])
Valor_Byte_Ini_en_Binario_L = list(
Valor_Byte_Ini_en_Binario) # convierte el byte de inicio
# de cada NAL a binario en un array
Valor_Byte_Ini_en_Binario_L_2 =
list(Valor_Byte_Ini_en_Binario_2)
longitud_Valor_Byte_Ini_en_Binario
= len(Valor_Byte_Ini_en_Binario_L)
longitud_Valor_Byte_Ini_en_Binario_L_2 =
len(Valor_Byte_Ini_en_Binario_L_2)

while longitud_Valor_Byte_Ini_en_Binario_L_2 < 8:
Valor_Byte_Ini_en_Binario_L_2 = concatenarLista
([0], Valor_Byte_Ini_en_Binario_L_2)
longitud_Valor_Byte_Ini_en_Binario_L_2 =
len(Valor_Byte_Ini_en_Binario_L_2)

while longitud_Valor_Byte_Ini_en_Binario < 8:
Valor_Byte_Ini_en_Binario_L = concatenarLista
([0], Valor_Byte_Ini_en_Binario_L)
longitud_Valor_Byte_Ini_en_Binario
= len(Valor_Byte_Ini_en_Binario_L)

header = concatenarLista(Valor_Byte_Ini_en_Binario_L ,
Valor_Byte_Ini_en_Binario_L_2)
HEADER.append(header)
Type = [str(header[1]), str(header[2]), str(header[3]),
str(header[4]), str(header[5]), str(header[6])],
LayerId = [str(header[7]), str(header[8]), str(header[9]),
str(header[10]), str(header[11]), str(header[12])]
Type_string = ''.join(Type)
LayerId_string = ''.join(LayerId)
Type_array.append(Type_string)

```

```

LayerId_array.append(LayerId_string)
Type_Decimal = int(str(Type_string), 2)
Type_Decimal_array.append(Type_Decimal)
LayerId_Decimal = int(str(LayerId_string), 2)
LayerId_Decimal_array.append(LayerId_Decimal)
[LAYER, Layer1, Layer2, Layer3, Layer4, Layer5, Layer6]
= Identifica_Layer(LayerId_Decimal, Layer1, Layer2,
    Layer3, Layer4, Layer5, Layer6)
    Tipo_LAYER.append(LAYER)
    if Type_Decimal == 0:
        TRAIL_N += 1
        Tipo_NAL.append("TRAIL_N")
        NAL_type_B += 1
    elif Type_Decimal == 1:
        TRAIL_R += 1 # 1
        Tipo_NAL.append("TRAIL_R")
        NAL_type_B += 1
    elif Type_Decimal == 2:
        TSA_N += 1 # 2
        Tipo_NAL.append("TTSA_N")
        NAL_type_B += 1
    elif Type_Decimal == 3:
        TSA_R += 1 # 3
        Tipo_NAL.append("TSA_R")
        NAL_type_B += 1
    elif Type_Decimal == 4:
        STSA_N += 1 # 4
        Tipo_NAL.append("STSA_N ")
    elif Type_Decimal == 5:
        STSA_R += 1 # 5
        Tipo_NAL.append("STSA_R")
    elif Type_Decimal == 6:
        RADL_N += 1 # 6
        Tipo_NAL.append("RADL_N")
        NAL_type_P += 1
    elif Type_Decimal == 7:
        RADL_R += 1 # 7
        Tipo_NAL.append("RADL_R")
        NAL_type_P += 1
    elif Type_Decimal == 8:
        RASL_N += 1 # 8
        Tipo_NAL.append(" RASL_N")
        NAL_type_P += 1
    elif Type_Decimal == 9:
        RASL_R += 1 # 9
        Tipo_NAL.append("RASL_R")
        NAL_type_P += 1
    elif Type_Decimal == 10:
        RSV_VCL_N10 += 1 # 10
        Tipo_NAL.append(" RSV_VCL_N10")
    elif Type_Decimal == 12:
        RSV_VCL_N12 += 1 # 12
        Tipo_NAL.append("RSV_VCL_N12")
    elif Type_Decimal == 14:
        RSV_VCL_N14 += 1 # 14
        Tipo_NAL.append("RSV_VCL_N14 ")
    elif Type_Decimal == 11:
        RSV_VCL_R11 = 0 # 11

```

