

# **TRABAJO ESPECIAL DE GRADO**

## **ESTUDIO DE LAS PROPIEDADES Y DESEMPEÑO DEL CODIFICADOR DE VIDEO DE LICENCIA LIBRE Y GRATUITA THEORA 1.0 Y COMPARACION CON EL CODEC H.264 EN LA ADAPTACION DE CONTENIDO MULTIMEDIA A SER TRANSMITIDO BAJO REDES IP**

**Presentado ante la ilustre  
Universidad Central de Venezuela  
Por el Br. Albornoz C., Manuel A.  
Para optar al título de  
Ingeniero Electricista**

**Caracas, 2010**

# **TRABAJO ESPECIAL DE GRADO**

## **ESTUDIO DE LAS PROPIEDADES Y DESEMPEÑO DEL CODIFICADOR DE VIDEO DE LICENCIA LIBRE Y GRATUITA THEORA 1.0 Y COMPARACION CON EL CODEC H.264 EN LA ADAPTACION DE CONTENIDO MULTIMEDIA A SER TRANSMITIDO BAJO REDES IP**

**Tutor Académico: Prof. Luis Fernández**

**Presentado ante la ilustre  
Universidad Central de Venezuela  
Por el Br. Albornoz C., Manuel A.  
Para optar al título de  
Ingeniero Electricista**

**Caracas, 2010**

## CONSTANCIA DE APROBACIÓN

Caracas, 18 de febrero de 2010

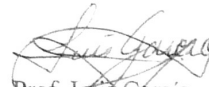
Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Ingeniería Eléctrica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller Manuel A. Albornoz C., titulado:

**“PROPIEDADES Y DESEMPEÑO DEL CODIFICADOR DE VIDEO DE LICENCIA LIBRE Y GRATUITA THEORA 1.0 Y COMPARACIÓN CON EL CODEC H.264 EN LA ADAPTACIÓN DE CONTENIDO MULTIMEDIA A SER TRANSMITIDO BAJO REDES IP”**


Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Electricista en la mención de Comunicaciones, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.



Prof. Zeldivar Bruzual  
Jurado



Prof. Luis García  
Jurado



Prof. Luis Fernández  
Tutor Académico

## **DEDICATORIA**

A mi madre, por su incondicional y constante apoyo durante todos estos años.

## RECONOCIMIENTOS Y AGRADECIMIENTOS

A mis amigos, que compartieron conmigo estos años de penurias y alegrías: Dennys, Sergio, Tulio, Alfonso, Juan Alejandro, Nicolas, Jhon, Edgardo, Maximiliano, Rafael, Gustavo.

A Raul Arreaza, Milla de la Roca, Rashmir Orellana, Carlos Fuenmayor, Rafael Arruebarrena, Vicente Lopez, Luis Prato; quienes, entre otros, me apoyaron en el transcurso de la carrera y me mostraron que la cualidad principal de un docente no es otra que la honestidad.

A María Auxiliadora Rojas, por ser pieza clave e indispensable en el departamento de Comunicaciones, por su apoyo desinteresado y su gran mística de trabajo.

A las autoridades de la Escuela de Ingeniería Eléctrica, por haber asumido como propios los valores y principios Bolivarianos y Revolucionarios de la V República, por convertirse en ejemplo vivo del empleado público Socialista, por emular de manera intachable las conductas, la forma de hacer política, de manejar las Instituciones, de elegir el personal calificado, de escuchar y atender la crítica, de realimentarse y evolucionar, en fin, por ser fieles representantes de la moral Socialista de su Comandante Presidente Hugo Chavez; a todos ellos, y en especial al Profesor Franklin Martínez, gracias por mostrarme un escenario de lo que no deseo para mi Universidad, para mi País ni para mi vida.

**Albornoz C., Manuel A.**

ESTUDIO DE LAS PROPIEDADES Y DESEMPEÑO DEL CODIFICADOR  
DE VIDEO DE LICENCIA LIBRE Y GRATUITA THEORA 1.0 Y  
COMPARACION CON EL CODEC H.264 EN LA ADAPTACION DE  
CONTENIDO MULTIMEDIA A SER TRANSMITIDO BAJO REDES IP

**Profesor guía: Ing. Luis Jacinto Fernández. Tesis. Caracas. U.C.V. Facultad de  
Ingeniería. Escuela de Ingeniería Eléctrica. Ingeniero Electricista. Opción:  
Comunicaciones. Trabajo de Grado. 2009.**

**Palabras Clave: Theora, H.264, video digital, codificación de video, calidad de  
video, comparación de codecs.**

**Resumen.** Dado el auge que la transmisión de vídeo tiene en el desarrollo de las comunicaciones, el presente estudio propone un análisis comparativo entre el CODEC de vídeo libre THEORA 1.0 y el CODEC propietario H.264 de MPEG, en la difusión de contenido multimedia bajo redes IP. La elección de una plataforma tecnológica para la difusión de contenido multimedia se verá influenciada por dos factores fundamentales; calidad y costos. En el caso específico de la codificación de video es necesario definir parámetros cuantificables que describan la “calidad de video” así como también la viabilidad de adoptar una codificación específica. El codec de video Theora, gratuito y de código abierto, se presenta como una alternativa a las soluciones propietarias como h.264. Las pruebas realizadas demostraron que Theora es capaz de entregar resultados comparables con los de h.264 y, a pesar de que las herramientas de codificación por hardware basadas en este codificador aun no han sido masivamente desarrolladas, se espera que tanto sus ventajas económicas como la inclusión de Theora en estándares importantes como el HTML5 sean fundamentales para convertirle en uno de los codificadores mas difundidos y utilizados. El estudio comparativo que se presenta no pretende establecer una jerarquía entre los codecs en estudio, tiene como finalidad contrastar sus ventajas y desventajas y dar las herramientas al desarrollador para elegir, con propiedad, la tecnología de codificación de video que más se adapte a sus necesidades tecnológicas y económicas.

## ÍNDICE GENERAL

Portada . . . . .	i
Constancia de aprobación . . . . .	ii
Dedicatoria . . . . .	.iii
Reconocimientos y agradecimientos . . . . .	iv
Resumen . . . . .	v
Índice General . . . . .	vi
Índice de tablas . . . . .	.xi
Índice de figuras . . . . .	xii
Índice de gráficas . . . . .	xii
Índice de ecuaciones . . . . .	xiii
Acrónimos . . . . .	.xiv
<b>INTRODUCCION . . . . .</b>	<b>1</b>
<b>OBJETIVOS . . . . .</b>	<b>4</b>
<b>CAPITULO I</b>	
<b>MARCO TEÓRICO . . . . .</b>	<b>6</b>
1.1.1.    Video Digital . . . . .	6
1.1.2.    Espacio de Color . . . . .	6
1.1.3.    Frame (Cuadro) . . . . .	7
1.1.4.    Bloque y Macrobloque. . . . .	7
1.1.5.    Píxel . . . . .	7
1.1.6.    Codec de Video. . . . .	8
1.1.7.    Contenedor. . . . .	8
1.2 Compresión de Video. . . . .	8
1.2.1.    Compresión SIN perdida. . . . .	9
1.2.2.    Compresión CON perdida. . . . .	9

1.3 Herramientas generales en el proceso de compresión. ....	9
1.3.1.    Compensación de Movimiento. ....	10
1.3.2.    Transformada discreta del coseno ....	10
1.3.3.    Cuantificación. ....	11
1.3.4.    Codificación de Entropía ....	11
1.3.5.    Codificación de longitud variable ....	12
1.3.6.    Codificación aritmética binaria adaptada al contexto ....	12

## **CAPITULO II**

<b>H.264 – MPEG</b> .....	13
2.1 Antecedentes de H.264. ....	13
2.2 Descripción del estándar H.264 .....	14
2.3 Perfiles de H.264. ....	14
2.4 Estructura de capas de H.264 .....	18
2.4.1.    NAL .....	18
2.4.2.    VCL .....	18
2.5 Algoritmo del Proceso de codificación de H.264. ....	18

## **CAPITULO III**

<b>Theora 1.0</b> .....	21
3.1 .1Antecedentes de Theora. ....	21
3.1.1.    Theora y VP3 .....	22
3.2 Proceso de Codificación de Theora. ....	23
3.2.1.    Consideraciones iniciales. ....	23
3.2.2.    Esquema de codificación de Theora 1.0	
3.3 Comparación de las características técnicas de Theora 1.0 y H.264 .....	24

## **CAPITULO IV**

<b>Transmisión de Video en redes IP</b> .....	33
---	----



## CAPITULO V

<b>Calidad de Video</b> .....	35
5.1 Degradación del Video Digital .....	35
5.2 Medición de la Calidad de Video .....	38
5.3 Procedimientos para realizar la medición de la calidad de Video codificado. ....	40
5.3.1.    Métodos Objetivos. ....	40
5.3.2.    Métodos Subjetivos. ....	41
5.3.3.    Métodos Mixtos. ....	43

## CAPITULO VI

<b>Pruebas de Desempeño</b> .....	47
6.1 Consideraciones Iniciales. ....	47
6.2 Parámetros a considerar. ....	48
6.3 Elección de la métrica y de las muestras. ....	49
6.4 Estructura general de las pruebas. ....	50
6.5 Pruebas de Calidad de Video Codificado. ....	52
6.5.1.    Calidad ofrecida por los codificadores. ....	52
6.5.1.1.    Medición del PSNR. ....	53
6.5.1.2.    Escala de Imparidad de Doble Estimulo ( <i>Double Stimulus</i> <i>Impairment Scale</i> , DSIS) .....	54
6.5.1.3.    Medición del SSIM. ....	55
6.5.2.    Relación Compresión-Bitrate Vs Calidad. ....	57
6.5.3.    Efecto de las características de la transmisión de datos en redes de paquetes en la calidad del video codificado. ....	59
6.5.4.    Efecto de la utilización de cuadros bipredictivos ( <i>B-frames</i> ) de H.264 en la calidad de video codificado. ....	62
6.5.5.    Consumo medio de recursos utilizados por los <i>codecs</i> durante el proceso de codificación. ....	63

6.6 Análisis de los resultados. ....	65
<b>CAPITULO VII</b>	
<b>Escalabilidad de H.264</b> .....	69
7.1 H.264/AVC .....	69
7.2 H.264/SVC .....	70
7.3 Escalabilidad y Theora 1.0. ....	72
<b>CAPITULO VIII</b>	
<b>Compatibilidad de Theora con las diversas plataformas tecnológicas</b> .....	73
8.1 Compatibilidad de Software .....	73
8.2 Compatibilidad de Hardware. ....	75
8.3 Theora y HTML5. ....	76
<b>CAPITULO IX</b>	
<b>Distribuciones y Licencias</b> .....	77
9.1 H.264 – Licencia. ....	79
9.2 Theora – Licencia .....	81
<b>CAPITULO X</b>	
<b>Thusnelda</b> .....	82
<b>CAPITULO XI</b>	
<b>Experiencias Propuestas</b> .....	84
<b>CONCLUSIONES</b> .....	92
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	96
<b>BIBLIOGRAFÍAS.</b> ....	98

## **ANEXOS**

Anexo 1. División del Cuadro en Macrobloques. ....	100
Anexo 2. Vectores de Movimiento. ....	101
Anexo 3. Comparación H.264 – Theora 1.0. ....	102
Anexo 4. Herramientas – Aplicaciones Utilizadas. ....	103
Anexo 5. Equipos Utilizados en las Pruebas. ....	106

## ÍNDICE DE TABLAS

TABLA 1 REQUERIMIENTOS POR APLICACIÓN .....	17
TABLA 2 CARACTERÍSTICAS TÉCNICAS DE H.264 Y THEORA 1.0 .....	31
TABLA 3 SECUENCIAS DE PRUEBA .....	50
TABLA 4 PARÁMETROS DE LOS CODIFICADORES .....	52
TABLA 5 RESULTADOS DEL PSNR PROMEDIO .....	53
TABLA 6 RESULTADOS DE LA MEDICIÓN DEL DSIS .....	54
TABLA 7 VALORES PROMEDIO DEL SSIM .....	56
TABLA 8 PSNR, SSIM VS. BER .....	60
TABLA 9 PSNR Y SSIM VS. VARIACIONES DEL ANCHO DE BANDA DISPONIBLE. ....	61
TABLA 10 DSIS ANTE VARIACIONES DEL ANCHO DE BANDA DISPONIBLE II .....	62
TABLA 11 PSNR, SSIM .....	62
TABLA 12 CARACTERÍSTICAS DE LAS MUESTRAS, PSNR Y SSIM .....	63
TABLA 13 COMPATIBILIDAD DE THEORA – SOFTWARE .....	74
TABLA 14 COSTOS DE H.264 .....	80
TABLA 15 OBJETIVOS A LOGRAR .....	85

## ÍNDICE DE FIGURAS

FIGURA 1 ALGORITMO CODIFICACIÓN DE H.264 .....	19
FIGURA 2 CUADROS I, P Y B .....	20
FIGURA 3A DIVISIÓN DEL FRAME, BLOQUES, SUPERBLOQUES .....	26
FIGURA 3B DIVISIÓN DEL FRAME, MACROBLOQUES, BLOQUES .....	26
FIGURA 4 ORDENAMIENTO DE BLOQUES, SUPERBLOQUES Y MACROBLOQUES .....	28
FIGURA 5 ESTRUCTURA DE LAS PRUEBAS .....	51

## ÍNDICE DE GRÁFICAS

GRÁFICA 1 PSNR – THEORA VS. H.264 – DELFINES_16CIF .....	54
GRÁFICA 2 H.264 VS. THEORA – DSIS .....	55
GRÁFICA 3 VARIACIONES DEL SSIM PARA H.264 Y THEORA .....	56
GRÁFICA 4 TASA DE BITS VS. PSNR .....	58
GRÁFICA 5 TASA DE BITS VS. SSIM .....	58
GRÁFICA 6 CONSUMO DE RECURSOS EN EL PROCESO DE CODIFICACIÓN, <i>COASTGUARD_CIF</i> .....	64
GRÁFICA 7, CONSUMO DE RECURSOS EN EL PROCESO DE CODIFICACIÓN II .....	64

## ÍNDICE DE ECUACIONES

ECUACIÓN 5.1 MSE .....	40
ECUACIÓN 5.2 RMSE .....	41
ECUACIÓN 5.3 PSNR .....	41
ECUACIÓN 5.4 SIMILITUD .....	45
ECUACIÓN 5.5 SSIM ..	46

## ACRÓNIMOS

<b>4MV</b>	4 Motion Vectors
<b>AWGN</b>	Additive White Gaussian Noise
<b>BER</b>	Bit Error Ratio
<b>BTFR</b>	British Telecom Full Reference
<b>CABAC</b>	Context-adaptive binary arithmetic coding
<b>CAVLC</b>	Contextbased Adaptive Variable Length Coding
<b>CIF</b>	Common Intermediate Format
<b>CLUF</b>	Contrato de Licencia para Usuario Final
<b>CODEC</b>	Codificador-Decodificador
<b>DCT</b>	Discrete Cosine Transform
<b>DPCM</b>	Differential Pulse Code Modulation
<b>DSIS</b>	Double Stimulus Impairment Scale
<b>EULA</b>	End User License Agreement
<b>FPGA</b>	Field Programmable Gate Array
<b>GNU</b>	General Public License
<b>AVC</b>	Advanced Video Coding
<b>SVC</b>	Scalable video Coding
<b>HTML</b>	HyperText Markup Language
<b>IP</b>	Internet Protocol
<b>ITU</b>	International Telecommunication Union
<b>MCU</b>	Multipoint Conference Unit
<b>MPEG</b>	Moving Picture Experts Group
<b>MSE</b>	Mean Square Error
<b>NAL</b>	Network Abstraction Layer
<b>PAL</b>	Phase Alternating Line
<b>PSNR</b>	Peak Signal-to-Noise Ratio

<b>QCIF</b>	Quarter Common Intermediate Format
<b>QoS</b>	Quality of Service
<b>RGB</b>	Red Green Blue
<b>RMSE</b>	Root Mean Squared Error
<b>RTCP</b>	Real time control protocol
<b>RTP</b>	Real-time Transport Protocol
<b>SSIM</b>	Structural Similarity
<b>UDP</b>	User Datagram Protocol
<b>VCEG</b>	Video Coding Experts Group
<b>VCL</b>	Video Coding Layer
<b>VGA</b>	Video Graphics Array
<b>VHDL</b>	Very high speed integrated circuits description language



## INTRODUCCION

Al igual que en el campo de las imágenes y el audio, en el mundo del video digital se establece la necesidad de lograr un equilibrio entre la compresión y la calidad, sin embargo, dado que el tamaño promedio de un video es generalmente mucho mayor que el de una imagen o una muestra de audio, es necesario incluir factores como la velocidad y el uso de recursos de procesamiento como partes fundamentales en la escogencia de una plataforma adecuada para convertir, comprimir y transmitir muestras de video. Así mismo, a nivel institucional-empresarial, es de vital importancia determinar los costos asociados con la elección de una tecnología determinada (patentes, licencias, etc.) y las consecuencias técnicas y compromisos adquiridos al adoptar una plataforma particular.

La integración del software en todo tipo de dispositivos electrónicos, diferentes al computador, abre una nueva vía de expansión en la cual el software libre se introduce como tecnología puntera, proporcionando independencia del proveedor, abaratando costos e incentivando la investigación y desarrollo de soluciones que ofrezcan una constante mejora en la calidad de servicio.

A nivel institucional y empresarial, la decisión de migrar hacia plataformas libres y de código abierto debe estar respaldada por estudios que evalúen las características técnicas y económicas de determinadas soluciones libres frente a sus contrapartes propietarias, determinando así su conveniencia y viabilidad.

Dado el auge que la transmisión de video tiene en el desarrollo de las comunicaciones, el presente estudio propone un análisis comparativo entre el códec de video libre THEORA 1.0 y el códec propietario H.264 de MPEG, en la difusión de contenido multimedia bajo redes IP.

Theora es un códec de video libre que está siendo desarrollado por la Fundación Xiph.org, como parte de su proyecto Ogg. Está basado en el códec VP3, donado por On2 Technologies, quien a su vez donó el código fuente del VP3.1 a la Fundación Xiph.org para, luego de ser refinado y extendido, ser publicado bajo

licencia BSD (*Berkeley System Distribution*). El códec propuesto, Theora, es un códec de transformación basado en bloque que utiliza la Transformada de coseno discreta 8 x 8 de Tipo-II y compensación de movimiento basada en bloque. Esto lo coloca en la misma clase de códecs que MPEG-1, MPEG-2, MPEG-4, y H.264. Sin embargo, los detalles de cómo los bloques individuales son organizados y cómo los coeficientes DCT son almacenados en el bitstream difieren substancialmente de estos códecs. [7]

Al intentar realizar una comparación entre *códecs* de video, se presenta el problema de “definir y cuantificar la calidad”. Dado que la calidad de un video dependerá directamente de la percepción del observador, la solución óptima a esta interrogante consiste básicamente en obtener los valores de diferentes parámetros específicos y bajo diversas condiciones de codificación, y, de esta forma, realizar una estimación sobre la calidad de video. Entre los parámetros que se incluirán en el presente estudio, se encuentran el PSNR, SSIM y DSIS, sin desestimar las mediciones y pruebas relativas al consumo de recursos de procesador y memoria.