```

        Tipo_NAL.append("RSV_VCL_R11")
elif Type_Decimal == 13:
    RSV_VCL_R13 = 0 # 13
    Tipo_NAL.append("SV_VCL_R13")
elif Type_Decimal == 15:
    RSV_VCL_R15 = 0 # 15
    Tipo_NAL.append("RSV_VCL_R15")
elif Type_Decimal == 16:
    BLA_W_LP += 1 # 16
    Tipo_NAL.append("BLA_W_LP")
    NAL_type_I += 1
elif Type_Decimal == 17:
    BLA_W_RADL += 1 # 17
    Tipo_NAL.append("BLA_W_RADL")
    NAL_type_I += 1
elif Type_Decimal == 18:
    BLA_N_LP += 1 # 18
    Tipo_NAL.append("BLA_N_LP")
    NAL_type_I += 1
elif Type_Decimal == 19:
    IDR_W_RADL += 1 # 19
    Tipo_NAL.append("IDR_W_RADL")
    NAL_type_I += 1
elif Type_Decimal == 20:
    IDR_N_LP += 1 # 20
    Tipo_NAL.append("IDR_N_LP")
    NAL_type_I += 1
elif Type_Decimal == 21:
    CRA_NUT += 1 # 21
    Tipo_NAL.append("CRA_NUT")
    NAL_type_I += 1
elif Type_Decimal == 22:
    RSV_IRAP_VCL22 += 1 # 22
    Tipo_NAL.append("RSV_IRAP_VCL22")
    NAL_type_I += 1
elif Type_Decimal == 23:
    RSV_IRAP_VCL23 += 1 # 23
    Tipo_NAL.append("BRSV_IRAP_VCL23")
    NAL_type_I += 1
elif Type_Decimal == 24
or Type_Decimal == 25 or Type_Decimal == 26
or Type_Decimal == 27 or Type_Decimal == 28
or Type_Decimal == 29 or Type_Decimal == 30
or Type_Decimal == 31:
    RSV_VCL24__RSV_VCL31 += 1 # 24,25,26,27,28,29,30,31
    Tipo_NAL.append("RSV_VCL24__RSV_VCL31")
elif Type_Decimal == 32:
    VPS_NUT += 1 # 32
    Tipo_NAL.append("VPS")
elif Type_Decimal == 33:
    SPS_NUT += 1 # 33
    Tipo_NAL.append("SPS")
elif Type_Decimal == 34:
    PPS_NUT += 1 # 34
    Tipo_NAL.append("PPS")
elif Type_Decimal == 35:
    AUD_NUT += 1 # 35
    Tipo_NAL.append("AUD_NUT")
elif Type_Decimal == 36:
    EOS_NUT += 1 # 36

```

```

        Tipo_NAL.append("EOS_NUT")
elif Type_Decimal == 37:
    EOB_NUT += 1 # 37
    Tipo_NAL.append("EOB_NUT")
elif Type_Decimal == 38:
    FD_NUT += 1 # 38
    Tipo_NAL.append("FD_NUT")
elif Type_Decimal == 39:
    PREFIX_SEI_NUT += 1 # 39
    Tipo_NAL.append("PREFIX_SEI_NUT")
else:
    SUFFIX_SEI_NUT += 1 # 40
    Tipo_NAL.append("PREFIX_SEI_NUT")
file = open(nombre_arch_txt, 'w')
file.write("No. Nals es \t %s \n" % (int(N_nal)))
file.write("No. Nals I es \t %s \n" % (int(NAL_type_I)))
file.write("No. Nals P es \t %s \n" % (int(NAL_type_P)))
file.write("No. Nals B es \t %s \n" % (int(NAL_type_B)))