La calidad de una muestra de video codificado no puede referirse solo al proceso de codificación en sí, es necesario estudiar también los posibles efectos que los parámetros inherentes a la transmisión del video pueden tener en la calidad del mismo. El material multimedia, y, en especial, el video, es muy sensible al retardo de la red, la pérdida de alguna información en el video sin comprimir no es muy relevante, ya que al perderse un fotograma, el siguiente fotograma proporciona la suficiente información para poder interpretar la secuencia. En cambio, el video comprimido es mucho más sensible a errores de transmisión, ya que las técnicas de compresión que se valen de la redundancia espacial y temporal pueden perder la información de esta redundancia y los efectos de la falta de datos pueden propagarse en los próximos fotogramas. Específicamente, se estudiarán los posibles efectos o alteraciones introducidas en las muestras de video codificadas durante el proceso de transmisión sobre redes IP de ancho de banda limitado.

Es necesario destacar que un estudio comparativo entre codificadores de video debe estar enmarcado en un contexto cuya base sea la aplicación o el tipo de solución

que se pretenda implementar con ellos, sin embargo, dada la poca información existente entre las características de desempeño de códecs como Theora, este trabajo tratará de representar una gama amplia de posibilidades y dejará, para futuras y específicas investigaciones, la responsabilidad de establecer un criterio de comparación más complejo y adaptado a las necesidades específicas de las posibles soluciones que incluyan los procesos de codificación y transmisión de video.

Se presume que el códec óptimo será aquel capaz de entregar la calidad de video requerida, que sea compatible y capaz de ser integrado a las infraestructuras existentes, que permita un desarrollo escalable y cuya adquisición, mantenimiento, actualización e implementación sean económicamente viables.

## OBJETIVOS

Para determinar si el códec Theora cumple con las características mencionadas, se plantean, en principio, los siguientes objetivos:

- Objetivo General:
  1. Estudiar en forma comparativa las propiedades y desempeño del códec para compresión de video Theora 1.0 (Ogg-Vorbis, Nov 08), con el códec H.264 (MPEG) en la adaptación de contenido multimedia para ser transmitido sobre redes IP.
- Objetivos Específicos:
  1. Caracterizar el códec Theora 1.0, los métodos de compresión y las herramientas matemáticas utilizadas en él.
  2. Determinar, mediante la medición de parámetros de Calidad de video, el rendimiento de los códec Theora 1.0 y H.264 en el proceso de codificación-transmisión-decodificación de muestras de video bajo diferentes parámetros de codificación y diversas condiciones de la Red.
  3. Determinar una relación entre los niveles de consumo de recursos de procesamiento y memoria que son requeridos en el proceso de codificación y decodificación por los códec Theora 1.0 y H.264.
  4. Obtener una relación “Calidad de video” Vs. tamaño del archivo codificado (Resolución y frame-rate Vs. bite-rate) para muestras de video codificadas con H.264 y Theora 1.0.
  5. Realizar un análisis comparativo entre las características técnicas, económicas, de adaptabilidad y oportunidad al desarrollo científico, de los códec Theora 1.0 y H.264.
  6. Evaluar la relevancia de la escalabilidad de capas de H.264 en el

desempeño del proceso de codificación-transmisión-decodificación de muestras de video para difusión sobre redes IP con y sin priorización de paquetes y el efecto de esta característica en el análisis comparativo con Theora 1.0.

7. Evaluar la compatibilidad que presentan los codificadores de video Theora 1.0 y H.264 en las plataformas tecnológicas (software y hardware) comúnmente utilizadas, para estimar los gastos inherentes a la implantación de dichos procesos en las infraestructuras existentes.
8. Analizar los resultados obtenidos en las distintas pruebas y simulaciones para realizar las recomendaciones pertinentes a la viabilidad de la adopción e implementación de soluciones de compresión de video bajo software gratuito y libre en diversos ámbitos sociales (Institucional, Educativo, Empresarial).
9. Proponer experiencias que puedan ser realizadas en los laboratorios de comunicaciones de la Escuela de Ingeniería Eléctrica de la U.C.V. y que den pie a la elaboración de futuras “Prácticas de laboratorio” donde se evidencie, no solo el proceso de codificación de video, sino las posibles alternativas gratuitas y libres para la implementación de soluciones con este propósito.

En fin, este trabajo no tiene como finalidad el dilucidar cuál es el mejor códec, por el contrario, espera proporcionar al lector las herramientas necesarias para estimar, según sus propias necesidades y basado en un conjunto claro de parámetros cuantificables, cuál de los códecs en estudio le conviene adoptar para la difusión de contenido multimedia sobre redes IP.

# CAPITULO I

## MARCO TEÓRICO

### 1.1 Video Digital.

Todo dispositivo para la grabación o reproducción de imágenes en movimiento basa su funcionamiento, no en el movimiento en si, sino en la captura y almacenamiento de imágenes fijas en intervalos determinados. Cuando estas imágenes son reproducidas secuencialmente y a una velocidad mínima determinada, el cerebro interpola las imágenes faltantes entre las capturadas y produce la impresión de movimiento. El proceso de captura de imágenes en intervalos de tiempo se denomina *sampleo*, y basta reproducir estas imágenes a una velocidad mínima de 24 cada segundo para que el cerebro humano sea “engañado” y tenga la sensación de que está observando movimiento constante.

Para almacenar un video es sólo necesario almacenar una sucesión de imágenes. En el mundo digital existen una serie de mecanismos para el almacenamiento de imágenes fijas y estos mecanismos tienen en común la necesidad de dividir la imagen en unidades mínimas de información (*pixels*) y representar las características de este *pixel* en forma discreta. La información a almacenar para cada *pixel* podrá ser, entre otras, un valor binario (como “1” o “0”) para el caso de las imágenes en blanco y negro, o incluso un vector de  $n$  dimensiones para las imágenes a color.

#### 1.1.2 Espacio de Color.

Un espacio de color es una representación abstracta que permite describir numéricamente el color “físico” de un píxel en forma de un vector numérico. En

general, tres componentes son necesarias y suficientes para representar el color y para ser interpretado por el ojo humano, típicamente, las diferentes combinaciones de los colores rojo, verde y azul permiten describir la totalidad de colores que éste es capaz de diferenciar. Esta representación tridimensional de rojo (R) , verde (G) y azul (B) definen el espacio de color RGB. Sin embargo, los procesos de digitalización suelen utilizar el hecho de que el sistema visual humano es mas sensible a las variaciones de brillo que de color para definir un espacio de color denominado YUV, en el cual se considera una dimensión (Y) para representar el brillo o luminancia y dos dimensiones (U y V) para definir las variaciones de color o crominancia. Visto que la percepción de la crominancia es menos sensible que la de la luminancia, esta puede ser muestreada considerando menos información, y por tanto, puede reducirse la cantidad total de bits a almacenar o enviar, manteniendo una correspondencia cuasi exacta con otros espacios como el RGB.

### **1.1.3 Frame (Cuadro)**

En el ámbito de video digital, se le denomina *Frame* al conjunto de  $n$  píxels verticales  $x$   $m$  *pixels* horizontales que definen a una imagen.

### **1.1.4 Bloque y Macrobloque.**

Un bloque es un arreglo bidimensional de píxels contiguos que se agrupan en función de mejorar el proceso de codificación de una imagen. Un macrobloque es un conjunto de bloques contiguos.

### **1.1.5 Píxel.**

Considerando que se puede dividir espacialmente una imagen en pequeñas unidades de información ordenadas en un arreglo bidimensional (generalmente vertical y horizontal), un píxel es la mínima división que puede tomarse y que

contiene la totalidad de la información que lo caracteriza. La cantidad de píxels definidas horizontalmente, multiplicada por la cantidad vertical, define la resolución de la imagen (por ejemplo 640x480 píxels representa la resolución VGA).

#### **1.1.6 códec de Video.**

El algoritmo que permite almacenar un video en forma digital se denomina codificador, así mismo aquel que permite recuperar la información codificada en un video digital para convertirla en una sucesión de imágenes es denominado decodificador. Un *códec* es el conjunto de algoritmos codificador-decodificador.

#### **1.1.7 Contenedor.**

Un Contenedor es un tipo de archivo informático con características definidas y en el cual es posible almacenar diversos contenidos multimedia codificados por sus respectivos *códecs*.

### **1.2 Compresión de Video.**

La compresión de video implica reducir y eliminar datos redundantes de él para que el archivo de video digital pueda enviarse y almacenarse de manera eficiente. La necesidad de la compresión viene dada por la magnitud de la cantidad de información que es necesario almacenar/enviar para representar un video. Por ejemplo, una sola imagen en resolución PAL (768x576) almacenada en un espacio típico de color RGB a 24-bit ocupa un espacio de 1.27MB, ya que PAL utiliza 25 *frames* cada segundo, para mantener la sensación de movimiento fluido, esto daría un total aproximado de 32MB por cada segundo de video, o 111 GB por cada hora.



### **1.2.1 Compresión SIN pérdida.**

Los algoritmos de compresión básicamente explotan la redundancia de datos en el contenido original, es así como, por ejemplo, se identifican aquellos elementos de información que se repiten frecuentemente y se codifican con una cantidad de bits inversamente proporcional a la recurrencia de dichos elementos, o, específicamente en video, no se almacena toda la información de un *frame*, sino que se almacenan solo los cambios o diferencias con respecto al *frame* anterior. Este tipo de algoritmos de compresión, entre otros, tienen la característica de que su contraparte (el algoritmo de-compresor) es capaz de restituir la información exacta del video original, es decir, no existe pérdida de información en el proceso.

### **1.2.2 Compresión CON pérdida.**

Son aquellos algoritmos de compresión, cuyo resultado es un video con menor calidad que el video original; la diferencia de calidad puede o no ser tal que sea percibida por el observador. Sin embargo, mientras los algoritmos SIN pérdida permiten en promedio una reducción de entre 10x y 20x el tamaño del flujo de información, los algoritmos CON pérdida han llegado a relaciones de hasta 200x con diferencias de calidad despreciables entre el video original y el codificado.

## **1.3 Herramientas generales en el proceso de compresión.**

La técnica de compresión de video consiste en tres pasos fundamentalmente, primero el preprocesamiento de la fuente de video de entrada, paso en el cual se realiza el filtrado de la señal de entrada para remover componentes no útiles y el ruido que pudiera haber en ésta. El segundo paso es la conversión de la señal a un formato intermedio común (CIF), y por último el paso de la compresión. Las imágenes comprimidas son transmitidas a través de la línea de transmisión digital y se

hacen llegar al receptor donde son reconvertidas al formato común CIF y son desplegadas después de haber pasado por la etapa de post-procesamiento.

### **1.3.1 Compensación de Movimiento.**

La compensación de movimiento es una estrategia mediante la cual un *frame* es descrito en términos de la transformación de un *frame* de referencia en el *frame* actual. El cuadro o *frame* de referencia puede ser un cuadro previo e incluso un cuadro posterior. La compensación de movimiento se basa en el hecho de que frecuentemente, entre un grupo de cuadros de video, la única diferencia entre uno y otro es el resultado de que la cámara (o toma) se mueve, o hay objetos o personas moviéndose en el cuadro. Esto implica que en un video, la información que representa un cuadro será la misma información necesaria para representar el siguiente, a esto se le denomina “*redundancia temporal*”. Las técnicas de compensación de movimiento aprovechan la redundancia temporal para, en lugar de enviar la información de cada *frame* por separado, enviar únicamente la diferencia entre el cuadro previo (almacenado) y el cuadro actual. La forma en la cual un *códec* determina el o los cuadros de referencia a utilizar son específicas de cada algoritmo, sin embargo, todos presentan la necesidad de enviar cuadros de referencias “completos” o que representen la totalidad de la información cada cierto tiempo para limitar que un posible error de transmisión se propague al resto del video codificado.

### **1.3.2 Transformada discreta del coseno**

Típicamente las imágenes presentan una alta “redundancia espacial”, esto es, el valor de cada píxel está muy relacionado con el de sus píxeles vecinos. Para aprovechar esta característica, los algoritmos de compresión utilizan herramientas como la Transformada del coseno. La transformada del coseno o DCT es una implementación específica de la transformada de Fourier donde la imagen es transformada de su representación espacial a su frecuencial equivalente. Cada

elemento de la imagen se representa por ciertos coeficientes de frecuencia. Las zonas con colores similares se representan con coeficientes de baja frecuencia y las imágenes con mucho detalle con coeficientes de alta frecuencia. Al aplicar la transformada discreta del coseno muchos de los coeficientes obtenidos son pequeños, es decir, la mayoría de la energía de los datos se almacena en pocos coeficientes. La DCT se aplica generalmente a bloques de 8x8 píxeles, representados por una matriz de 8x8 elementos; la matriz resultante presenta características óptimas para el posterior proceso de *Cuantificación*. En general, la transformada discreta del coseno tiene una excelente compactación de energía para datos altamente correlacionados, la eficiencia de la compactación dependerá también, en gran medida, de la forma en la cual se agrupan y ordenen los coeficientes de la matriz transformada, proceso que cada algoritmo de codificación puede realizar de manera distinta.

### **1.3.3 Cuantificación.**

La cuantificación es el mecanismo que permite reducir la precisión con la que los coeficientes de la DCT se representan cuando se convierte la DCT a una representación entera. Esto puede ser muy importante en compresión de imagen, en donde se tiende a anular muchos coeficientes, especialmente los de altas frecuencias espaciales. El proceso de cuantificación involucra pérdida de información, sin embargo, dado que la capacidad del sistema visual humano para detallar los cambios en frecuencia es limitada, las aproximaciones en la cuantificación pueden realizarse sin arriesgar en gran medida la calidad de la percepción del video.

### **1.3.4 Codificación de Entropía**

Los codificadores de entropía se basan en asignar palabras de código de longitud variable a cada muestra. Asignan palabras cortas a los valores de muestras más probables, y palabras largas a los valores de muestras menos probables. El modelo de probabilidad para la asignación de palabras de código, puede obtenerse

directamente de los valores de entrada o de suposiciones previas sobre dichos valores.

### **1.3.5 Codificación de longitud variable (Variable-length code - VLC).**

VLC es un tipo de codificación en el cual los símbolos a codificar son mapeados utilizando un número variable de bits.

### **1.3.6 Codificación aritmética binaria adaptada al contexto (CABAC)**

*Cabac* es un método de codificación VLC que hace uso de múltiples modelos probabilísticos, primero realiza una conversión binaria de todos los símbolos a codificar, luego, para cada símbolo, elige el modelo probabilístico más adecuado para la codificación, por último, utiliza la información de los símbolos próximos para mejorar la estimación probabilística del modelo elegido.

## CAPITULO II

### H.264 – MPEG

#### 2.1 Antecedentes de H.264.

Desde principios de los años 90's el grupo de expertos en codificación de video de la ITU-T conocido como VCEG (*Video Coding Expert Group*) a la par de el grupo MPEG (*Moving Picture Expert Group*), han dedicado sus esfuerzos al estudio de diversas técnicas de codificación de video y sus aplicaciones. La ITU-T comenzó por desarrollar el estándar H.261 [1] para aplicaciones de video- conferencia; por su parte, y con la finalidad de lograr el almacenamiento eficiente de video en discos compactos, MPEG desarrollo el MPEG-1 y su extensión MPEG-2 cuya aplicación inicial fue la televisión digital, tanto estándar como de alta definición (SDT y HDTV). A mediados de los años 90, la ITU-T estandarizó el MPEG-2 como H.262 [2]. El funcionamiento del estándar H.262 y sus antecesores se basa en la división en rebanadas de los cuadros de video de entrada; posteriormente estas rebanadas de video se dividen en macrobloques y bloques. Entre 1.996 y 1.998 se desarrolló y estandarizó la norma H.263 orientado específicamente a aplicaciones de video móvil. [3]

Dada la necesidad de un estándar capaz de manejar distintas aplicaciones de video, la ISO/IE desarrolló un estándar que agregaba a las características de H.262 la capacidad de segmentar objetos de video con el fin de codificar cada objeto por separado y permitir a las aplicaciones interactuar con ellos, a este estándar se le denominó como MPEG-4 parte 2 [4] y entre sus características principales se destaca el hecho de estar compuesto por varios perfiles que manejan diferentes velocidades compatibles con diversas aplicaciones.

## 2.2 Descripción del estándar H.264

El estándar H.264 hace uso de herramientas como la transformación y predicción para la reducción de la correlación espacial, el control del *bit-rate*, la predicción por compensación de movimiento para reducir la redundancia temporal, así como la codificación de la entropía para reducir la correlación estadística. Sin embargo lo que aporta H.264 a la mejora en la eficiencia del proceso de codificación es la forma en la que opera cada uno de sus bloques funcionales. H.264 utiliza predicción *intra* cuadro (INTRA), *transformación* de bloques de 4x4 muestras cuyos coeficientes transformados resultan enteros, referencias múltiples para la predicción temporal, tamaño variable de los macrobloques, entre otras. Todas estas mejoras vienen acompañadas de un aumento en la complejidad de la implementación.

El estándar H.264, al igual que sus antecesores, se define mediante diferentes perfiles y niveles, cada perfil establece una serie de características y límites del decodificador; los niveles establecen los valores máximos y mínimos que pueden tomar los elementos tanto del flujo de bits como de la sintaxis del estándar. En la primera versión del H.264 existen tres perfiles, el línea base o *baseline*, el principal o *main* y el extendido o *extended*. El perfil línea base se aplica a los servicios de conversación en tiempo real, como video conferencia y video teléfono. El perfil principal es para aplicaciones de almacenamiento digital de video y datos, así como de transmisión de televisión. El perfil extendido es aplicable también a servicios de multimedia en Internet.

## 2.3 Perfiles de H.264.

Entre las características que presenta el estándar H.264 vale la pena mencionar:

### **Presentes en todos los perfiles:**

- Rebanadas tipo I (Rebanadas codificadas utilizando predicción INTRA): Rebanada codificada utilizando la predicción de las muestras decodificadas dentro de la misma rebanada.
- Rebanadas tipo P (Rebanadas codificadas utilizando codificación predictiva en un solo sentido): Rebanada codificada utilizando predicción intercuadro (INTER). Se utilizan como referencia los cuadros previamente decodificados con un vector de movimiento y un índice de referencia para predecir los valores de las muestras de cada bloque.
- Utiliza el codificador CAVLC (Contextbased Adaptive Variable Length Coding) para la codificación de la entropía.

### **En el perfil de Línea Base:**

- Ordenamiento flexible de macrobloques: Los macrobloques no necesariamente se ordenan horizontalmente y de izquierda a derecha (*raster scan*). Existe un mapa que asigna los macrobloques a un determinado grupo de rebanadas.
- Ordenamiento arbitrario de rebanadas: La dirección del primer macrobloque de una rebanada puede ser menor que la dirección del primer macrobloque de alguna rebanada anterior dentro del mismo cuadro codificado.
- Rebanada redundante: Esta rebanada pertenece a datos redundantes ya codificados con igual o diferente velocidad binaria en comparación con los mismos datos codificados pertenecientes a la misma rebanada.

### **En el perfil Principal:**

- Rebanadas tipo B (Rebanadas codificadas utilizando codificación predictiva bidireccional): Rebanada codificada utilizando predicción INTER con referencia de un cuadro previamente decodificado y utilizando a lo más, dos vectores de movimiento e índices de referencia para predecir los valores de las muestras de cada bloque.
- Predicción con peso: Esta es una operación de escalamiento, en la cual se aplica un factor de peso a las muestras resultantes de la compensación de movimiento en rebanadas tipo P o B.
- Utiliza el codificador CABAC (*Contextbased Adaptive Binary Arithmetic Coding*) para codificación de la entropía.

### **En el Perfil Extendido:**

- Se incluyen todas las partes del perfil línea base.
- Rebanadas tipo SP: Es un tipo de rebanada especial, codificada para cambiar eficientemente de resoluciones entre tramas de video, es similar a la codificación de una rebanada tipo P.
- Rebanadas tipo SI: Igual que el anterior, pero es similar a codificar una rebanada tipo I.
- Particionamiento de datos: Los datos codificados se colocan en particiones separadas en capas diferentes. Se utiliza para reducir errores durante la transmisión y para el empaquetamiento efectivo de los datos.

### **En el perfil Alto:**

- Incluye todas las partes del perfil principal (Main).
- Transformación por bloques de tamaño adaptivo: DCT de 4x4 y



transformación DCT entera para los bloques de luminancia. El tamaño del bloque a codificar dentro de un macrobloque puede ser diferente.

- Matrices de cuantización escaladas: Se utilizan diferentes escalas de acuerdo con frecuencias específicas, asociadas con los coeficientes transformados, para optimizar la calidad subjetiva del video.

Los niveles de cada perfil establecen los diferentes tamaños de los cuadros de entrada que el *códec* es capaz de manejar eficientemente (QCIF, CIF, 4CIF, SDTV, HDTV, S-HDTV, etc.). Cada nivel ajusta los límites de las velocidades binarias, tamaño de la memoria para almacenar cuadros de referencia, entre otros parámetros.

La Tabla 1. “Requerimientos por aplicación” contiene la información que relaciona el uso de los diferentes perfiles de H.264 y los requerimientos necesarios para codificar muestras de video destinadas a diversas aplicaciones.

<b>APLICACIÓN</b>	<b>REQUERIMIENTOS</b>	<b>PERFIL DE H.264</b>
<b>Transmisión de TV</b>	<ul style="list-style-type: none"> <li>• Eficiencia en la codificación</li> <li>• Confiabilidad en canales de distribución controlados</li> <li>• Decodificador de baja complejidad.</li> </ul>	<b>PRINCIPAL</b>
<b>Video por Cable o Internet</b>	<ul style="list-style-type: none"> <li>• Eficiencia en la codificación.</li> <li>• Confiabilidad en canales NO controlados para redes basadas en paquetes.</li> </ul>	<b>EXTENDIDO</b>
<b>Videoconferencia</b>	<ul style="list-style-type: none"> <li>• Alta eficiencia del codificador/decodificador</li> </ul>	<b>BASICO</b>
<b>Video a través de redes Inalámbricas</b>	<ul style="list-style-type: none"> <li>• Eficiencia de codificador/decodificador.</li> <li>• Bajo consumo de potencia y recursos de procesamiento.</li> </ul>	<b>BASICO</b>

**Tabla 1,** Requerimientos por aplicación, fuente: Autor

## **2.4 Estructura de capas de H.264**

H.264 se compone de dos capas, la capa de la red de abstracción ( *Network Abstraction Layer - NAL*) y la capa de codificación del video ( *Video Coding Layer - VCL* ).

### **2.4.1 NAL**

NAL abstrae los datos para hacer compatible al tren de bits de salida del codificador con casi todos los canales de comunicación o medios de almacenamiento. Esta unidad de red, especifica los datos en un formato de bytes (*byte-stream*) o de paquetes. El formato de bytes define patrones de bytes o de bits utilizados en el estándar H.320 o en el MPEG-2. El formato de paquetes, define paquetes de datos identificables por protocolos de transporte para aplicaciones de RTP/UDP/IP.

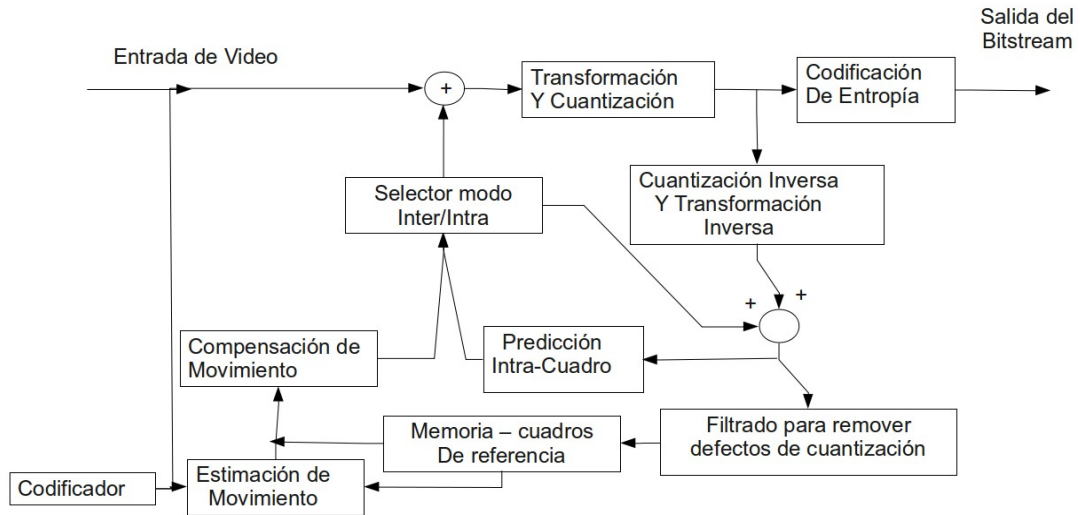
### **2.4.2 VCL**

La capa VCL constituye el núcleo de los datos codificados. Ésta consiste en la secuencia de video a codificar, cuadros o campos dentro de la secuencia de video con tres arreglos de muestras (luminancia, crominancia o RGB), rebanadas dentro de cada cuadro y macrobloques dentro de cada rebanada, así como bloques dentro de cada macrobloque.

## **2.5 Algoritmo del Proceso de codificación de H.264.**

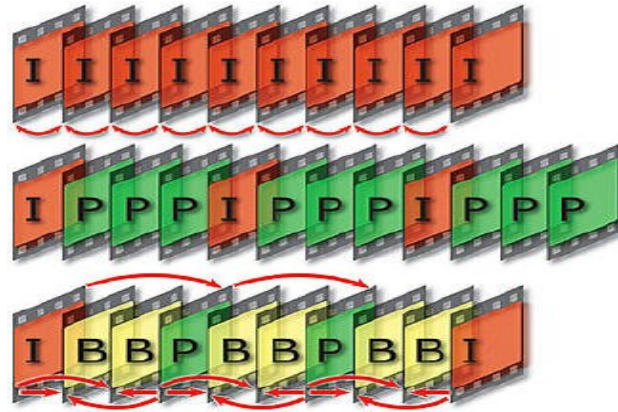
La **Figura 1** representa el algoritmo simplificado del proceso de codificación de H.264:

**Figura 1, Algoritmo Codificación de H.264, [5]**



El codificador debe seleccionar entre codificación *Intra* o *Inter*. En codificación *Intra* se utilizan varios modos de predicción para reducir la redundancia espacial de un solo cuadro. La codificación *Inter* es más eficiente y se utiliza en la codificación tipo P o B (predictiva o bidireccional) de cada bloque de muestras. En esta codificación se utilizan como referencia los cuadros decodificados previamente. La codificación *Inter* utiliza vectores de movimiento para reducir la redundancia temporal entre cuadros. La predicción se obtiene después de filtrar el bloque anterior reconstruido. El filtro reduce las posibles distorsiones introducidas debido a la cuantización. Los vectores de movimiento y el modo de predicción *Intra* pueden tener varias especificaciones, dependiendo del tamaño de los bloques a codificar. Antes de ser cuantizado, el error o predicción residual se comprime aún más utilizando una transformada, la cual remueve la correlación espacial del bloque. Finalmente, los vectores de movimiento y los coeficientes cuantizados se codifican utilizando codificadores de entropía tales como el CAVLC (*Context-Adaptive Variable Length Codes*) o el CABAC (*Context-Adaptive Binary Arithmetic Codes*).

La **Figura 2** muestra el flujo de *frames* y su asignación como cuadros I, P y B



**Figura 2,** Cuadros I, P y B, [6]

## CAPITULO III

### Theora 1.0

#### 3.1 Antecedentes de Theora.

Theora es un *códec* de video libre que está siendo desarrollado por la Fundación *Xiph.org*, como parte de su proyecto *Ogg*. Está basado en el códec VP3, donado por *On2 Technologies*, quien a su vez donó el código fuente del VP3.1 a la Fundación *Xiph.org* para, luego de ser refinado y extendido, ser publicado bajo licencia BSD (*Berkeley System Distribution*).

En junio de 2002, On2 Technologies y la Fundación Xiph.org, anunciaron una alianza para desarrollar Ogg Theora, la integración de VP3 con el proyecto Ogg y Vorbis. El objetivo es la posibilidad de tener ficheros Ogg, donde Theora actúa como la capa de video, mientras que Vorbis actúa, generalmente, como la capa de audio. Esto permite trabajar con audio y video sin tener que maniobrar con formatos cerrados o de pago, es decir, teniendo una alternativa libre y competitiva a los formatos del grupo MPEG.

La Fundación Xiph.org anunció, en Noviembre del 2008, la salida al público de la versión 1.0 de su códec de video Ogg Theora (libre), con el que pretenden ofrecer una calidad encomiable junto a un consumo reducido de CPU para su utilización, y quizá el punto más importante es que no requiere ningún tipo de licencia para su uso, sea cual sea el dispositivo o software que lo use. [7]

Theora es un códec de transformación basado en bloque que utiliza la Transformada de coseno discreta 8 x 8 de Tipo-II y compensación de movimiento basada en bloque. Esto lo coloca en la misma clase de códecs que MPEG-1, MPEG-2, MPEG-4, y H.264. Sin embargo, los detalles de cómo los bloques individuales son organizados y cómo los coeficientes DCT son almacenados en el bitstream difieren

substancialmente de estos códecs. Theora soporta únicamente cuadros intra (cuadros I en MPEG) y cuadros inter (cuadros P en MPEG). No existe un equivalente a los cuadros bi-predictivos (cuadros B) encontrados en códecs MPEG.

### 3.1.2 Theora y VP3

El proceso de codificación de VP3 inicia con la separación de cada *frame* en sus componentes en un espacio de color YUV y subdividiendo cada plano en series de 8x8 bloques a los que se denominan fragmentos. Las especificaciones de VP3 incluyen también el concepto de superbloques, que no son mas que agrupaciones de 16 bloques en matrices de 4x4. Al igual que los definidos en otros estándares como JPEG y MPEG, VP3 hace uso de la noción de macrobloques (un macrobloque incluira 4 bloques del plano Y en una matriz de 2x2 y un bloque para cada uno de los planos de crominancia. La diferencia fundamental, además del tamaño, entre los superbloques y los macrobloques es que mientras un superbloque contiene información de un único y específico plano, la información de un macrobloque involucra al espacio de color completo (planos Y,U y V).

VP3 define diferentes tipos de *frames* o cuadros al igual que la mayoría de los códecs de video, denomina “*golden frame*” a los cuadros de referencia (*I-Frames* de MPEG) y mantiene la nomenclatura de los *P-frames* para los cuadros codificados por predicción.

El algoritmo de VP3 transforma cada fragmento de un *golden-frame* utilizando la DCT, cada grupo de coeficientes de la transformación es muestreado y cuantificado, y los coeficientes DC son reducidos utilizando DPCM en combinación con los coeficientes DC de fragmentos (bloques, superbloques etc) adyacentes. Posteriormente los coeficientes son codificados entrópicamente.

La codificación de un *golden-frame* solo puede hacerse de forma *Intra*. Para los *frames* siguientes, VP3 puede elegir entre hasta 8 modos de codificación distinto, que incluyen, entre otros, tratarlo como si se tratara de un cuadro de referencia (*golden frame*) o codificarlo de forma predictiva utilizando como referencia el cuadro

anterior o el *golden-frame* anterior (de la misma coordenada o de una coordenada diferente y agregando la información de un vector de movimiento).

Para el proceso de cuantización, VP3 posee 3 matrices estáticas, una para los fragmentos de un *gold-frame* de luminancia, otra para los de crominancia de un *golden-frame*, y una última que puede ser utilizada tanto en los *golden-frames* como en los *intra-frames* en los planos de luma o croma. A pesar de que las matrices de cuantización son estáticas, éstas deben ser pre-calculadas y/o ajustadas de acuerdo al índice de calidad (*qi*) establecido en el encabezado (*header*) del stream de video y que es fijada por el usuario.

El código VP3 estableció las bases para el desarrollo de Theora el cual conserva muchas de sus características. Un video codificado con VP3 puede ser convertido o recodificado SIN pérdida con Theora, sin embargo, dado que Theora incorpora una serie de nuevas características, el proceso inverso, sin pérdida, no es posible. [8]

## **3.2 Proceso de Codificación de Theora.**

### **3.2.1 Consideraciones iniciales**

1. La versión de Theora en estudio (Theora 1.0) es capaz de codificar flujos de video de dimensiones arbitrarias, el flujo de cuadros debe ser constante (*frame-rate*) en el tiempo y estar representado en uno de muchos espacios de color de la forma *YcbCr* y puede manejar sub-muestreo de croma de 4:2:2, 4:2:0 y 4:4:4. El codificador Theora no soporta como entrada video interlazado, de *frame-rate* variable y de profundidad de bit de muestreo mayor a 8bit por componente. Las limitaciones expresadas son válidas solo para la versión 1.0, futuros desarrollos consideran eliminarlas.
2. Theora no provee ningún tipo de ordenamiento de trama, sincronización o protección contra errores de transmisión, por tanto, los paquetes resultado de una codificación con Theora deben ser re-empaquetados en un contenedor

adecuado para el mecanismo de transporte. Dado que Theora no presenta limitaciones de *bit-rate*, los paquetes codificados con este *códec* no tendrán valores mínimos, máximos o esperados-por-paquete, facilitando la inclusión en contenedores de media comunes. Para la realización de este trabajo se supondrá, a menos se indique lo contrario, que el contenedor por defecto de los paquetes Theora será *Ogg*, sin que esto de ninguna manera represente una norma.

3. Un cuadro de video codificado con Theora debe tener dimensiones (alto y ancho) que sean múltiplos de 16 para permitir el trabajo con números enteros de bloques aun cuando los planos de croma sean sub-muestreados. Sin embargo, dentro de un *frame* puede establecerse una región menor que contenga el video en si y cuya dimensión puede ser arbitraria.
4. La configuración general del *stream* codificado consiste en un conjunto de datos que incluyen, entre otros, el *frame rate*, el tamaño del cuadro y de la región de imagen (en el caso de que sean diferentes) y su posición dentro del cuadro, el espacio de color, el formato de píxel y la versión del codificador.
5. Theora incluye en la cabecera del flujo de datos toda la información referente al modelo probabilístico para el cálculo de los coeficientes de la DCT al igual que los datos referentes a los parámetros de cuantificación. Por esta razón, un decodificador Theora requiere ser inicializado con el encabezado del *stream* antes de comenzar el decodificado, esto hace imposible la decodificación de un *frame* cualquiera en el flujo sin contar con el encabezado del *stream* para inicializar el decodificador. Sin embargo, una vez que el decodificador ha sido inicializado, el decodificador es capaz de decodificar cualquier flujo arbitrario del *stream*.

### **3.2.2 Esquema de codificación de Theora 1.0**

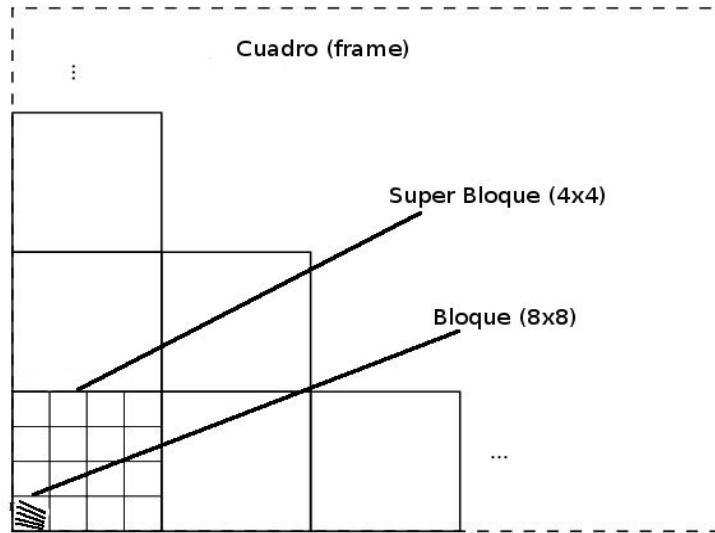
El proceso de codificado de un flujo de imágenes en movimiento con Theora 1.0 es similar al de otros *códecs* de video conocidos, incluso comparte muchas de las



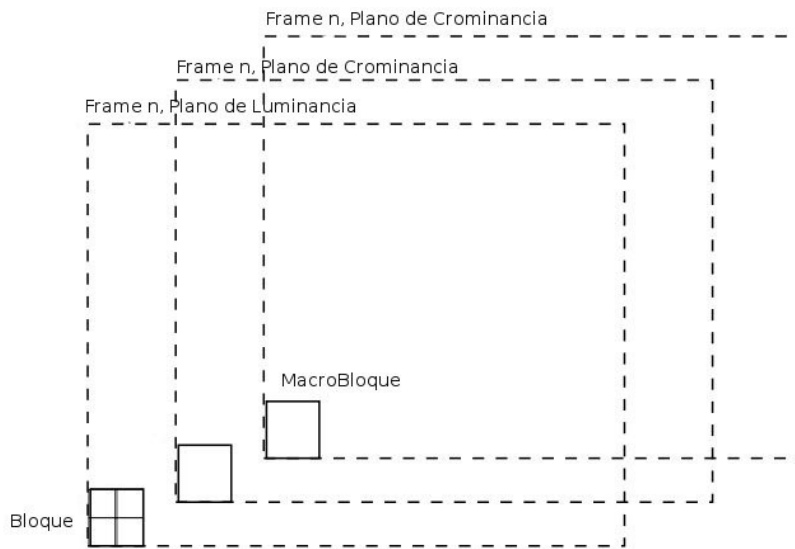
características del codificador H.264 de MPEG con el que se realiza este estudio comparativo. Sin embargo, y aunque no con todo detalle, se procederá a realizar un esquema del proceso de Theora y sus principales características:

- Theora divide el arreglo de pixels para separar los planos de color Y (luma), Cb y Cr (croma). En ciertos espacios de color determinados, y como se explicó anteriormente, los planos de croma son sub-sampleados en un factor de 2, esto, en términos físicos, implica que el tamaño (altura o anchura) de los planos de croma puede llegar a tener la mitad del tamaño que el plano de luminancia.
- Cada plano de color es subdividido en bloques de 8x8 pixels y son agrupados en arreglos de 4x4 bloques, a este arreglo se le denomina *super-bloque*. Cada plano de color contendrá su propio y único set de bloques y superbloques. En el caso de que los planos de croma sean producto de un sub-sampleo, estos serán divididos también en bloques de 8x8 pixels, y en este caso, la única diferencia es que habrá menos bloques en los planos de croma que en los de luma. Los límites de ubicación espacial de los bloques de luma no necesariamente coincidirán con los de croma en el caso de que los planos de croma hayan sido sub-sampleados. En la **Figura 3a** se muestra el proceso de división explicado.
- Cada plano de color es también subdividido en arreglos de 2x2 bloques denominados macro-bloques, a diferencia de los super-bloques, un macro bloque representa 4 bloques del plano de luma más los bloques en igual posición en los planos de croma, esto implica que un macrobloque puede representar desde 6 hasta 12 bloques dependiendo de si los planos de croma han sido sub-muestreados o no. Como se estableció anteriormente, los superbloques contienen la información referente a sólo un único plano de color, en contraste, un macrobloque contiene información de los tres planos en una posición específica lo que es de especial utilidad para realizar las operaciones pertinentes a codificar la redundancia espacial. En general, un macrobloque contendrá la información relacionada con el modo de

codificación y los vectores de movimiento correspondiente a un bloque en cada plano de color. La **Figura 3b** muestra la división en macrobloques.