file.write("El Numero de NAL es la Layer 1 es \t %s \n"
% (int(Layer1)))
file.write("El Numero de NAL es la Layer 2 es \t %s \n"
% (int(Layer2)))
file.write("El Numero de NAL es la Layer 3 es \t %s \n"
% (int(Layer3)))
file.write("El Numero de NAL es la Layer 4 es \t %s \n"
% (int(Layer4)))
file.write("El Numero de NAL es la Layer 5 es \t %s \n"
% (int(Layer5)))
file.write("El Numero de NAL es la Layer 6 es \t %s \n"
% (int(Layer6)))

file.write("Numero de NAL Tipo TRAIL_N \t %s \n"
% (int(TRAIL_N)) # 0)
file.write("Numero de NAL Tipo TRAIL_R \t %s \n"
% (int(TRAIL_R)) # 1)
file.write("Numero de NAL Tipo TSA_N \t %s \n"
% (int(TSA_N)) # 2)
file.write("Numero de NAL Tipo TSA_R \t %s \n"
% (int(TSA_R)) # 3)
file.write("Numero de NAL Tipo STSA_N \t %s \n"
% (int(STSA_N)) # 4)
file.write("Numero de NAL Tipo STSA_R \t %s \n"
% (int(STSA_R)) # 5)
file.write("Numero de NAL Tipo RADL_N \t %s \n"
% (int(RADL_N)) # 6)
file.write("Numero de NAL Tipo RADL_R \t %s \n"
% (int(RADL_R)) # 7)
file.write("Numero de NAL Tipo RASL_N \t %s \n"
% (int(RASL_N)) # 8)
file.write("Numero de NAL Tipo RASL_R \t %s \n"
% (int(RASL_R)) # 9)
file.write("Numero de NAL Tipo RSV_VCL_N10 \t %s \n"
% (int(RSV_VCL_N10)) # 10)
file.write("Numero de NAL Tipo RSV_VCL_N12 \t %s \n"
% (int(RSV_VCL_N12)) # 12)
file.write("Numero de NAL Tipo RSV_VCL_N14 \t %s \n"

```

```

% (int(RSV_VCL_N14))) # 14
file.write("Numero de NAL Tipo RSV_VCL_N11 \t %s \n")
% (int(RSV_VCL_R11))) # 11
file.write("Numero de NAL Tipo RSV_VCL_N13 \t %s \n")
% (int(RSV_VCL_R13))) # 13
file.write("Numero de NAL Tipo RSV_VCL_N15 \t %s \n")
% (int(RSV_VCL_R15))) # 15
file.write("Numero de NAL Tipo BLA_W_LP \t %s \n")
% (int(BLA_W_LP))) # 16
file.write("Numero de NAL Tipo BLA_W_RADL \t %s \n")
% (int(BLA_W_RADL))) # 17
file.write("Numero de NAL Tipo BLA_N_LP \t %s \n")
% (int(BLA_N_LP))) # 18
file.write("Numero de NAL Tipo IDR_W_RADL \t %s \n")
% (int(IDR_W_RADL))) # 19
file.write("Numero de NAL Tipo IDR_N_LP \t %s \n")
% (int(IDR_N_LP))) # 20
file.write("Numero de NAL Tipo CRA_NUT \t %s \n")
% (int(CRA_NUT))) # 21
file.write("Numero de NAL Tipo CSV_IRAP_VCL22 \t %s \n")
% (int(RSV_IRAP_VCL22))) # 22
file.write("Numero de NAL Tipo CSV_IRAP_VCL23 \t %s \n")
% (int(RSV_IRAP_VCL23))) # 23
file.write("Numero de NAL Tipo RSV_VCL24__RSV_VCL31 \t %s \n")
% (int(RSV_VCL24__RSV_VCL31))) # 24,25,26,27,28,29,30,31
file.write("Numero de NAL Tipo VPS \t %s \n")
% (int(VPS_NUT))) # 32
file.write("Numero de NAL Tipo SPS \t %s \n")
% (int(SPS_NUT))) # 33
file.write("Numero de NAL Tipo PPS \t %s \n")
% (int(PPS_NUT))) # 34
file.write("Numero de NAL Tipo AUD_NUT \t %s \n")
% (int(AUD_NUT))) # 35
file.write("Numero de NAL Tipo EOS_NUT \t %s \n")
% (int(EOS_NUT))) # 36
file.write("Numero de NAL Tipo EOB_NUT \t %s \n")
% (int(EOB_NUT))) # 37
file.write("Numero de NAL Tipo FD_NUT \t %s \n")
% (int(FD_NUT))) # 38
file.write("Numero de NAL Tipo PREFIX_SEI_NUT \t %s \n")
% (int(PREFIX_SEI_NUT))) # 39
file.write("Numero de NAL Tipo SUFFIX_SEI_NUT \t %s \n \n \n \n")
% (int(SUFFIX_SEI_NUT))) # 40