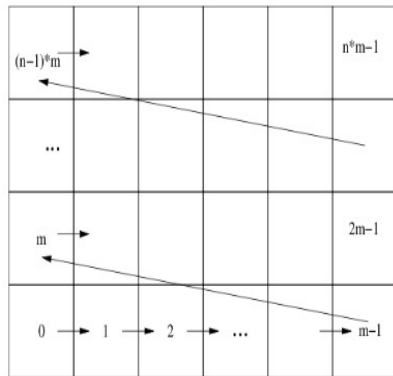


**Figura 3a, División del Frame, Bloques, Superbloques, [9]**

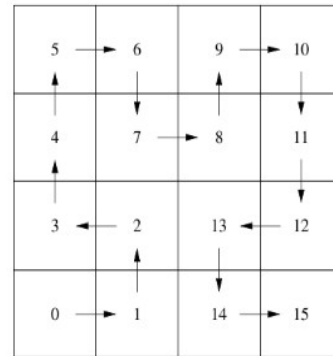


**Figura 3b, División del Frame, Macrobloques, bloques, [9]**

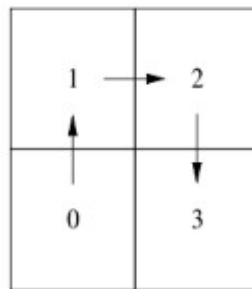
- Acceso a la información: La metodología para acceder (leer, calcular y escribir en memoria) la información contenida en los diferentes sub-grupos de almacenamiento dentro de un cuadro es uno de los factores que más varía en los diferentes códecs de video en busca de mejorar la eficiencia en la compresión y de disminuir la latencia. En Theora, los bloques son accedidos en formas diferentes de acuerdo al proceso específico que el *códec* esté ejecutando. El primer modo es de forma secuencial, comenzando con el elemento situado en la esquina inferior izquierda y siguiendo hacia la derecha hasta terminar la fila inferior, para luego, repetir el proceso con la fila inmediata superior hasta llegar al límite superior del cuadro y al último bloque ubicado en la esquina superior derecha. El segundo modo de acceder u ordenar los bloques es de acuerdo a un código asignado en relación a los superbloques en los que está contenido. En un *frame*, los super-bloques se ordenan de forma secuencial (similar al proceso descrito para los bloques), sin embargo, dentro de cada super-bloque, los bloques son accedidos en un patrón descrito por la curva de Hilbert. El índice de asignación de códigos definido por la curva Hilbert comienza en la esquina inferior izquierda del plano de luma, continua hasta recorrer todo el plano de luma y sigue aumentando (no retorna a un valor 0) al pasar a los planos de croma. Los macro-bloques son también accedidos en orden de acuerdo a un código asignado al acceder a cada super-bloque de manera secuencial pero codificando los cuatro macro-bloques dentro de cada super-bloque con una pequeña curva Hilbert. La **Figura 4** muestra el ordenamiento de los bloques (a), superbloques (b) y macrobloques (c).



(a)



(b)



(c)

**Figura 4:** Ordenamiento de Bloques, SuperBloques y MacroBloques, [9]

- **Codificación y Predicción:** Cada bloque es codificado utilizando uno de entre varios tipos de modos de codificación que definen como el bloque será predicho a partir de cuadros previos. Un bloque es predicho a partir de uno o dos cuadros de referencia elegidos de acuerdo al modo de codificación. Un cuadro de referencia es un cuadro completamente decodificado de un *frame* anterior en el *stream* de video. El primer cuadro disponible para servir de referencia será, evidentemente, el cuadro *Intra* anterior, el cual es denominado *golden-frame*. El segundo cuadro para servir como referencia es el inmediato anterior, sea este un cuadro *Intra* o *Inter*, si el cuadro inmediato anterior era una *Intra*, entonces ambos cuadros de referencia serán el mismo.

El proceso y modo de codificación de un bloque dependerá, como se dijo, de varios factores, en primer lugar, si el bloque pertenece a un cuadro *Intra*, entonces todos los bloques de ese cuadro en los tres planos de color serán etiquetados como Bloques-*Intra* y su modo de codificación será *Intra*. Si por el contrario, el bloque pertenece a un cuadro *Inter*, se efectúan una serie de lecturas y comparaciones para determinar entre 8 diferentes modos de codificación cuál es el más adecuado.

La selección del modo de codificación es uno de los factores mas importantes en el procesado de video. Entre los modos de codificación, vale la pena destacar:

1. *INTRA*: El modo *intra* se utiliza para codificar aquellos bloques que no han sido predichos basados en un cuadro de referencia, es el único modo disponible para los *INTRA-frames*, sin embargo y de acuerdo a las características de un trame en particular, puede ser utilizado también para codificar *INTER-frames*.
2. *INTER\_NOMV*: Este modo utiliza la información contenida en los bloques de igual posición que el actual pero del cuadro anterior para predecir el actual. El modo *INTER\_NOMV* es el modo por defecto para los cuadros *INTER*.

El resto de los modos incluirán la codificación predictiva, ya sea tomando como referencia el cuadro anterior o el *golden-frame*, sea de la misma posición o de una posición adyacente más un vector de movimiento. Toda la información referente al modo de codificación y a los vectores de movimiento son almacenadas y codificadas en el *stream* del *INTER-frame*.

- **Vectores de Movimiento**: La estimación de movimiento es uno de los procesos que más recursos demanda en la codificación (en Theora es responsable de casi la mitad de la latencia). Theora utiliza un algoritmo de búsqueda de coincidencias por zona (*Zonal Search*) para determinar los vectores de movimiento, tanto en cuadros anteriores como en bloques adyacentes en el cuadro actual. Los vectores de movimiento, una vez determinados, son

almacenados para cada macrobloque (solo en los *INTER-frames*) en los tres planos de color. La información referente a los vectores de movimiento se codifica conjuntamente con los datos del cuadro.

- Transformación y cuantificación: La transformación se realiza, como se mencionó, basada en las bondades de la Transformada Discreta del Coseno de tipo II, la cual toma como entrada una matriz de 8x8 pixels y retorna una matriz, también 8x8, de coeficientes.

Theora permite la utilización de hasta 384 diferentes matrices de cuantificación, dependiendo del tipo de codificación y cuantificador a utilizar y del índice de cuantificación ( $qi$ ). El índice de cuantificación ( $qi$ ) representa generalmente un rango de “calidad” o “compresión” que está acotado entre 0 y 63, donde 0 representa poca calidad y 63 alta calidad. Sin embargo, esta interpretación de  $qi$ , aunque se aproxima en la práctica, no corresponde exactamente al cálculo y resultado matemático.

- Codificación: Para realizar una codificación en base a la entropía, Theora hace uso de 80 códigos binarios configurables Huffman para representar las 32 muestras o símbolos usados para codificar los coeficientes de la DCT. Cada uno de los 32 símbolos representa desde un valor de coeficiente, una agrupación de ceros o una combinación de los anteriores, también se codifica con estos símbolos las marcas para definir el final de un bloque.
- Ordenamiento y Paquetización: Theora, en la práctica, utiliza 4 tipos de paquetes para almacenar la información:
  - Paquete de Identificación: Identifica el *stream* como Theora, provee información sobre el número de la versión y define las características del video (tamaño del cuadro, etc).
  - Paquete de Comentarios: Incluye los comentarios del usuario y del vendedor/creador de la aplicación que crea el *stream*.

- Paquete de Setup: Incluye la información sobre las opciones de configuración del *códec*, entre otras, se almacenan la totalidad de los juegos de matrices de cuantificación y el libro de códigos Huffman necesarios para decodificar los coeficientes de la DCT.
- Paquete de Video: se incluyen los datos codificados del *stream* de video.

Las especificaciones de Theora 1.0 incluyen una serie de paquetes “reservados” para futuras mejoras y aplicaciones, estos paquetes, por el momento, son ignorados en el proceso de codificación/decodificación.

## Comparación de las características técnicas de Theora 1.0 y H.264

La siguiente tabla (Tabla 2) presenta las principales características técnicas de los *códecs* H.264 y Theora 1.0

**Tabla 2:** Características Técnicas de H.264 y Theora 1.0, Autor.

<b>Característica</b>	<b>H.264</b>	<b>Theora</b>
Tipos de <i>frames</i>	I,P,B. . . SP,SI	I,P
Codificación Entrópica	UVLC CAVLC CABAC	SI, VLC, adaptativa
Resolución Vector de Movimiento	1/4pix	½ pix
Transformación	4x4, 8x8 DCT	8x8 DCT
Predicción espacial INTRA	si	si
Tamaño de macrobloque	Variable para la compensación de movimiento	fijo
<i>Profiles</i> /aplicación	SI, múltiples <i>profiles</i> y niveles	NO
Selección de matrices de cuantificación	Configurable, dinámico.	Selección por bloques
Soporte para codificación SIN Pérdida	SI	NO
Escalamiento/cambios de resolución, bitrate	Si, gracias a los cuadros de <i>switc</i>	NO, con la posibilidad de juntar <i>stream</i> distintos en un contenedor



## CAPITULO IV

### Transmisión de Video en redes IP

Como se mencionó con anterioridad, la codificación digital de video utiliza algoritmos de compresión capaces de adaptar un video analógico para ser transmitido en un ancho de banda limitado.

En redes IP, el transporte de video se realiza básicamente mediante los protocolos RTP y RTCP. El protocolo RTP establece los estándares y normativas para el transporte de información en “tiempo real” en la cual no se garantiza la calidad de servicio, establece la manera de generar paquetes que incluyan no solo la información (en el caso en estudio, el video en tiempo real), pero además información sobre el proceso de transmisión en sí (números de secuencia, monitores de entrega, marcas de tiempo, etc.). En general, RTP se asienta sobre protocolos de red de baja confiabilidad como el UDP.

El Proceso previo a la transmisión comienza con el encapsulamiento de las muestras codificadas de video en paquetes RTP, y cada paquete RTP es, posteriormente, encapsulado en un segmento UDP. RTP permite incluir, dentro de su encabezamiento, información referente al tipo de datos transportados.

El hecho de que los *códecs* en estudio utilicen codificación VLC y de que la codificación y compresión son procesos estadísticos, cuyos modelos no dependen solo del proceso de codificación sino de las características propias de las muestras de video a codificar, hace imposible predecir el ancho de banda necesario para transmitir un video en condiciones óptimas bajo redes IP. Por tanto, el ancho de banda dependerá del tipo de codificación utilizada, de la resolución de la muestra, del tipo de cuantización utilizada y de las características de movimiento y textura de la imagen. A todo lo anterior, es necesario agregar el consumo de ancho de banda utilizado por la red (paquetes IP, UDP y RTP, tramas *ethernet*, etc).

La transmisión de video sobre redes de paquetes presenta características determinadas, se utilizan rangos de anchos de banda variables, hay congestión y pérdida de paquetes. El contenido multimedia, y en especial el video, presenta alta susceptibilidad ante este tipo de problemas y la percepción visual en el receptor se ve gravemente afectada. Entre los factores específicos de las redes IP capaces de corromper la información de video vale la pena mencionar:

1. Pérdida de paquetes: Dado que la codificación de video se realiza de forma diferencial, la pérdida de un paquete no solo afecta la capacidad de decodificar la información contenida en el paquete perdido, afecta también el proceso de decodificación de los bloques siguientes en el cuadro, además, si el paquete perdido corresponde a un cuadro de referencia, el error tiende a esparcirse a los cuadros predichos a partir de él. A pesar de que los algoritmos poseen rutinas para eliminar estos efectos (como la inclusión de cuadros *INTER* redundantes) es evidente que éstas aumentan la complejidad del proceso de codificación y la cantidad de bits necesarios.
2. Efectos de demora (*Jitter*): El decodificador debe recibir los paquetes en un flujo constante para realizar el proceso de decodificación, para ello, y dado que los retardos son inevitables en las redes de paquetes, se dispone de un *buffer* en el cual los paquetes que van llegando son almacenados y despachados al algoritmo a un ritmo constante, sin embargo, la necesidad de éste *buffer* implica un retardo adicional y un aumento en la complejidad del receptor. Cuando las sumas de los retardos sobrepasan la capacidad de *buffer* y el decodificador no obtiene los datos necesarios, suelen producirse discontinuidades o saltos en el video decodificado. [5]

No existe, hasta el momento, un modelo que permita predecir de forma objetiva el efecto de estos y otros factores inherentes a las redes de paquetes en la calidad del video transmitido. Existen sin embargo antecedentes en el área del audio digital como el “*Modelo E*” (ITU-T G.107) que es capaz de, dentro de un margen de error, predecir la calidad percibida en base a los *códecs* utilizados y las características de la red IP.

## CAPITULO V

### Calidad de Video

#### 5.1 Degradación del Video Digital:

El proceso de codificación y compresión aplicado por los codificadores CON-pérdida, genera, sin duda alguna, cambios en la percepción visual respecto al video original. La transformación al dominio de la frecuencia (DCT) y posterior cuantificación, con el subsecuente descarte de componentes frecuenciales además de las aproximaciones derivadas de las técnicas de predicción y compensación de movimientos tienden a degenerar la señal de video en niveles perceptibles.

Los diferentes tipos de degradación traen como consecuencia diversos efectos en la percepción visual del video, y, a la vez, pueden ser definidos de manera relativamente independiente.

Entre los efectos de degradación mas notorios o perceptibles por el ojo humano podemos encontrar:

1. *Blocking* – Efecto de bloques - : El efecto de bloques tiene su origen en el proceso de dividir la imagen en unidades espaciales de menor tamaño (bloques, superbloques, etc) para la ejecución de la DCT. Se manifiesta como discontinuidades en los bordes entre bloques adyacentes al reconstruir la imagen. Dentro de un cuadro, el efecto de bloque es inversamente proporcional a la precisión en el proceso de cuantificación utilizado. En cuadros predictivos, el efecto se puede dar entre macro-bloques, presentado discontinuidades entre sus bordes, o dentro de un macro-bloque, entre los cuatro bloques que lo componen. Dentro de un mismo cuadro, cuanto más “gruesa” sea la cuantificación realizada, más visible es el efecto de bloques. En general, el efecto de bloque es menos percibido en imágenes con mucho movimiento y en cuadros con mucho o muy poco brillo, por el contrario, los

coeficientes de bajas frecuencias espaciales, son los que determinan en mayor grado la visibilidad del efecto de bloques. Como se mencionó, la percepción del efecto de bloques responde en muchos casos a las características propias de los cuadros a codificar, por tanto, no existe un umbral de cuantificación que permita determinar numéricamente un nivel mínimo para evitar este tipo de degradación.

2. Borrosidad o falta de definición (*Blurring*) : La falta de definición, se manifiesta como una pérdida de los detalles de la imagen. El efecto *Blurring* se introduce cuando se suprimen los coeficientes DCT de mayor orden, que son los que aportan los detalles finos dentro de sus bloques.
3. Corrimiento del color (*Color bleeding*): La falta de definición de la información de luminancia resulta en la difusión de los detalles en un área visible, por su parte, la poca definición en los valores de la crominancia, produce “manchas” de colores sobre áreas de colores contrastantes. Al igual que el caso anterior, este efecto se debe a la supresión de los coeficientes de alta frecuencia de los componentes de crominancia. Dado que la crominancia es sub-muestreada el efecto se propaga en todo el macrobloque.
4. Efecto escalera y *Ringing* : Cuando la imagen contiene bordes diagonales respecto a los ejes verticales u horizontales, se puede presentar el “efecto escalera”. Dado que las imágenes base de la DCT forman patrones horizontales y verticales, los bordes diagonales no siempre son bien representados cuando se utilizan técnicas de compresión que eliminan la información de alta frecuencia espacial. En los casos en los que la diagonal se distribuye entre varios bloques, se forma un patrón del tipo “escalera”; cuando las secciones adyacentes al borde tienen alto contraste, el efecto es especialmente notorio, se denomina “*Ringing*”.
5. Contornos falsos : El efecto de cuantificación de los valores de luminancia de los píxeles lleva a que se generen falsos contornos en los lugares de transición de un valor cuantizado a otro, especialmente en zonas de transiciones graduales. Este efecto es notorio cuando la cantidad de niveles de

cuantificación es insuficiente. Algo similar sucede con la cuantificación de los valores de DC y de baja frecuencia de los coeficientes de DCT. En este tipo de imágenes, el efecto se ve como saltos de luminancia o tonalidad en lugares donde debería haber una transición gradual.

6. Bordes falsos: Este efecto se presenta como consecuencia de transportar el efecto de bloques hacia cuadros predictivos, con compensación de movimiento. Si se produce el efecto de bloques en una imagen tomada como referencia para próximos cuadros, estas discontinuidades producidas entre los bloques pueden ser convertidas a bordes falsos dentro de bloques predictivos debido a la estimación de movimiento.
7. Error de Compensación de Movimiento (*MC Mismatch*): En el proceso de compensación de movimiento se compara la información de un macrobloque con posibles secciones de los cuadros siguientes (buscando típicamente un valor mínimo del MSE) y basados en la hipótesis de que todos los píxeles del macrobloque tendrán un desplazamiento similar (al pertenecer a la misma figura en la imagen). En determinados casos, cuando el macrobloque contiene imágenes de figuras “parecidas” pero diferentes, el proceso de compensación de movimientos puede verse comprometido, los vectores de movimiento elegidos pueden no ser los adecuados y podrán observarse efectos de bloques con altas componentes de frecuencia espacial en los bordes de las imágenes en movimiento.
8. Efecto mosquito: Se denomina “efecto mosquito” a la fluctuación de la luminancia o crominancia alrededor de áreas de alto contraste o de figuras en movimiento. Este efecto está relacionado con los efectos de “ringing” y de errores de compensación de movimiento ya vistos. Es consecuencia de tener diferentes codificaciones en cuadros consecutivos, para una misma sección de la imagen (las que pueden darse por cambios en el tipo de predicción, en los niveles de cuantificación, vectores de movimiento, etc.).
9. Otros errores: Entre otros errores, menos perceptibles, se encuentran: Fluctuaciones en áreas estacionarias, errores de crominancia, patrones de

mosaicos y el efecto de imagen de base. [5]

A pesar de que los diferentes codificadores tienen sub-rutinas especialmente diseñadas para disminuir los efectos de degradación, estos son objeto continuo de estudio del VJG. En el caso de la transmisión de video por redes de baja confiabilidad es necesario agregar a estas degradaciones los efectos introducidos y explicados con anterioridad inherentes a las redes de paquetes. La suma de todos los efectos en el proceso de digitalización – codificación – envío constituyen la percepción del video reconstruido.

## **5.2 Medicion de la Calidad de Video.**

El proceso de definir y cuantificar la “calidad” de un video codificado presenta un gran desafío a los desarrolladores de códigos de compresión. A diferencia de muchos otros procesos, la calidad de una muestra de video puede ser estimada desde dos puntos de vista cuyos resultados y conclusiones, en muchas ocasiones, no se corresponden. El primer impulso es, sin duda, establecer un patrón matemático que describa “qué tan diferente” es un video codificado de su referencia original, sin embargo, la complejidad del sistema visual humano hace prácticamente imposible que este “modelo matemático” describa la “percepción” que el usuario tiene sobre los cambios que el proceso de codificación introduce en el video.

Las primeras medidas objetivas de la calidad del video están basadas en obtener las diferencias, píxel a píxel, entre las imágenes originales (previo a la compresión y transmisión) y las imágenes presentadas (luego de la recepción y reconstrucción). Dado que los sistemas de video utilizan técnicas de compresión con pérdida de información, y que los medios de transmisión a su vez pueden introducir factores distorsionantes, las imágenes presentadas serán matemáticamente diferentes a las originales.

A la par del desarrollo de los modelos para la medición objetiva de la calidad, se han propuesto diversos modelos subjetivos cuya finalidad es el evaluar

estadísticamente el nivel de satisfacción del observador con relación a la comparación entre un video codificado y su correspondiente referencia.

Los modelos para determinar la calidad de video de manera objetiva han sido duramente criticados por su inconsistencia con la sensación de satisfacción (percepción) del observador, por su parte, los modelos subjetivos presentan serias dificultades para ser aplicados y cuantificados. La meta, sin duda alguna, es estandarizar un modelo capaz de conjugar la correlación matemática y las características perceptivas del proceso visual, que sea económico, práctico y fácil de aplicar.

En general, los modelos para determinar la calidad de video codificado pueden dividirse en 3 diferentes tipos: De referencia completa (*Full Reference* – RF -), de referencia reducida (*Reduced Reference* – RR-) y sin referencia (*No Referenced* – NR -). [5]

- *Full Reference*: Corresponden a esta categoría los algoritmos utilizados para medir la calidad de un video codificado cuando se dispone tanto del video codificado como del video original o referencia. Los métodos FR son ampliamente utilizados ya que permiten que la comparación (codificado/referencia) sea realizada utilizando todos los parámetros cuantificables, a la vez, son fácilmente programables y pueden ser evaluados con herramientas de computo.
- *Reduced Reference*: En los algoritmos RR, el emisor del video extrae del original ciertos parámetros para ser enviados conjuntamente con el video codificado (utilizando una pequeña porción del ancho de banda), el receptor extrae los mismos parámetros (del video codificado) y se realiza una comparación sobre estos parámetros extraídos. Los métodos RR, a pesar de recargar el ancho de banda y producir resultados no siempre exactos, son útiles para medir la calidad en grandes redes *multicast* en donde enviar el video de referencia quedaría absolutamente descartado.

- *No Reference*: Los modelos NR, intentan predecir la calidad subjetiva de una imagen o video basándose únicamente en el análisis de la señal recibida. Esta tarea, realizada en forma sencilla y natural por las personas, es sumamente difícil de realizar de manera automática. Las variaciones NR existentes consisten en realizar la estimación de calidad en base a las medidas de algún conjunto de “degradaciones tipo”, que en la mayoría de las situaciones afectan a la calidad percibida. En general, los algoritmos NR solo proveen información aproximada sobre parámetros de calidad.

### 5.3 Procedimientos para realizar la medición de la calidad de Video codificado.

Los modelos FR pueden ser catalogados en: Objetivos (aquellos que solo establecen la correlación matemática del video codificado y del original), subjetivos (los que se basan en la percepción del observador) y mixtos (aquellos ajustan las métricas de la correlación introduciendo factores específicos del proceso visual).

#### 5.3.1 Métodos Objetivos.

Las medidas más simples son las de error cuadrático medio (MSE - *Mean Square Error*) y su raíz cuadrada (RMSE - *Root Mean Square Error*) y la relación señal a ruido pico (PSNR – *Peak Signal to Noise Ratio*), definidas en las Ecuaciones 1.1 a 1.3 . Estas métricas son del tipo FR (*Full Reference*), ya que requieren de la referencia completa para poder ser calculadas . [10]

$$MSE = \frac{1}{TMN} \sum_{n=1}^N \sum_{m=1}^M \sum_{t=1}^T [x(m,n,t) - y(m,n,t)]^2 \quad \text{Ecuación 5.1, MSE, [10]}$$



$$RMSE = \sqrt{MSE}$$

**Ecuación 5.2** , RMSE, [10]

$$PSNR = 10 \log_{10} \left( \frac{L^2}{MSE} \right)$$

**Ecuación 5.3**, PSNR, [10]

Donde la imagen tiene  $N \times M$  píxeles y  $T$  cuadros,  $x$ ,  $y$  son los píxeles de la imagen original y la distorsionada respectivamente.  $L$  es el rango dinámico que pueden tomar los valores de  $x$  o  $y$ , y toma el valor 255 para 8 bits por píxel.

Estas métricas son fáciles de calcular, y tienen un claro significado. Por estas razones, han sido ampliamente usadas en la estimación de la calidad de video. Es necesario enfatizar la importancia de la alineación espacial y temporal de las imágenes a comparar, ya que la referencia y la imagen degradada pueden estar desfasadas en el tiempo o en el espacio.

Evidentemente, un valor alto del PSNR implicará una “alta calidad” de video, teóricamente, el PSNR pudiera tomar valores infinitos, sin embargo, y dado que los algoritmos de compresión en estudio son por definición “CON-pérdida”, siempre existirá una degradación en la codificación y, por tanto, un PSNR finito.

### **5.3.2 Métodos Subjetivos.**

El principal objetivo de los métodos subjetivos es el de recopilar de manera “automática” la opinión de el observador en cuanto a la relación entre un video codificado y su referencia con el fin de determinar el nivel de “satisfacción” entregado por el proceso de codificación. En general, la idea es similar al método *Mean Opinion Score* (que busca determinar la “calidad percibida” en audio digital).

Las recomendaciones para la utilización de métodos subjetivos en el área de video se encuentran descritas por la ITU-T (ITU-T/BT.500). La metodología típica consiste en mostrar secuencias de video original seguidas de las secuencias

codificadas y pedirle al usuario que de su opinión en base a escalas preestablecidas. Los métodos subjetivos, al igual que la precisión del proceso de visión, son puramente estadísticos y sus resultados pueden variar en relación a parámetros no relacionados con el proceso de codificación (edad, raza, nivel de experticia en el área de video digital, etc.), sin embargo, y para una población de muestra determinada, son sumamente útiles para determinar los niveles de satisfacción del cliente ante un producto determinado que incluya la codificación de video.

En la práctica, los métodos subjetivos se diferencian, principalmente, en la forma en la que son expuestas las muestras al observador, y en como se realiza la recolección de datos. La ITU-T BT.500-11 establece pautas y recomendaciones para realizar los procedimientos, entre otras, las condiciones del ambiente (iluminación, espacio físico), las características y configuraciones de los equipos utilizados para la reproducción (resolución, contraste, distancia al observador) y las características de la muestra de video (resolución, tamaño, duración) entre otros.

Entre los métodos subjetivos más utilizados se encuentra el DSIS (*Double Stimulus Impairment Scale* - ITU-R BT.500-11 ) que básicamente consiste en presentar a un grupo de sujetos de prueba, bajo condiciones establecidas, una secuencia original (referencia) y seguidamente la secuencia codificada, una vez que los sujetos observen ambas imágenes, deben proceder a votar según el grado de “imparidad” o diferencia entre las secuencias mostradas en una escala pre-establecida. [11]

Escala de DSIS:

- 5 Imperceptible
- 4 Perceptible, pero NO molesto
- 3 Ligeramente molesto
- 2 Molesto
- 1 Muy Molesto

Evidentemente, una puntuación mayor implica que la degradación perceptual de la secuencia de video al ser codificada fue menor, por tanto, el video codificado será de “mayor calidad” según este método.

### **5.3.3 Métodos Mixtos.**

Debido a la baja correlación entre los resultados de los métodos puramente objetivos y los puramente subjetivos, en los últimos tiempos, se ha realizado un gran esfuerzo para desarrollar nuevos modelos que tengan en cuenta las características de percepción del sistema visual humano y que permitan calcular métricas objetivas que lo simulen. El principal problema de estudio para el desarrollo de los métodos mixtos esta en encontrar el “cómo” y “qué tanto” afectan los diferentes parámetros del sistema de percepción visual a los resultados de los métodos “puramente” matemáticos.

Los métodos Mixtos tienen, en general, una estructura similar: En principio las señales (original y codificada) son pre-procesadas (alineadas espacial y temporalmente, etc.) y luego descompuestas en sus respectivos planos de color (por ejemplo YUV en Y,Cr,Cb) sobre los cuales se calcula un “error” matemático según alguno de los métodos objetivos. Finalmente, los resultados de los métodos objetivos son “corregidos” de acuerdo a parámetros obtenidos de la caracterización del modelo de visión, para luego ser totalizados obteniendo así una única medida de “calidad”.

Para que un método mixto pueda ser considerado un estándar debe tener ciertas características:

1. Consistencia: el modelo debe mantener cierto nivel de precisión para diversos tipos de muestras (con poco o mucho movimiento, con imágenes naturales o animadas, y bajo diferentes niveles de compresión ).
2. Precisión: los datos interpolados del modelo deben estar dentro de un rango determinado de error respecto a los reales.
3. Universalidad: Los datos obtenidos deben representar lo mismo para

diferentes sujetos, esto es, deben ser independientes de factores humanos no relacionados con los procesos de codificación.

4. Factibilidad: la capacidad de procesamiento, memoria y recursos requeridos ha de hacer accesible a los usuarios la utilización del modelo.

Entre los modelos que han sido propuestos como candidatos a ser estándares, pueden encontrarse:

- *British Telecom BTFR* : El algoritmo BTFR (*British Telecom Full Reference*) consiste en una detección seguida de una integración. La detección supone el cálculo de un conjunto de parámetros perceptuales significativos del detector, a partir de secuencias de video sin distorsionar (de referencia) y distorsionadas (degradadas). Estos parámetros se introducen a continuación en el integrador que produce una estimación de la calidad de video percibida mediante la ponderación de cada uno de los parámetros. La selección de detectores y factores de ponderación se basa en el conocimiento de las propiedades de enmascaramiento espacial y temporal del sistema de visión humano. [10]
- *Yonsei University/Radio Research Laboratory/SK Telecom*: El modelo propuesto por *Yonsei University* se basa en la observación de que el sistema visual humano es altamente sensible a la degradación en torno a los bordes. De acuerdo con esta observación, el modelo proporciona un método de medición objetiva de la calidad de video que mide la degradación en torno a los bordes. En este modelo, se aplica en primer lugar un algoritmo de detección de bordes a la secuencia de video de referencia para localizar las zonas de los bordes. A continuación, se mide la degradación de dichas zonas de bordes en el video codificado calculando el error cuadrático medio. [10]
- *National Telecommunications and Information Administration*: Este modelo tiene parámetros objetivos para medir los efectos perceptuales de una amplia gama de degradaciones tales como la borrosidad, el efecto de bloques, el

movimiento entrecortado o poco natural, el ruido (tanto en la luminancia como en la crominancia), y los bloques con errores (por ejemplo, lo que podría verse normalmente cuando hay errores en la transmisión digital) . Dentro de este modelo, una “característica de calidad” se define como la cantidad de información asociada a una subregión espacio-temporal de una secuencia de video (original o procesada). Comparando las características extraídas del video procesado (degradado) con las extraídas del video original, puede calcularse un conjunto de parámetros de calidad, que son indicativos de las variaciones perceptuales de la calidad de video. [10]

- SSIM (*Structural Similarity*): Se basa en que “la función principal de los ojos humanos es extraer información estructural del campo visual, y el sistema visual humano está altamente adaptado para este propósito. Por lo tanto, una medida de las distorsiones estructurales debería ser una buena aproximación a la distorsión percibida”. El modelo SSIM establece que la similitud entre dos imágenes depende de la similitud entre sus luminancias, contrastes y estructuras. La similitud general ( $S$ ) entre una imagen  $x$  y una imagen  $y$  es, según este modelo, una función ponderada de las similitudes de estas tres componentes (Ecuación 2.1). [10]

$$S(x, y) = f(l(x, y), c(x, y), s(x, y)) \quad \text{Ecuación 5.4, Similitud, [10]}$$

donde  $l$ ,  $c$  y  $s$  representan las funciones de similitud de luminancia, contraste y estructura respectivamente. El modelo SSIM, luego de diversas consideraciones y aproximaciones, establece:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad \text{Ecuación 5.5, SSIM, [10]}$$

Siendo  $\mu$  valores medios,  $\sigma$  varianza o covarianza, y  $C1$  y  $C2$  dos variables de ajuste. Este índice puede ser calculado en ventanas móviles de, por ejemplo, 8 x 8 píxeles, obteniendo un mapa de valores SSIM, los que luego pueden ser promediados para obtener el índice de la imagen completa (MSSIM). Si se conocen las áreas de la imagen de fijación de la vista, se pueden ponderar los índices en esta zona, de manera que pesen más en la suma total. Como es de esperarse, un valor de SSIM=1 implica la similaridad absoluta entre los videos referencia-codificado (caso imposible en codificación CON pérdida), sin embargo, mientras más cercano se encuentre a este valor, será indicativo de “mejor calidad”

## CAPITULO VI

### Pruebas de Desempeño

#### 6.1 Consideraciones Iniciales.

Establecer una metodología óptima para la medición de la calidad de un *códec* es una tarea difícil, intereses económicos y dificultades tecnológicas han hecho imposible que hasta la fecha se establezca una metodología estándar para realizar dichas mediciones. Con la finalidad de implementar una serie de pruebas comparativas adecuadas, y, ante la ausencia de una “Guía de Procedimientos”, la existencia de un gran número de algoritmos de medición y el constante desarrollo de nuevos codificadores, es necesario mencionar una serie de consideraciones previas, generales y relacionadas más con las características de los procesos de codificación que con el codificador o el algoritmo de medición:

1. **Cada codificador ofrecerá diferentes “calidades” para diferentes cuadros de una misma secuencia de video:** Diversos factores pueden ser considerados como causantes de estas diferencias, en principio, cada codificador cuenta con sus propios mecanismos de control de la tasa de bits, lo que evidentemente produce fluctuaciones en la calidad de la muestra codificada y ya que estas fluctuaciones se relacionan con el proceso mismo de codificación, para cada *códec*, según sus características, serán diferentes. Por otra parte, es necesario recordar que los procesos de codificación en estudio utilizan cuadros “clave” cuyo proceso de compresión y elección puede ser diferente en cada *códec*, por consiguiente, la elección de estos cuadros será distinta en cada algoritmo de compresión y su efecto en la predicción de cuadros posteriores afectará directamente la calidad de la muestra.

2. **Diferentes codificadores han sido diseñados, ajustados y optimizados considerando diferentes aplicaciones:** A pesar de que los desarrolladores buscan establecer un estándar, y crear un algoritmo compresor capaz de ser adaptado a todo tipo de aplicaciones, es innegable que, a pesar de que los codificadores “funcionen” correctamente en una amplia gama de ellas, algunos permitirán implementar diferentes soluciones multimedia en forma más eficiente que otros. Algunos codificadores, por tanto, estarán ajustados para trabajar con mayor eficiencia en muestras de video con *bit-rates* altos y otros con niveles bajos, algunos para muestras con mucho movimiento y otros que no lo hagan tan eficientemente; en fin, a pesar de que las configuraciones de los parámetros de codificación sean iguales, y que el tamaño del archivo codificado sea igual, la calidad del resultado de la codificación variará también de acuerdo a la aplicación en la cual se implemente.
3. **La calidad del proceso de codificación dependerá directamente de la configuración de los parámetros del codificador:** Es necesario considerar que, aunque existen una serie de parámetros generales, configurables por el usuario y que son comunes en la mayoría de los codificadores (como el *bite-rate* o el número de cuadros por segundo), existen también otros parámetros que el usuario del *códec* puede variar y que dependerán de cada algoritmo. Estos parámetros no necesariamente tienen una implicación directa con la calidad del video codificado, sin embargo, la optimización del conjunto completo de parámetros, considerando el tipo de muestra a codificar y la aplicación a utilizar, será determinante en la calidad del resultado.

## 6.2 Parametros a considerar.

Como se refirió con anterioridad, la elección de un codificador dependerá en gran medida de la aplicación en la cual se pretenda implementar. Bajo la premisa de realizar una comparación general que sirva como referencia para ensayos más



especializados de acuerdo al tipo de aplicación y de muestras de video, se determinó un conjunto de parámetros específicos que representan las posibles soluciones que incluyen el codificado de video. Los parámetros considerados fueron los siguientes:

1. Gama de calidades ofrecidas por los codificadores.
2. Relación “calidad de video” Vs. “Tamaño del archivo codificado” (Calidad Vs. Compresión).
3. Impacto de los efectos inherentes a las redes de paquetes en la calidad de muestras de video codificadas enviadas a través de ellas.
4. Impacto de características únicas y propias de H264 en la calidad de video codificado frente a la calidad ofrecida por Theora 1.0.
5. Consumo medio de recursos utilizados por los *códecs* durante el proceso de codificación.
6. Compatibilidad con las diversas plataformas de hardware y software existentes en el mercado.

### **6.3 Elección de la métrica y de las muestras.**

En capítulos anteriores se hizo referencia a los diferentes tipos de procedimientos destinados a medir la calidad de un video codificado. De igual forma se categorizaron los procedimientos en tres categorías de acuerdo a los factores que utilizan para evaluar el resultado (objetivos, subjetivos y mixtos). Con el fin de realizar una comparación lo mas completa posible entre los *códecs* en estudio se procedió a elegir una metodología que incluyera un procedimiento de cada categoría. Es así como, en base a la factibilidad de realizar el procedimiento de medición, a las características generales de las muestras de video seleccionadas y al tipo de algoritmo, se tomó la decisión de utilizar en el estudio comparativo 3 de los métodos más generales y representativos: La medición del PSNR (Objetivo), DSIS (*Double Stimulus Impairment Scale* - ITU-R BT.500-11 ) (subjetivo) y SSIM (mixto).

Para la realización de las distintas mediciones se seleccionaron una serie de muestras de video en distintas resoluciones, desde QCIF hasta HD720 cuyas características se muestran en la **Tabla 3**.