file.write("IDENTIFICACION DE CADA NAL\n \n")
file.write("NAL/UBICACION/UBICA_HEX/TIPO/LAYER/TAMANO\n")

i = 0
suma = 0
Tama_NAL = [] #array almacenar tamaño en Byte
#de cada NAL

while i < N_nal:
    if N_nal - 1 == i:
        T = h - Ubic_NAL[i]
    else:
        T = Ubic_NAL[i + 1] - Ubic_NAL[i]

```

```

        Tama_NAL.append(T)
file.write("/ %s / %s / %s / %s / %s / %s \n " % (
    i, Ubic_NAL[i], Ubic_NAL_HEX[i],
        Tipo_NAL[i], Tipo_LAYER[i], Tama_NAL[i]))
    i += 1
    suma = T + suma
Media = suma / N_nal
file.close()
return (Media, N_nal, Ubic_NAL, Ubic_NAL_HEX,
        LayerId_Decimal_array, Type_Decimal_array)

```

- Código para la emulación de pérdidas de paquetes en transmisiones inalámbricas
-

```

from Identificacion import Identificacion
import random
import math
class Perdidas():
def perdida_aleatoria(stream_Modu, N_nal,
catidad_packet_perdidad, Ubic_NAL, Ubic_NAL_HEX,
nombre_arch_txt, LayerId_Decimal_array,
Type_Decimal_array):
def listaAleatorios(catidad_Nal_perdidad, N_nal):
    lista = [0] * catidad_Nal_perdidad
    for i in range(catidad_Nal_perdidad):
        lista[i] = math.floor( N_nal * (random.random()))
    return lista
    Packet_a_perder = []
    Packet_realmente_perdidos = len(Packet_a_perder)
while Packet_realmente_perdidos < catidad_packet_perdidad:
    Nals_perdidad = listaAleatorios((catidad_packet_perdidad
- Packet_realmente_perdidos), N_nal)
# llama la funcion de una lista de NAL aleatorias
# a las cuales eliminar
        U = 0
        while U < catidad_packet_perdidad:
            T = Nals_perdidad[U]
if ((T != 0) or (T != 1) or (T != 2)
or (T != 3) or (T != 4) or (T != 5))
and ((Type_Decimal_array[T] != 32)
or (Type_Decimal_array[T] != 33)
or (Type_Decimal_array[T] != 34)):
            Packet_a_perder.append(T)
            U += 1
        else:
            U += 1
    Packet_realmente_perdidos = len(Packet_a_perder)
    s = 0
    Packet_realmente_perdidos = len(Packet_a_perder)
while s < Packet_realmente_perdidos:

```

```

Q = Packet_a_perder[s]
ini = Ubic_NAL[Q] - 1
fin = Ubic_NAL[Q + 1] - 2
D = ini
with open('cero.bit', 'rb') as fh:
    prueba = fh.read()
while D >= ini and D < fin:
    with open(stream_Modi, 'r+b') as fh:
        fh.seek(D)
        fh.write(prueba)
        D = D + 0x1
s += 1

file = open(nombre_arch_txt, 'w')
file.write("\n\n\n PERDIDAS \n \n \n \n" )
file.write('Nal que se perdio \t %s \n'
% (Packet_a_perder))
file.write("Cantidad de Nal que se perdieron \t %s \n"
% (int(catidad_packet_perdidad)))
file.write("Cantidad de Nal SI se perdieron \t %s \n"
% (int(Packet_realmente_perdidos)))
for v in Packet_a_perder:
file.write('ubicacion de las Nal que se perdieron \t %s \n'
% (Ubic_NAL_HEX[v]))
file.close()