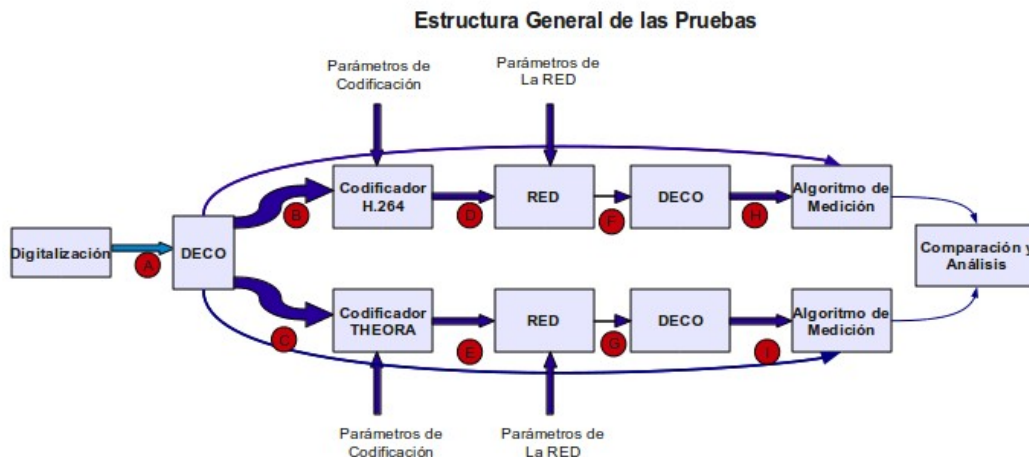
**Tabla 3:** Secuencias de Prueba, Autor

<b>Secuencia</b>	<b>Resolución</b>	<b>Características</b>
<i>Coastguard</i>	QCIF (176x144)	Cantidad de movimiento medio, bajo nivel de detalle espacial.
	CIF (352x288)	
<i>Silent</i>	QCIF (176x144)	Cantidad de movimiento medio, nivel de detalle espacial medio
	CIF (352x288)	
<i>Mobile</i>	CIF (352x288)	Cantidad de movimiento medio, nivel de detalle espacial medio.
<i>Mother and Daughter</i>	QCIF (176x144)	Cantidad de movimiento bajo, bajo nivel de detalle.
	CIF (352x288)	
Delfines	4CIF (704x576)	Cantidad de movimiento alto, nivel de detalle espacial alto.
	VGA (640x480)	
	SVGA (800x600)	
	HD720 (1280x720)	

#### 6.4 Estructura general de las pruebas

A pesar de que los algoritmos de medición son completamente diferentes, el procedimiento general para realizar las pruebas presenta una estructura esquemática similar. La Figura 5 representa la estructura general de los procedimientos para aplicar los algoritmos de medición.

**Figura 5, Estructura de las Pruebas, Autor**



- **A** El primer paso es, sin duda, la digitalización del video (en el caso de que la muestra de video sea analógica o que la muestra se obtenga en-vivo). El proceso de digitalización introduce degradación en la calidad del video, sin embargo, estas no son parte del estudio de este trabajo.
- **AB,AC y FH,GI** . A pesar de que se están comparando videos codificados, los algoritmos de medición de calidad realizan comparaciones *pixel-pixel*; las aplicaciones en el mercado realizan una “decodificación” de las muestras de video (original y codificada) para convertirlos “sin-pérdida” en formatos intermedios representados en espacio YUV. El proceso de “decodificación” realizado por las aplicaciones existentes, aunque adecuado en la mayoría de los casos, presenta dos inconvenientes principales: 1) Es imposible determinar si el “decodificado” introduce variaciones en el video y si además estas variaciones dependen del *códec* utilizado para decodificar. 2) La capacidad de decodificar un video depende estrictamente de que la aplicación soporte el *códec*, de la implementación del *códec* dentro del programa (los *códec*, aunque tengan funciones de transferencia “iguales” pueden ser implementados de forma distinta con el fin de optimizar el proceso o ajustar los resultados

según el tipo de media). Para eliminar los inconvenientes planteados, es evidente la necesidad de un decodificador “DECO” independiente del algoritmo de medición y cuya implementación sea la misma para los diferentes *códecs*. Estos DECO están representados en la Figura 5 entre los puntos A-B, A-C (decodificado del video original) y F-H, G-I (decodificado de los videos codificados con los *códecs* en estudio).

- BD, CE representan los *códecs* en estudio, reciben un volcado de video en formato intermedio YUV y los parámetros de codificación a utilizar, procesan, codifican y empaquetan los videos en sus respectivos formatos.
- DF, EG. Estos bloques representan a la red de transmisión en la cual se envía el video. Cuando se realizan las pruebas “sin-transmisión” este bloque puede ser entendido como “localhost”; no existe la posibilidad de que la RED-”localhost” introduzca degradaciones en el video. Al realizar las pruebas de transmisión, los parámetros de entrada comprenderán, en general, los parámetros de configuración y estado de la red de datos.

## 6.5 Pruebas de Calidad de Video Codificado

### 6.5.1 Calidad ofrecida por los codificadores.

Para la realización de esta prueba se procedió a medir la calidad de muestras codificadas con H264 y con Theora; los parámetros de los codificadores fueron ajustados como se indica en la **Tabla 4**.

**Tabla 4:** Parámetros de los Codificadores, Autor

<b>códec</b>	<b>Cuadros por segundo</b>	<b>Rata de bits</b>	<b>Observaciones</b>
H.264	30	800Kbps	Perfil Principal
THEORA	30	800Kbps	Calidad de codificación 5 ( <i>default</i> )

### 6.5.1.1 Medición del PSNR.

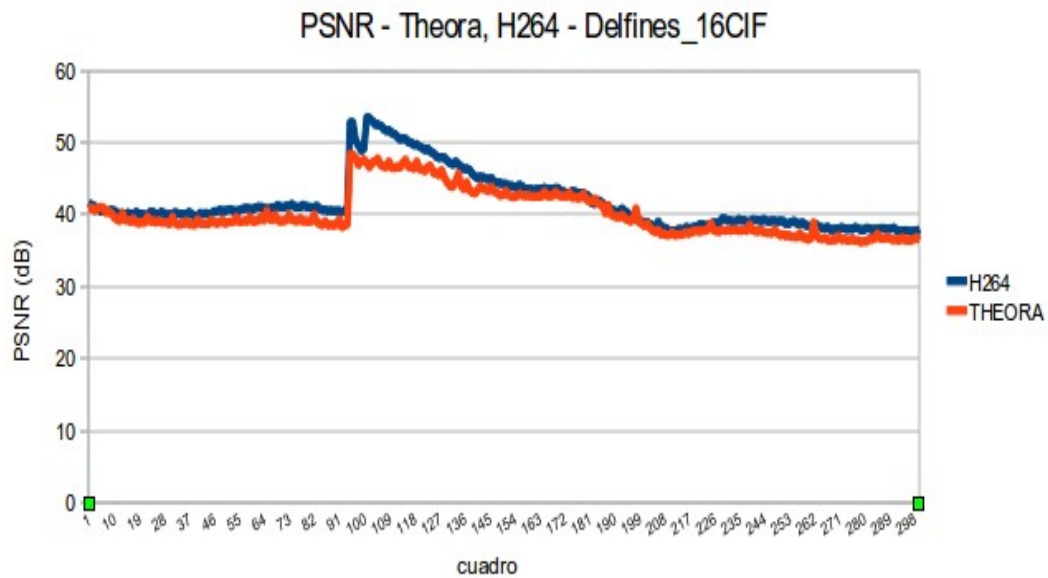
Los resultados de PSNR promedio obtenidos para las muestras codificadas se muestran en la **Tabla 5**.

**Tabla 5: Resultados del PSNR promedio**

<b>Muestra</b>	<b>Resolucion</b>	<b>PSNR H264 (dB)</b>	<b>PSNR Theora (dB)</b>
<i><b>Coastguard</b></i>	QCIF	20,37	20,5
	CIF	33,32	32,31
<i><b>Silent</b></i>	QCIF	24,35	24,14
	CIF	28,38	27,96
<i><b>Mobile</b></i>	CIF	16,78	16,89
<i><b>Mother and Daughter</b></i>	QCIF	27,14	26,7
	CIF	43,59	41,31
<b>Delfines</b>	VGA	42,38	31,58
	4CIF	41,04	39,45
	HD720	40,02	32,33

En el siguiente gráfico (Gráfica 1) se muestran las variaciones de la medida del PSNR a lo largo del flujo de video, el gráfico representa los valores de PSNR medidos para la muestra de video Delfines\_4CIF codificado con Theora y con H264. (gráficas para las distintas muestras pueden encontrarse en los anexos de este informe)

**Grafica 1,** “PSNR – Theora Vs. H.264 – Delfines\_16CIF, Autor



### 6.5.1.2 Escala de Imparidad de Doble Estimulo (*Double Stimulus Impairment Scale, DSIS*)

Para realizar las mediciones del DSIS se contó con la colaboración de 15 voluntarios quienes fueron expuestos a diferentes muestras de video (original y codificado) siguiendo, en la medida de lo posible, las recomendaciones de la ITU-R BT.500-11. Los resultados (promedio) obtenidos en la prueba se representan en la Tabla 6.

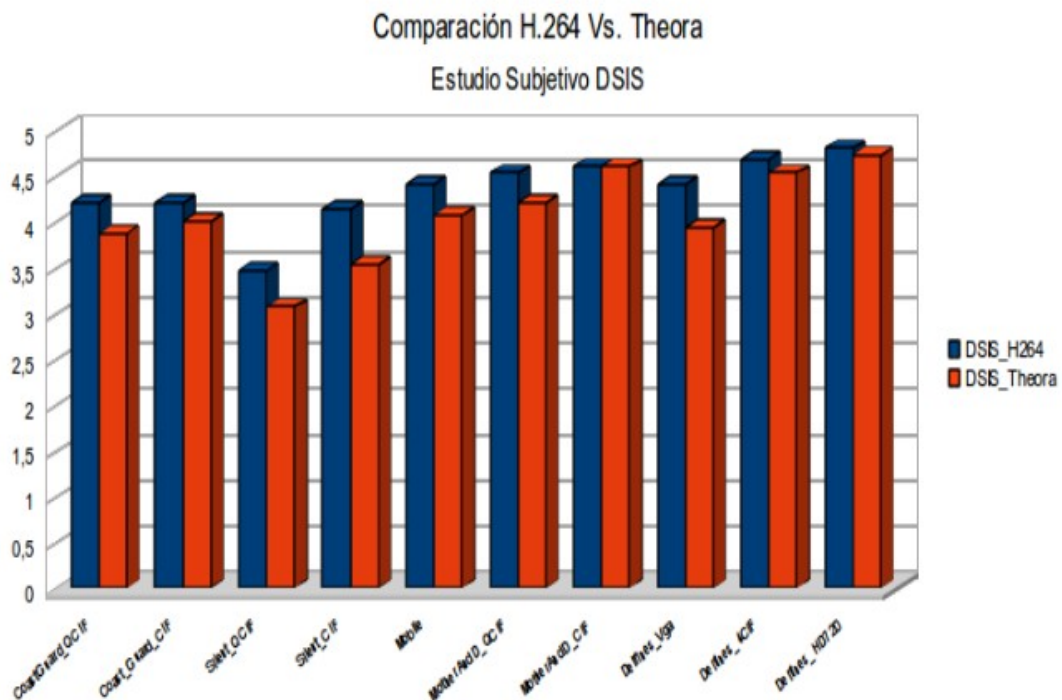
**Tabla 6: Resultados de la medición del DSIS**

Muestra	Resolución	Puntuación H264	Puntuación Theora
<i>Coastguard</i>	QCIF	4,2	3,87
	CIF	4,2	4
<i>Silent</i>	QCIF	3,47	3,07
	CIF	4,13	3,53
<i>Mobile</i>	CIF	4,4	4,07

Muestra	Resolución	Puntuación H264	Puntuación Theora
<i>Mother and Daughter</i>	QCIF	4,53	4,2
	CIF	4,6	4,6
Delfines	VGA	4,4	3,93
	4CIF	4,67	4,53
	HD720	4,8	4,73

La **Gráfica 2** permite visualizar el resultado del desempeño de los códecs en las pruebas subjetivas.

**Gráfica 2**, H.264 Vs. Theora – DSIS, Autor



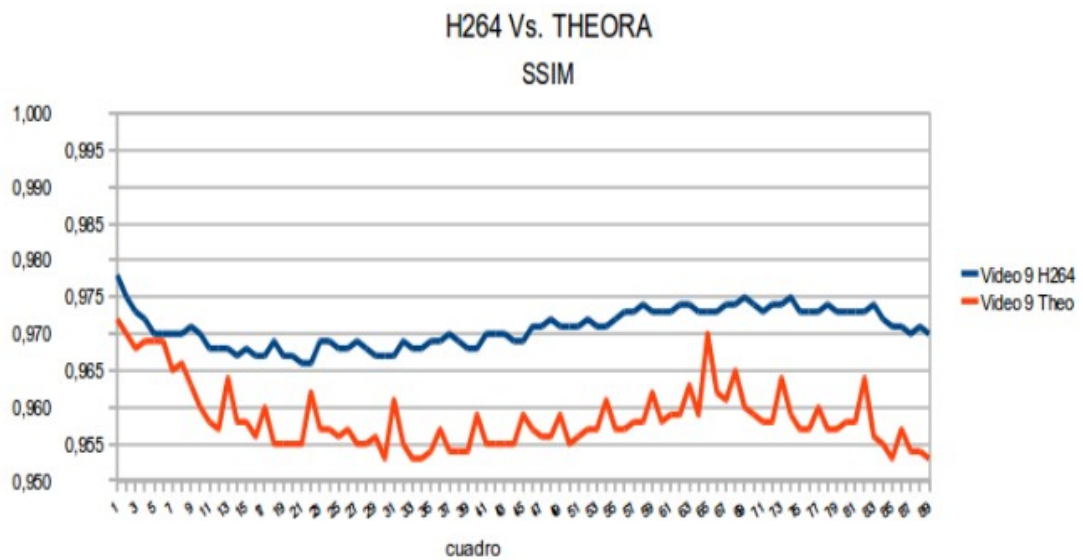
### 6.5.1.3 Medición del SSIM.

La **Tabla 7** y la **Gráfica 3** representan los resultados promedios del Índice de Similitud SSIM y las variaciones del SSIM por *frame* a lo largo del flujo de video para la muestra Delfines\_4CIF respectivamente.

**TABLA 7: Valores promedio del SSIM**

<b>Muestra</b>	<b>Resolucion</b>	<b>H264</b>	<b>Theora</b>
<i>Coastguard</i>	QCIF	0,568	0,570
	CIF	0,928	0,908
<i>Silent</i>	QCIF	0,810	0,804
	CIF	0,895	0,868
<i>Mobile</i>	CIF	0,497	0,497
<i>Mother and Daughter</i>	QCIF	0,761	0,759
	CIF	0,983	0,971
<b>Delfines</b>	VGA	0,980	0,852
	4CIF	0,975	0,961
	HD720	0,523	0,534

**Gráfica 3: Variaciones del SSIM para H.264 y Theora**





Observaciones:

- Los resultados de las pruebas (en general) son consecuentes entre sí, esto es, cuando se obtienen PSNR “altos” en H.264 también se obtienen valores “altos” en Theora y viceversa, incluso, puede verse en la gráfica 1 que las variaciones del PSNR a lo largo del flujo de video siguen un mismo patrón (similar para los dos códecS). De igual forma sucede con los valores del SSIM para ambos casos (H.264 y Theora) y en todas las secuencias.
- Los resultados de las pruebas de calidad establecen, en principio y bajo parametros de codificación estándar que:
  - Ambos *códecS* son capaces de codificar muestras de video en diversas resoluciones entregando resultados de calidad que van de “ACEPTABLE” a “EXCELENTE”.
  - Los resultados obtenidos en la medición de calidad de muestras codificadas con H264, bajo las condiciones descritas, son ligeramente superiores a los de Theora según los tres algoritmos de medición utilizados.
  - Los valores obtenidos, en base a los 3 algoritmos de medición, “mejoran” a medida que aumenta la resolución de la muestra.

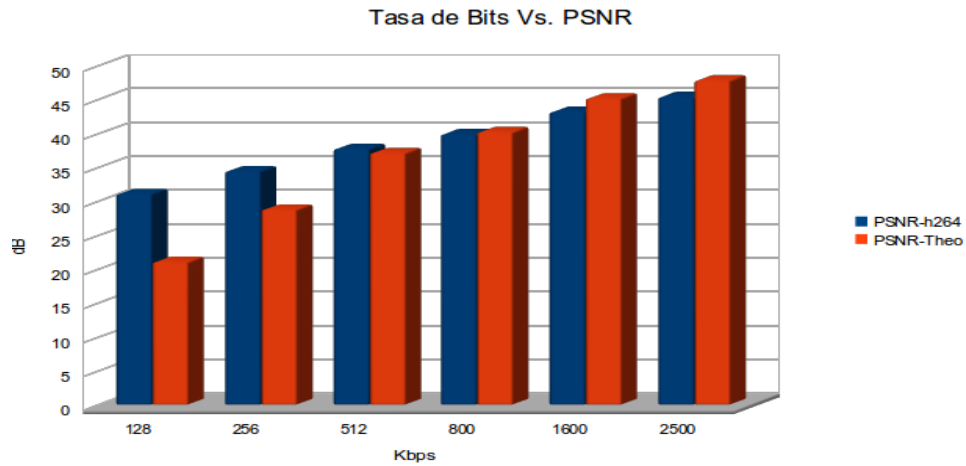
### **6.5.2 Relacion Compresion-Bitrate Vs Calidad.**

Para establecer una relación entre la capacidad de compresión del codificador y la calidad del video codificado se fijaron tasas de bits estándar, se realizó la codificación con los *códecS* H.264 y Theora 1.0 a esas tasas de bits y se comparó cada video codificado con el video de referencia según los procedimientos de cada algoritmo de medición. Se seleccionaron por su factibilidad los métodos de la Similaridad Estructural (SSIM) y la medición del PSNR.

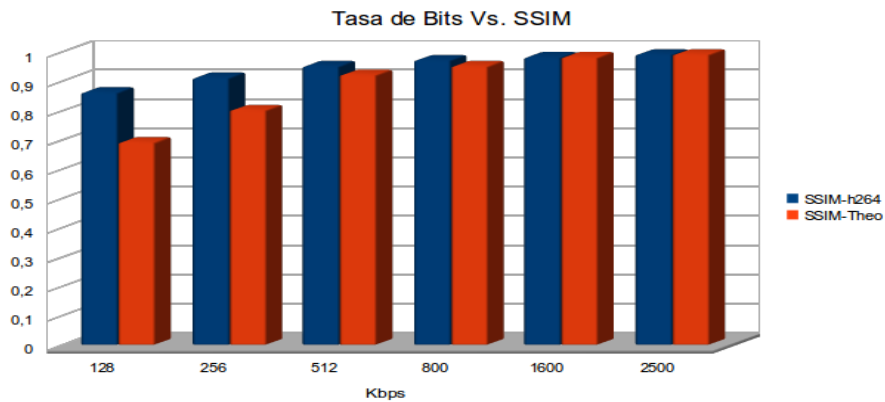
Las gráficas 4 y 5 representan respectivamente las medidas de PSNR y SSIM

de una muestra codificada (delfines\_4CIF) con H264 y con Theora a diferentes tasas de bits.

**Gráfica 4, Tasa de Bits Vs. PSNR, Autor**



**Gráfica 5, Tasa de Bits Vs. SSIM, Autor**



**Observaciones:**

- El códec H.264 mantiene una calidad relativamente constante para todas las tasas de bits, por su parte, Theora aparenta ser mas susceptible a disminuir la calidad que puede entregar a bajas tasas.

- Contrario a lo que se esperaría según el estudio realizado, Theora, a pesar de ser un *códec* diseñado con la idea fundamental de transmitir información por redes de ancho de banda limitado, presenta un desempeño pobre en *bit-rates* bajas y un desempeño alto (incluso mejor que el de H.264 en estas condiciones de codificación) para *bit-rates* altos.

### **6.5.3 Efecto de las características de la transmisión de datos en redes de paquetes en la calidad del video codificado.**

Para realizar el estudio del efecto de la transmisión sobre redes IP en la calidad de los videos codificados se procedió de dos diferentes maneras:

**6.5.3.1** Se simuló el envío de videos en un ambiente en donde pudiera controlarse el BER del proceso de transmisión.

**6.5.3.2** Se implementó un servidor de media en tiempo real bajo protocolos UDP/RTP, se enviaron muestras de video codificado con H.264 y THEORA bajo diferentes condiciones de la red y se midió la calidad del proceso de codificación comparando el video enviado con el recibido.

#### **6.5.3.1.**

Para introducir bits erróneos en los videos codificados se utilizó la herramienta *EvalVid* (especificaciones en los anexos). Muestras de videos codificados tanto con Theora como con H264 fueron alteradas considerando un modelo de *Ruido Blanco Gaussiano Aditivo* (AWGN). Los videos alterados y sus respectivas referencias fueron sometidos a las pruebas de PSNR y SSIM. La **Tabla 8** muestra los valores obtenidos para diversos BER introducidos.

**Tabla 8 :PSNR, SSIM Vs. BER, Autor**

BER	PSNR		SSIM	
	h.264	Theora	h.264	Theora
<b>0</b>	41,04	39,45	0,98	0,96
<b>1x10<sup>-6</sup></b>	41,04	39,45	0,98	0,96
<b>1x10<sup>-5</sup></b>	41,02	39,33	0,96	0,92
<b>1x10<sup>-4</sup></b>	38,8	21,72	0,93	0,76
<b>1x10<sup>-3</sup></b>	22,11	12,33	0,88	0,3

#### 6.5.3.2.

Para implementar el servidor RTSP se utilizó el programa Video Lan (VLC) cuyas especificaciones se encuentran descritas en los anexos de este trabajo. Las variaciones en la red se realizaron principalmente modificando el ancho de banda disponible para el equipo receptor con el programa *pyshaper*. Para cada flujo de video se realizó la siguiente secuencia:

1. Se midió el ancho de banda disponible para el enlace.
2. El video original (en formato intermedio YUV) se envió bajo protocolo RTP siendo transcodificado con el *códec* respectivo.
3. En el receptor se recibió el video y se realizó la decodificación a formato intermedio.
4. Se envió la muestra original al receptor (sin degradación) y se comparó con la muestra previamente decodificada.
5. Se alteró el ancho de banda disponible para el enlace y se realizó la prueba nuevamente desde el punto 1.

La muestra utilizada fue Delfines\_4CIF.

Los resultados obtenidos se muestran en la siguiente tabla:

**Tabla 9: PSNR y SSIM Ante variaciones del Ancho de Banda disponible.**

Ancho de Banda	PSNR		SSIM	
	H.264	Theora	H.264	Theora
<b>1.67Mbits/s</b>	40.03	39.32	0,98	0,96
<b>1.01 Mbits/s</b>	41,04	39,19	0,98	0,96
<b>448Kbits/s</b>	28,23	29,65	0,77	0,72
<b>238Kbits/s</b>	28,02	27,13	0,68	0,53
<b>132Kbits/s</b>	-	-	-	-

**Observaciones:**

Uno de los mayores problemas que presentan los métodos objetivos es la necesidad de que los flujos de video (original y codificado) estén completamente sincronizados en tiempo, esto es, que para cada cuadro del video original, corresponda el mismo cuadro del video decodificado y, por tanto, pueda realizarse la métrica respectiva (cuadro-cuadro, *pixel-pixel*). Desafortunadamente, al transmitir los datos por redes de paquetes, es frecuente que debido al limitado ancho de banda de la red, la capacidad del *buffer* del receptor y la latencia del proceso codificación-decodificación, entre otros factores, se pierda la sincronía.

Uno de los fenómenos que se presenta debido a la falta de sincronía, tanto en H.264 como en Theora, es la pérdida de cuadros al decodificar. En el caso de las pruebas realizadas, y para anchos de banda menores a 1Mbits/s, se decodificaron “menos” cuadros que los enviados. Este fenómeno “visualmente” puede no ser determinante, pero en definitiva, pone en duda los valores de la métrica del PSNR y el SSIM. Por las razones expuestas, se tomó la decisión de realizar pruebas de tipo subjetivo (DSIS) para determinar, al menos desde el punto de vista perceptual, la posible degradación de videos codificados con los códecs en estudio en redes de paquetes.

Los resultados de las pruebas de DSIS para videos codificados con H.264 y Theora 1.0, y transmitidos sobre protocolo RTP, se muestran en la **Tabla 10**:

**Tabla 10: DSIS Ante variaciones del Ancho de Banda disponible**

Ancho de Banda	DSIS	
	H.264	Theora
1.6Mbits/s	5	5
1.11 Mbits/s	5	5
467Kbits/s	3.06	2
272Kbits/s	2.33	2.13
121Kbits/s	-	-

#### 6.5.4 Efecto de la utilización de cuadros bipredictivos (B-frames) de H.264 en la calidad de video codificado.

Para poder determinar en que grado afecta a la calidad del video codificado la utilización de los cuadros tipo B, se procedió a codificar una muestra de video con H.264 variando la cantidad de cuadros bipredictivos que podía incluir y comparando la calidad de los videos obtenidos con el codificado con Theora. La siguiente tabla muestra las características de los videos codificados y los resultados obtenidos al medir la calidad respecto a la muestra de referencia.

**Tabla 11: PSNR, SSIM**

Muestra Codificada	Cuadros/tamaño del cuadro (bytes)			PSNR (dB)	SSIM
	I	P	B		
delfines_sinB-h264	5/15071	811/3891	0	45.03	0.99
delfines_conB-h264	5/14826	750/4039	61/2316	44.63	0.99
delfines_con3B-h264	5/14608	726/4092	85/2378	44.46	0.99
delfines_con4B-h264	5/14118	667/4206	144/2567	44.03	0.98
delfines_theo-Theora	-	-	-	41.46	0.97

### 6.5.5 Consumo medio de recursos utilizados por los *códecs* durante el proceso de codificación.

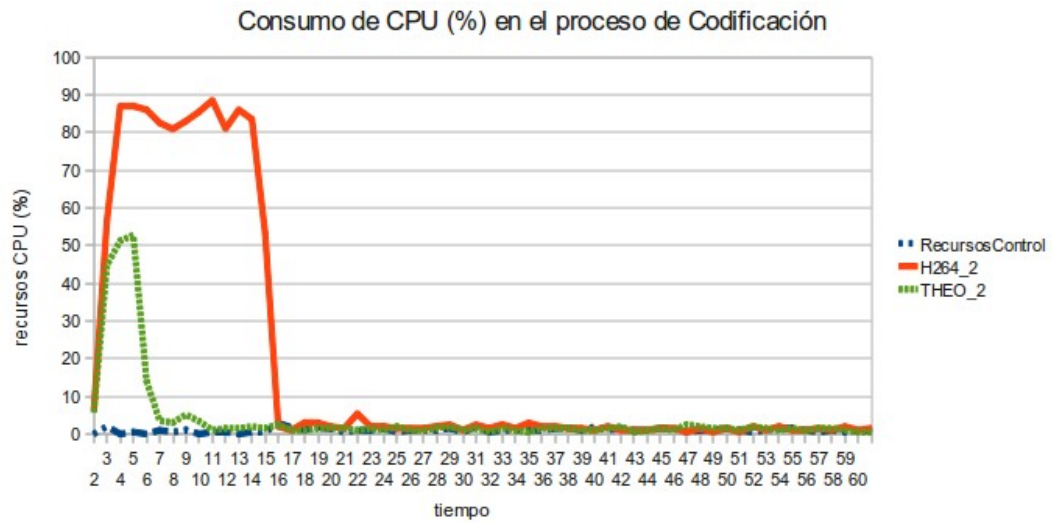
El consumo de recursos de procesamiento es uno de los factores más importantes al momento de elegir una tecnología de codificación. Para medir las diferencias entre el consumo de recursos requerido por el proceso de codificación de Theora y el de H.264, se determinó, en principio, el consumo de recursos mínimo de un computador, luego se procesó una muestra de video con ambos codificadores (uno a la vez). Las especificaciones de los computadores de prueba pueden encontrarse en los anexos de este trabajo, las características de la muestra codificada y los ajustes de los codificadores, así como las pruebas de calidad, se presentan en la **Tabla 12**.

**Tabla 12: Características de las muestras, PSNR y SSIM**

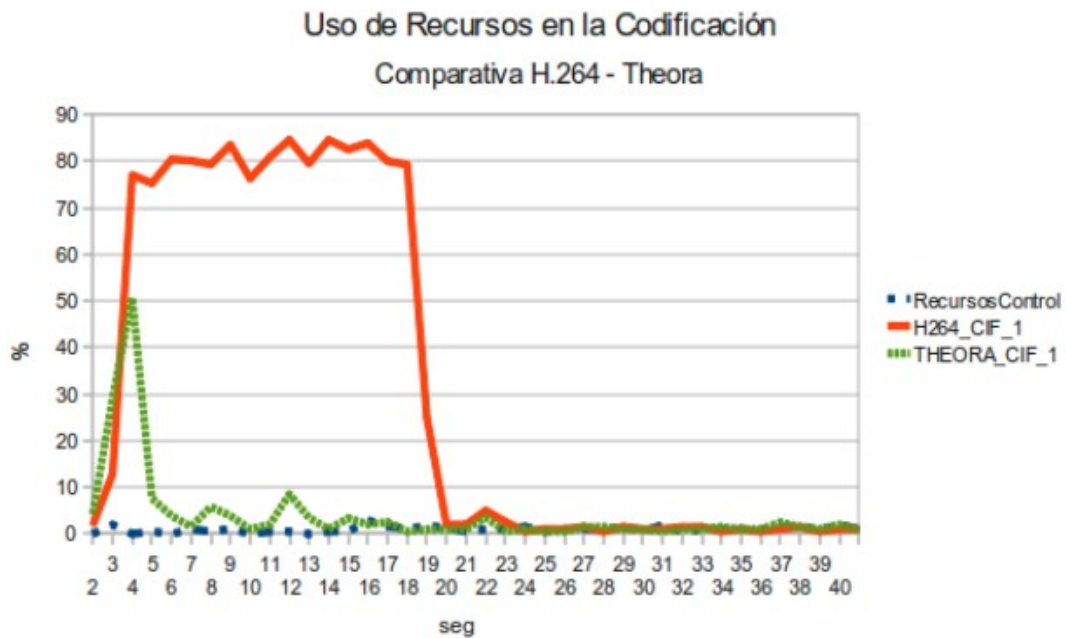
Muestra	código (parámetros)		Tamaño archivo codificado		PSNR promedio (dB)		SSIM promedio	
	h.264	Theora	h.264	Theora	h.264	Theora	h.264	Theora
Mother and Daughter CIF 300 frames	800Kbps 30fps Principal	800Kbps 30fps qi=5	1013,4 KiB	1,1 MiB	42,32	41,31	0,98	0,97
Coastguard CIF 300 frames	800Kbps 30fps Principal	800Kbps 30fps qi=5	1016,9 KiB	1152,5 KiB	31.46	32.34	0,97	0,91

Las siguientes gráficas muestran el consumo de recursos (%) durante el proceso de codificación de las muestras *Coastguard\_CIF* y *MotherAndDaughter\_CIF* en las PC1 y PC2 respectivamente:

**Gráfica 6,** Consumo de Recursos en el Proceso de Codificación, *Coastguard\_CIF*



**Gráfica 7,** Consumo de Recursos en el Proceso de Codificación, *MotherAndDaughter\_CIF*





## 6.6 Análisis de los resultados.

Como se mencionó durante el desarrollo de las pruebas, la caracterización de un *códec* y la determinación de su rendimiento están íntimamente relacionados al entorno en el cual se desea implementarlo. Sin embargo, en vista de los resultados obtenidos, podemos destacar ciertos aspectos importantes:

1. Calidad Objetiva-Subjetiva: Los *códecs* H.264 y Theora demostraron, a través de las pruebas realizadas, ser adecuados para codificar muestras de video a distintas resoluciones y tasas de bits, esto es, capaces de entregar calidades de video cónsonas con los parámetros de distribución comercial. En general, el desempeño entregado por H.264 fue ligeramente superior tanto en las pruebas objetivas como en las subjetivas, sin embargo, en la mayoría de los casos, esta “superioridad” solo fue perceptible al contrastarlos directamente con el video de referencia (no codificado). Mención aparte merecen los casos en los cuales se codificaron videos de dimensiones cuyas resoluciones no eran múltiplos de 16 (VGA, HD720, etc.), en estos casos, la diferencia entre la calidad entregada por H.264 y Theora fue mucho mas grande. Recordemos que Theora 1.0 está limitada a codificar videos de dimensiones múltiplos de 16 para mantener entero el número de macrobloques, y, a pesar de tener los mecanismos necesarios para realizar la codificación de un video de dimensiones aleatorias, enmarcándolos en cuadros que cumplan con esta condición, el mecanismo de ajuste de tamaño parece no estar suficientemente maduro y optimizado aun. Por su parte, H.264 entrega niveles de calidad “constantes” para todos los videos y resoluciones.
2. Relación Calidad Vs. Tasa de *bits*: Dado que Theora fue especialmente diseñado para la transmisión de videos en redes de ancho de banda limitado,

es lógico pensar que su eficiencia al codificar videos a bajas tasas de *bits* debería ser óptima, sin embargo, los resultados de las pruebas destinadas a establecer la relación entre la calidad y la tasa de bits no fueron consecuentes con esta idea. Theora mostró un desempeño mediocre en comparación con H.264 a bajas tasas de bits, y, contrario a lo esperado, igualó e incluso superó a H.264 para altas tasas. Las razones que expliquen la situación antes expuesta no pudieron ser dilucidadas en el transcurso de la realización de este trabajo, sin embargo, y a la luz de la documentación de ambos codificadores, es lógico pensar que la elección dinámica del tipo de cuantización a utilizar y la selección de las unidades de codificación fueron factores importantes en esta diferencia. Ambos *códec* utilizan diferentes algoritmos para elegir el procedimiento óptimo al realizar el proceso de compresión-codificación, esta “elección” dependerá, no solo de las características propias de la muestra de video, sino de la configuración de los parámetros del codificador, por tanto, es de esperarse que un ajuste cuidadoso de estos parámetros, tanto en Theora como en H.264 tenga las consecuencias esperadas sobre la calidad del proceso.

3. Efecto de las características de la transmisión de datos en redes de paquetes en la calidad del video codificado: Las pruebas físicas sobre los efectos de degradación que introducen las redes de ancho de banda limitado en el video codificado estuvieron marcadas por la pérdida de sincronía y el retardo. En general, la captura de cuadros y su posterior decodificación introdujeron pérdidas en la información, que, a su vez, provocaron la pérdida o “descarte” de cuadros (la cantidad de cuadros volcadas a las red por el emisor fue ligeramente inferior a las decodificadas por el emisor). Esta “pérdida” de cuadros (tanto en las muestras codificadas con Theora como con H.264), aunque no es perceptible por el ojo humano, hizo matemáticamente inviable realizar las pruebas objetivas de PSNR y SSIM que basan su procedimiento en

la comparación cuadro-cuadro, en este caso, dado que el ajuste visual requerido para sincronizar los cuadros siguientes es, a todas luces, improbable, se tomó la decisión de utilizar el método subjetivo de DSIS para realizar la comparación. Las pruebas basadas en la simulación de una transmisión, en la cual se introdujo matemáticamente un número determinado de bits erróneos en el flujo de bits del video codificado, no fueron afectadas por los problemas de pérdida de cuadros descritos anteriormente y, por tanto, fue factible utilizar las pruebas objetivas. Tanto en las pruebas físicas y su respectiva medición subjetiva, como en la simulación y las pruebas objetivas, H.264 demostró poseer herramientas más robustas para lidiar con los errores introducidos en el flujo de video que su contraparte Theora. Sin embargo, y como puede observarse en los cuadros de ejemplo tomados para cada video codificado, los efectos de degradación introducidos en el proceso de transmisión fueron evidentes para ambos codificadores y evidencian la importancia de establecer un futuro Estándar que permita determinar una relación métrica entre las características mínimas de una red de datos y la calidad de video que puede transmitirse a través de ella.

4. Efecto de la utilización de cuadros bipredictivos de H.264 en la calidad de video codificado. Una de las características introducidas por H.264 que más ha dado de que hablar ha sido, sin duda, el uso de cuadros bipredictivos. Sin embargo en las pruebas realizadas la presencia o no de estos cuadros no demostró tener una relación determinante en la calidad del video codificado. Debemos recordar que los cuadros B están relacionados directamente con el proceso de compresión (la información necesaria para codificar un cuadro B es mucho menor que la necesaria para codificar otros tipos de cuadros) , por tanto, es de esperarse que su utilización permita un factor de compresión mayor que cuando el video es codificado sin usar esta característica. Lo que demuestran los resultados obtenidos, es que, aun cuando se realiza una mayor compresión de los datos, la calidad permanece relativamente constante. Sin duda el uso de cuadros bipredictivos presenta una gran ventaja a H.264 frente

a otros codificadores.

5. Consumo medio de recursos utilizados por los *códecs* durante el proceso de codificación: El consumo de recursos necesarios para realizar la codificación de video puede no ser importante a ciertas escalas (pequeñas empresas y consumo personal), sin embargo, es de vital importancia al momento de elegir una plataforma para la distribución masiva de contenido multimedia. En las pruebas realizadas, Theora 1.0 demostró poder realizar la tarea de codificar y decodificar muestras de video utilizando una cantidad de recursos de procesamiento mucho menor que H.264 y ofreciendo a la vez niveles de calidad competitivos. Estos resultados no presentan ninguna sorpresa después de haber examinado las características de los *códecs*, evidentemente, H.264 es un *códec* mas complejo y robusto que Theora, y claro, esta complejidad y robustez trae como obvia consecuencia un aumento en el número y complejidad de las operaciones de computo necesarias. Queda en manos del diseñador de la solución para distribuir video la tarea de encontrar el equilibrio entre la calidad de video que debe entregar, según la aplicación y el tipo de cliente, y el costo en poder de procesamiento que esta dispuesto a pagar por ella.

## CAPITULO VII

### Escalabilidad de H.264

Garantizar la entrega de video sobre redes con una calidad relativamente constante presenta serias dificultades. El ancho de banda disponible, las condiciones del enlace, entre otros factores, cambian constantemente y suelen degradar la calidad del video transmitido. Además de las condiciones propias del “envío” de video, también es necesario proporcionar soluciones capaces de responder a los diferentes niveles de calidad que requieren los usuarios.

H.264 ofrece dos diferentes maneras de enfrentar la necesidad de “adaptar” la codificación de video a las necesidades del usuario y las condiciones del enlace. La forma más simple de este proceso de adaptación para muestras pre-codificadas es la de codificar el video en múltiples e independientes secuencias codificadas con parámetros distintos y dejar que el servidor, dinámicamente y en tiempo real, cambie entre uno y otro dependiendo de las condiciones de la red (Estandar H.264/AVC). La forma, menos simple pero quizás mas eficiente, es la de codificar el video en una sola secuencia más compleja utilizando un *códec* “escalable” que permita enviar diferentes “partes” de la información contenida en la secuencia de acuerdo a las condiciones cambiantes de la red (H.264/SVC).

#### 7.1 H.264/AVC.

Como se mencionó anteriormente, el estándar H.264/AVC permite codificar un video en diversas secuencias independientes y de diferente calidad y, cambiar, entre una y otra secuencia para adaptarse a las necesidades y características del medio de transmisión. Anteriormente, este “cambio” entre las diferentes secuencias solo podía realizarse desde los cuadros *Intra* (ya que los cuadros *Inter* son, de hecho, simples referencias modificadas y no contienen la información suficiente), la

desventaja evidente de utilizar solo cuadros I durante el cambio viene dada por la eficiencia en su codificación, un cuadro I requiere el envío de mayor información respecto a los P y/o B.