return (stream_Modi)

```

ANEXO C: CÓDIGOS REFERENCIADOS DE C++

– El código presentado en este apartado corresponde al sistema de transmisión MIMO 2x2 simulado usando las librerías de comunicaciones de IT++. Basándose en éste programa se puede lograr la implementación de los sistemas de transmisión MIMO 4x4 y 8x8. El objetivo de esta simulación es obtener la Tasa de Error de Bits (BER) de la transmisión inalámbrica a diferentes valores de Relación Señal/Ruido (SNR). Éste valor de BER es necesario para analizar la pérdida de paquetes en un caso real de transmisión de video 3D. Éste código ha sido diseñado e implementado usando como base trabajos de simulación presentados del grupo de Investigación UNITCOM del programa de Ingeniería Electrónica de la Universidad Surcolombiana.

• Código completo del sistema de transmisión MIMO 2X2

```
// *----- Librerias incluidas para el desarrollo del proyecto -----*
#include <iostream>
#include <string>
#include <itpp/itcomm.h>
#include <itpp/itbase.h>
#include <stdlib.h>
#include <itpp/stat/misc_stat.h>
using std::cout;
using std::endl;
using std::string;
using namespace itpp;
{
/*----- Declaracion de clases y variables -----*/
//..... VIDEO.....
    bvec bVideo, bVideo_acon; //Variables en binario
//..... TURBO CODEC.....
// Clases
    Turbo_Codec turbo;
    BPSK bpsk;
//Variables
    ivec gen(2);
    gen(0) = 013; gen(1) = 015; // Polinomios generadores
    int constraint_length = 4; // Longitud de restriccion
    int block_length = pow2i(6); // Longitud del entrelazador
    ivec interleaver_sequence = lte_turbo_interleaver_sequence(block_length);
    // Entrelazador turbo interno
    int iterations = 10; // Numero de iteraciones de decodificacion
    string metric = "LOGMAX"; //Metrica del decodificador
    double logmax_scale_factor = 0.7;
    bool adaptive_stop = true;
    turbo.set_parameters(gen, gen, constraint_length, interleaver_sequence,
        iterations, metric, logmax_scale_factor,
        adaptive_stop);
    // Ajusta los parametros para el codificador/decodificador turbo
    vec bpsk_symbols, received;
    bvec input, coded_bits, decoded_bits, transmitted, coded_bits_acon;
    ivec nb_used_it;
//..... MODULACION/DEMULACION.....
//Variables
    cvec qam_symbols;
    bvec rx_bits, rx_bits_acon;
    int M = 16; //Orden de la constelacion (M = 16,64,256)
    int k = log2(M); //Bits por símbolo
//Clases
    QAM qam(M); //Clase modulador QAM
//Variables
    int qamLength;
    cmat H, R, Hh;
    cmat Hht, Hpl;
    cmat recR0R1, recR0R1_trans;
//..... CANAL.....
//Clases
    AWGN_Channel awgn_channel(0.1); //Clase de canal AWGN
```

```

BERC          berc;
//Variables
cvec          signal_rec;
cvec          r0, r1, r2, r3;
cvec          r0n, r1n, r2n, r3n;
cvec          recib;
cvec          s0, s1;
cvec          h0, h1, h2, h3;
cvec          h0neg, h2neg;
vec           SNRdB, BER;
vec           EbN0dB, EbN0, N0;
double       Ec, Eb;

/*-----Parametros de simulacion-----*/
//Parametros del ruido AWGN
Ec = 1.0;          //La energia transmitida por simbolo QAM es 1
Eb = Ec / k;       //La energia a transmitida por bit es Ec/tamano de palabra
SNRdB = "-10:2:10";
EbN0dB = SNRdB - 10*log10(k); //Se convierte el parametro SNR a N0, para poder
EbN0 = inv_dB(EbN0dB); // utilizar la clase AWGN predefinida en la libreria
N0 = Eb * pow(EbN0, -1.0);

bVideo = randb(pow2i(14)); //Crear mapa de bits a transmitir
cout<<"\nTamano_de_bVideo_="<<bVideo.length()<<"_bits_\n";

turbo.encode(bVideo, coded_bits); //Proceso del TurboCodec
// cout<<"\ncoded_bits = "<<coded_bits<<"\n";
cout<<"\nTamano_de_coded_bits_="<<coded_bits.length()<<"\n";

//.....Acondicionamiento de la senal de entrada codificada.....
int origLength;
int adjLength;
origLength = coded_bits.length();
adjLength = 256 - (origLength%256); //1536 = (4(6x64))
bvec acon(adjLength);
acon.zeros();
coded_bits_acon = concat(coded_bits, acon);
cout<<"\nTamano_de_coded_bits_acondicionado="<<coded_bits_acon.length()<<"_bits_\n";

/*-----MODULACION-----*/
qam.modulate_bits(coded_bits_acon, qam_symbols); //Modula los bits en simbolos QAM
cout<<"\nTamano_de_qam_symbols_="<<qam_symbols.length()<<"_simbolos_\n";

/*-----COD ALAMOUTI-----*/
//Dividimos el vector de simbolos en s0 y s1, un ejemplo a continuacion:
//qamSymbols = [a b c d e f];
//s0 = [a c e]
//s1 = [b d f]
qamLength = qam_symbols.length();
s0 = zeros_c(1, qamLength/2);
s1 = zeros_c(1, qamLength/2);
int z = 0;
for (int j=0; j < qamLength; j = j+2)
{
    if (j == 0)
    {
        s0(j) = qam_symbols(j);
        s1(j) = qam_symbols(j+1);
    }
    else
    {
        s0(z) = qam_symbols(j);
        s1(z) = qam_symbols(j+1);
    }
    z++;
}

/*-----CANAL-----*/
//Caracterizacion del canal h0 y h1, canales Rayleigh
h0 = sqrt(0.5) * (randn_c(1, qamLength/2));
h1 = sqrt(0.5) * (randn_c(1, qamLength/2));
h2 = sqrt(0.5) * (randn_c(1, qamLength/2));
h3 = sqrt(0.5) * (randn_c(1, qamLength/2));
//Senal recibida sin ruido
r0 = elem_mult(h0, s0) + elem_mult(h1, s1);
//r0 = h0*s0 + h1*s1
h0neg = -1*h0;
r1 = elem_mult(h0neg, conj(s1)) + elem_mult(h1, conj(s0));
//r1 = -h0*conj(s1) + h1*conj(s0)
r2 = elem_mult(h2, s0) + elem_mult(h3, s1);
//r2 = h2*s0 + h3*s1

```

```

h2neg = -1*h2;
r3 = elem_mult(h2neg, conj(s1)) + elem_mult(h3, conj(s0));
    //r3 = -h2*conj(s1) + h3*conj(s0)

BER = zeros(1,SNRdB.length());
    //BER es un vector para guardar los resultados de BER totales

//Ciclo de codificacion de Alamouti con ruido, decodificacion de Alamouti,
//demodulacion, decodificacion turbo y calculo de BER

mat COR = zeros(1,SNRdB.length());
mat ERR = zeros(1,SNRdB.length());

int num_it =20000;
vec ber_array(num_it);

for (int i=0; i < SNRdB.length(); i++)
{
    cout << "\nSNRdB:_" << SNRdB(i) << "\n";

    for (int j=0; j < num_it; j++)
    {
        //Senal recibida con ruido
        awgn_channel.set_noise(N0(i));
        r0n = awgn_channel(r0); //r0 = h0*s0 + h1*s1 + n0
        r1n = awgn_channel(r1); //r1 = -h0*conj(s1) + h1*conj(s0) + n1
        r2n = awgn_channel(r2); //r2 = h2*s0 + h3*s1 + n0
        r3n = awgn_channel(r3); //r3 = -h2*conj(s1) + h3*conj(s0) + n1

        H = zeros_c(4,2); //Vector vacio para guardar la matriz de canal
        R = zeros_c(4,1); //Vector vacio para guardar la matriz de senal recibida
        recib = zeros_c(1,(r0.length()+r1.length()));
        //reciv es un vector para almacenar los simbolos recibidos

/*-----DECOD ALAMOUTI-----*/
        int y = 0;
        for (int k = 0; k < h0.length(); k++)
        {
            //Formamos la matriz de canal H = [h0 h1;h1* -h0*;h2 h3;h3* -h2*]
            H(0,0) = h0(k);
            H(0,1) = h1(k);
            H(1,0) = conj(h1(k));
            H(1,1) = -1*conj(h0(k));
            H(2,0) = h2(k);
            H(2,1) = h3(k);
            H(3,0) = conj(h3(k));
            H(3,1) = -1*conj(h2(k));

            //Calculamos la matriz pseudoinversa
            Hh = conj(H);
            Hht = Hh.transpose();
            Hpi = inv(Hht*H)*Hht;
            //Matriz pseudo inversa, equivalente a H = pinv(H) en MATLAB

            R(0,0) = r0n(k);
            R(1,0) = conj(r1n(k));
            R(2,0) = r2n(k);
            R(3,0) = conj(r3n(k));
            //Vector de senal recibida R = [r0; conj(r1);r2; conj(r3)]

            recR0R1 = Hpi*R; //Decodificacion de simbolos
            recR0R1_trans = recR0R1.transpose();

            //Simbolos decodificados guardados en la variable recib
            recib(y) = recR0R1_trans(0,0);
            recib(y+1) = recR0R1_trans(0,1);

            y = y + 2;
        }

/*-----DEMODULACION-----*/
        qam.demodulate_bits(recib, rx_bits_acon); //Demodula los simbolos QAM
        rx_bits = rx_bits_acon(0,origLength-1);
        //Quitamos los valores agregados por conveniencia

/*-----DECODIFICADOR TURBO-----*/
        bpsk.modulate_bits(rx_bits, bpsk_symbols);
        turbo.set_metric("LOGMAX",1.0);
        turbo.decode(to_vec(bpsk_symbols),decoded_bits, nb_used_it);

        //Calculo de BER
        berc.clear();
        berc.count(bVideo, decoded_bits);
        ber_array(j) = berc.get_errorate();
    }
}

```

```

cout<<" iteracion " <<j<<" , ber_array " <<ber_array(j)<<endl;
}
BER(i) = mean(ber_array);
cout << "\nBER: " << BER(i) <<" \n";
}
cout << "\nSNRdB: " << SNRdB <<" \n";
cout << "\nBER: " << BER <<" \n";
return 0;
}

```

ANEXO D: FORMATO CALIFICACIÓN PRUEBA SUBJETIVA

En el momento de que los espectadores entraran al espacio adecuado para la realización de la prueba subjetiva, se les entregó el formato que aparece a continuación. Un formato en físico fácil de manejar y utilizar de acuerdo a las recomendaciones internacionales para la realización del método DSIS.

EVALUACION SUBJETIVA DE LA CALIDAD DE VIDEO 3D				
Nombre	_____			
Edad	_____	Fecha	_____	
Ocupación	_____			
<p>Bienvenidos. Esta prueba se realiza para dar cumplimiento con uno de los objetivos planteados para la realización del proyecto de grado "EVALUACIÓN DE LA CALIDAD DE VIDEO 3D EN TRANSMISIONES INALÁMBRICAS CON EL NUEVO ESTÁNDAR HEVC".</p> <p>Por favor calificar las secuencias de video de acuerdo a su criterio según la explicación dada en la introducción a este examen. Esta prueba está para un estimado de 15 a 20 minutos</p>				
Secuencia No. 1				
1	2	3	4	5
Secuencia No. 2				
1	2	3	4	5
Secuencia No. 3				
1	2	3	4	5
Secuencia No. 4				
1	2	3	4	5
Secuencia No. 5				
1	2	3	4	5
Agradecemos su valioso tiempo.				