H.264/AVC introduce en el proceso de codificación dos nuevos tipos de cuadros, SP y SI que permiten, sin enviar imágenes *intra* muy costosas en tiempos de procesamiento, pasar de un video a otro utilizando predicción temporal o espacial como antes, pero con la ventaja que permiten la reconstrucción de valores específicos exactos de la muestra aunque se utilicen imágenes de referencia diferentes o un número diferente de imágenes de referencia en el proceso de predicción. A pesar de las ventajas de la codificación utilizando cuadros SI y SP, su uso viene acompañado de un aumento en la latencia y en la necesidad de recursos de procesamiento.

## **7.2 H.264/SVC**

El *Join Video Team* (JVT) del grupo ISO/MPEG y el ITU-T ratificaron el estándar SVC (*Scalable video Coding*) como parte de los anexos de MPEG-4. El objetivo del SVC es permitir reconstruir el flujo de video en base a uno o mas flujos de video primario llamados “capas” codificados con diferentes niveles de calidad. Una capa primaria (capa base) puede contener la información completa para visualizar el video con calidad “reducida” (temporal o espacialmente), y las siguientes capas pueden ser añadidas para mejorar la calidad de acuerdo a las condiciones de la red y los requerimientos del usuario. Para permitir la compatibilidad con el estándar H.264/AVC se garantiza que la capa base sea compatible con AVC, las siguientes capas serán codificadas de acuerdo al tipo de SVC que se utilice. [12]

La escalabilidad puede lograrse de diversas formas; escalabilidad Temporal, Espacial y la denominada Fidelidad/Escalabilidad.

- Escalabilidad Temporal: Se logra la escalabilidad eliminando cuadros completos del flujo de video. El video con menor número de cuadros será la

capa base (de menor calidad) y mientras más capas se agreguen, se irán agregando mas cuadros por cada segundo de video hasta que, una última capa, represente la totalidad de los cuadros del video original. Esto, evidentemente, reduce la cantidad de información enviada y por tanto el ancho de banda requerido.

- Escalabilidad Espacial: El video es codificado en múltiples resoluciones espaciales. Los flujos de video primario y de menor resolución son utilizados como capa base para predecir los datos de los de mayor resolución con la finalidad de reducir la tasa de bits necesarios para la múltiple codificación.
- Fidelidad/Escalabilidad: El video es codificado en una sola resolución espacial pero en diferentes calidades. Los flujos de video primarios, de calidades menores, son usados como base para predecir los datos para los de calidades superiores.

Como todo proceso, la escalabilidad en H.264 presenta ventajas y desventajas. Es evidente que la necesidad de recursos de procesamiento aumenta cuando se hace uso de estos procedimientos; sin embargo, y dado que los elementos de las redes son cada vez mas poderosos en cuanto a su poder de procesamiento se refiere, este factor puede no ser decisivo. Entre las ventajas que debemos mencionar están:

- *QoS* en emisiones *multicast*: Evidentemente, la capacidad de adaptar el flujo de acuerdo al usuario, permite ofrecer diferentes niveles de calidad de servicio a cada cliente, por ejemplo, la misma emisión podrá ser recibida por clientes de TV-Internet-T-movil adaptado a los diferentes anchos de banda.
- Control de errores: En el caso de disminuciones abruptas en el ancho de banda, el video no necesariamente se interrumpe, simplemente se disminuye dinámicamente la calidad pero manteniendo la continuidad de la transmisión.
- Videoconferencias: SVC permite realizar conferencias con multitud de participantes sin necesidad de utilizar una *Multipoint Conference Unit* (MCU),

lo que resulta muy poco asequible. En su lugar se puede incorporar un dispositivo inteligente capaz de multiplexar y entregar a cada usuario participante tan sólo la cantidad de información que puede asumir, sin necesidad de codificar de nuevo la señal como lo hacen las MCUs. [12]

### **7.3 Escalabilidad y Theora 1.0.**

Por el momento, Theora no tiene planes de incluir extensiones de escalabilidad en sus códec, por el contrario, debemos recordar que Theora no permite ni siquiera cambios de resolución o *bit-rate* en el video (dada la necesidad de inicializar el codificador al inicio con toda la información). A pesar de que un contenedor *Ogg* pudiera tener varios videos diferentes codificados con Theora, las necesidades de desarrollar un sistema dinámico de selección que permita reproducir uno u otro están aun lejos de poder ser implementadas. En el presente, las soluciones H.264/SVC están en su etapa de desarrollo y son pocos los servicios que a nivel comercial utilizan esta característica, sin embargo, la carencia de Theora de mecanismos de escalabilidad podría, en un futuro cercano, inclinar la balanza hacia las soluciones de MPEG.



## CAPITULO VIII

### Compatibilidad de Theora con las diversas plataformas tecnológicas

Uno de los factores determinantes para elegir un *códec* de video es, sin duda, su interoperabilidad con las diversas plataformas tecnológicas disponibles en el mercado. Un codificador, para ser considerado como *estandar* debe ser capaz de ser utilizado a plenitud (codificación-decodificación eficiente) sobre los sistemas operativos comunes; además, su implementación en hardware debe ser factible y económicamente viable.

#### 8.1 Compatibilidad de Software

Para analizar la posible compatibilidad de un *códec* de video es necesario destacar ciertos parámetros:

- Tipo: La herramienta para la codificación-decodificación puede ser solo la implementación del algoritmo del *códec* para ser incluida en una aplicación, o una aplicación completa destinada a la codificación de video.
- Capacidad: La implementación de una plataforma de video, según el SO en el cual está implementada y dependiendo de su desarrollador puede tener capacidades para: reproducir, decodificar y/o codificar.

La siguiente tabla representa diversas soluciones de software capaces de utilizar Theora 1.0 en diversas plataformas de software:

**Tabla 13:** Compatibilidad de Theora – Software, Autor

Descripción	Soporte de Sistema Operativo		
	Linux	Windows	Mac
Libtheora: Es la implementación oficial de Theora por <i>Xiph</i> . Capaz de ser integrada en diversas aplicaciones de transcodificación de video.	SI	SI	SI
Ffmpeg2theora: Es la aplicación oficial desarrollada por <i>Xiph</i> para ser usada con la implementación libtheora.	SI	SI	SI
FFMPEG:herramienta multiplataforma capaz de codificar-decodificar video en gran número de formatos. Posee una implementación propia de Theora 1.0	SI	SI	SI
FireFox 3.5, Google Chrome, Opera: Exploradores WEB con soporte nativo para ogg-Theora	SI	SI	SI
VLC, Mplayer: Reproductores de multimedia	SI	SI	SI
Icecast, Fluemotion: Aplicaciones para la emisión <i>unicast-multicast</i> de media.	SI	SI	SI
Ekiga: aplicación para video-conferencias	SI	SI	SI

## 8.2 Compatibilidad de Hardware.

Realizar el proceso de codificación-decodificación de un video requiere, para cualquier *códec*, un alto poder de procesamiento. Las opciones de componentes físicos específicamente diseñados para el manejo de video no son nuevas, sin embargo, la implementación de hardware para Theora se encuentra en sus primeras etapas de desarrollo. A continuación se enumeran algunos de los dispositivos capaces de manejar videos codificados con Theora:

- Productos destinados al consumidor:
  - COWON D2, O2: Reproductores multimedia desarrollados por *CowonAmerica* con capacidad para reproducir videos codificados con Theora.
  - KiSS DP-1500: Reproductor de DVD desarrollado por LinkSys (Cisco System) con compatibilidad para diversos formatos, incluyendo Theora.

- Productos NO destinados al consumidor:

Diversos adelantos se han realizado en la implementación del algoritmo de Theora utilizando múltiples soluciones físicas; hasta ahora, las que han tenido mayor aceptación han sido aquellas basadas en el uso de FPGAs (*Field Programmable Gate Array*).

- Elphel 333: Cámara web desarrollada con FPGAs y capaz de codificar videos con Theora.

En la actualidad existe una implementación para decodificar videos Theora desarrollada por los equipos de *The Google Summer Of Code* realizada en VHDL (*Very High Speed Integrated Circuits hardware description language*) sobre FPGAs.

### 8.3 Theora y HTML5.

Dentro del área de la compatibilidad, el desarrollo de las especificaciones de HTML-5 y su relación con Theora merecen ser estudiados a parte. La creciente competitividad entre los exploradores WEB ha sido la piedra de tranca para la elección de un *códec* que estandarice el video en html y que suplante a Flash en la difusión de contenido multimedia en forma nativa. En general, el equipo de desarrollo de HTML-5 se debate entre adoptar H.264 o Theora como *códec* estándar y la solución hasta ahora ha sido el no incluir a ninguno como parte de las especificaciones preliminares. Esta decisión, evidentemente problemática, impide que hasta el momento los creadores de contenido puedan publicar videos en un solo formato que sea capaz de ser reproducido por los elementos nativos de los exploradores de Internet.

La discusión entre Theora y H.264 en HTML-5 se basa fundamentalmente en dos factores; Patentes y Calidad. Mientras compañías como Mozilla y Opera argumentan que Theora es lo suficientemente maduro para competir en el mercado y se basan en su gratuidad para recomendarlo, otras como Apple sostienen que no es capaz de entregar niveles de calidad comparables con H264 y algunas terceras como Google apoyan la idea de implementar ambos. Sea cual sea la decisión que se tome, ésta se verá marcada por los costos asociados, si bien es cierto que la relación “calidad/*biterate*” es valiosa económicamente, los pagos de licencias y la dependencia a futuro es un riesgo que no muchas compañías están dispuestas a asumir. Lo que es cierto y evidente gracias a esta controversia es que Theora y H.264 son, por el momento, los codificadores que presentan mejores características y por tanto los candidatos a convertirse en un estándar general.

## CAPITULO IX

### Distribuciones y Licencias

Uno de los factores determinantes para la elección de una plataforma tecnológica es el referente a los costos de una posible migración o implementación de la misma. En el área de los codificadores de video los costos asociados no siempre pueden ser establecidos de manera exacta debido a la falta de homogeneidad entre los diferentes tipos de “permisos” de uso. Para realizar una descripción de los factores económicos a considerar debemos definir brevemente el marco legal relacionado a la propiedad intelectual.

- Licencia: Contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquel a quien éste haya cedido los derechos de explotación, quien elige la licencia según la cual se distribuye el software.
- Patente: Conjunto de derechos exclusivos garantizados por un gobierno o autoridad al desarrollador de un nuevo producto susceptible de ser explotado comercialmente.
- Derecho de autor (*copyright*): forma de protección legal en la cual se reconoce el derecho del desarrollador de un nuevo producto sobre su distribución.

Además de fijar las pautas de distribución, los desarrolladores deben establecer el tipo de licencia a la que deberá atenerse el usuario final del producto. La licencia abarca los posibles costos, los permisos y condiciones que el autor establece para el uso del software que distribuye. Estas condiciones pueden abarcar el territorio de aplicación, plazos de utilización, límites para la modificación de la estructura del

programa, redistribución, créditos o ámbitos en los que éste deberá ser utilizado, junto con cualquier otro accionar que el desarrollador desee regular.

Dentro de los tipos de licencia, y en cuanto al factor económico se refiere, podemos realizar una clasificación básica: Licencias de software de distribución gratuita y licencias de software que implican el pago de beneficios al desarrollador. Si bien, esta no es una clasificación de origen legal, nos permite hacernos una idea inicial en el complejo ámbito de las licencias de software. Es común que exista una confusión entre la terminología utilizada para definir un tipo de licencia, en general, los términos “*libre*” y “*gratis*” son erróneamente entendidos como sinónimos, y, erróneamente también, el término “privativo” se relaciona con “de pago” o que “genera beneficios económicos al desarrollador”. Con el fin de aclarar este tipo de confusiones, debemos clasificar los tipos de licencia de acuerdo con los criterios más utilizados a nivel mundial:

1. Licencias de Software NO libre: También conocidas como de “software privativo”, son aquellas en el que el desarrollador establece los derechos de uso, distribución, redistribución, copia, modificación y, en general, cualquier otra consideración que estime necesaria. Usualmente entre las condiciones establecidas se prohíbe o limita que el software sea modificado, desensamblado, y que se aplique en él cualquier tipo de ingeniería inversa. De la misma forma, se establece la inversión económica que debe hacer el licenciatario de acuerdo al tipo de uso que le dará al software y a las condiciones del licenciante. Las licencias de software no-libre no necesariamente implican el pago de *royalties* o beneficio económico al desarrollador, en algunos casos la distribución del software se realiza de manera gratuita siempre y cuando el licenciatario cumpla las condiciones establecidas (por ejemplo: que el uso del software no tenga fines comerciales, que sea utilizado en una escala determinada, que se mantenga la publicidad introducida en el software, etc.). Los fabricantes de programas sometidos a este tipo de licencias por lo general ofrecen servicios de soporte técnico y

actualizaciones durante el tiempo de vida del producto. Algunos ejemplos de este tipo de licencias son las llamadas **CLUFs**: **Contrato de Licencia para Usuario Final** o **EULAs**: *End User License Agreement*, por sus siglas en Inglés.

2. **Licencias de Software Libre**: Las licencias de software libre tienen la finalidad de difundir un software o parte de él de manera masiva, para ello, establece pocas limitaciones en cuanto al tipo de distribución y posibles usos del mismo. Aunque no es una norma, las licencias para software libre permiten al usuario copiar, distribuir y modificar el software siguiendo las pautas y condiciones establecidas en la licencia. De la misma forma en que “privativo” no implica “costoso”, el termino “libre” no implica gratuidad (aunque generalmente así sea su distribución), el desarrollador de software libre podrá, según las condiciones establecidas en el contrato de licencia, establecer costos asociados, aunque generalmente, estos costos son vinculados a la prestación de servicios como la asesoría técnica relacionada al software, actualizaciones, etc. Entre los tipos mas comunes de licencias de software libre podemos nombrar: *BSD license*, *GNU (General Public License)*, *Common Public License*, etc.

### **9.1 H.264 – Licencia.**

La compañía MPEG LA es la encargada a nivel mundial de distribuir, bajo una sola licencia, la patente de uso de H.264 la cual a su vez está conformada por un grupo de patentes especiales propiedad de diversos titulares. En la licencia AVC/H.264 se prevé la concesión de sublicencias a los fabricantes de mecanismos de codificación y de decodificación para que tengan derecho a fabricar y vender el equipo y los algoritmos utilizados para la compresión. Se prevé también un derecho limitado de utilización personal para los usuarios o entre usuarios (por ejemplo, en el caso de las videoconferencias o de los servicios de mensajería móvil). En una parte separada de la licencia se estipula la concesión de sublicencias para los proveedores

de contenido de video o los suministradores de servicios. Por consiguiente, existe por un lado una licencia para la fabricación y la venta de un producto conforme a las normas, y por otro, una licencia para la utilización del producto en determinadas aplicaciones comerciales. [13]

La licencia para H.264 está definida bajo los términos de un EULA, y entre las restricciones y limitaciones que estipula, podemos nombrar:

- Se prohíbe la explotación, duplicación, modificación o transferencia del producto y de su documentación, de forma total o parcial.
- Se prohíbe la alteración, total o parcial de las condiciones de derecho de autor, marcas, publicidad.
- Se prohíbe la decompilación y el desensamblado, la ingeniería inversa y cualquier otro intento de acceder al código fuente. [14]

Los costos presentados en la Tabla 14 representan la inversión necesaria para el periodo 2.009-2.010, los costos asociados a emisión de video vía Internet en la cual el cliente no asume los costos directos del video estarán exentos de pago hasta el año 2.011 luego del cual se tendrá que cancelar el pago equivalente al de la TV-gratuita.

**Tabla 14:** Costos de H.264, [13][14] [15]

Tipo Licencia	Costos	
	Unidades	Inversión
Sub-Licencia para incorporar codificador-decodificador en Hardware para ser vendidos al consumidor como parte de sistemas informáticos, pero NO parte del Sistema Operativo.	0-100K	0\$/unidad
	+100k	0,20\$/unidad
Sub-Licencia para incorporar codificador-decodificador para ser vendidos al consumidor como parte de sistemas informáticos, y parte del Sistema Operativo.	+5M	0,10\$/unidad



Tipo Licencia	Costos
<p>Distribución de contenido donde el cliente final asume los costos directos del servicio de video.</p> <p>Distribución de contenido donde la remuneración proviene de fuentes alternas al contenido del video. (TV, Internet, etc)</p>	<p>Por Titulo: 0,02\$ ó</p> <p>1-100K Subscriptores: 0\$</p> <p>hasta 250k Subscriptores 25000\$</p> <p>hasta 500k Subscriptores 50.000\$</p> <p>hasta 1M Subscriptores 75.000\$</p> <p>+1MSubscriptores: 100.000\$</p> <p>100k a 500k clientes 2500\$ por Emisión/año.</p> <p>+500k clientes 5000\$ por Emisión/año</p> <p>+1M clientes 10000\$ Emisión/año</p>
<p>Costos Maximos anuales (2.010) = 5.000.000 \$</p>	

### 9.1 Theora – Licencia.

Theora, al igual que todas las tecnologías relacionadas con él y liberadas por la fundación *Xiph* están amparadas bajo licencia BSD-Style (gratuita, de distribución libre y uso permisivo). Lo que implica que cualquier uso, comercial o no, es gratuito y no implica ningún tipo de pago de patentes.

Las características del tipo de licencia de Theora implica que cualquier desarrollador podrá escribir, ensamblar e incluir el código abierto del *códec* en cualquier aplicación, también es permitida la modificación y redistribución del *códec* bajo mínimas condiciones.

## CAPITULO X

### THUSNELDA

Una segunda generación de Theora se encuentra ya en desarrollo bajo el nombre de Thusnelda 1.1. Thusnelda no solo corrige algunos errores encontrados en Theora y generalmente referidos al control de la tasa de bits, también introduce mejoras en la eficiencia del proceso de codificación. Thusnelda no es una mejora de Theora, es una completa reestructuración del *códec* basado en las especificaciones, esto ha permitido implementar estructuras que, hasta el momento, permanecían solo entre las características teóricas.

Entre las ventajas introducidas por Thusnelda se encuentran:

- Implementación de 4MV: Como se mencionó, Theora divide los cuadros en Macrobloques, estos Macrobloques, dentro del procedimiento de compensación de movimiento, pueden ser representados por hasta 4 vectores de movimiento, pueden tomarse los 4, un promedio, o incluso algunos pueden ser descartados. El proceso de decisión con respecto a que configuración utilizar es aun básico en Theora 1.0, Thusnelda, por su parte, introduce esquemas de compensación de movimiento que permiten tomar una decisión óptima y rápida sobre los datos a codificar, permitiendo así reducir el numero de bits enviados manteniendo una alta calidad de video.
- Elección de matrices de cuantización por bloques: Theora provee un grupo de diferentes matrices de cuantización que son elegidas para cada video y por cada cuadro. Thusnelda no solo conserva las capacidades de Theora de aplicar diversas matrices por-video y por-cuadro, también introduce la capacidad de cambiar la matriz de cuantización para cada bloque.
- Soporte para espacios de color 4:2:2 y 4:4:4 y la habilidad para utilizar diferentes matrices de cuantizacion para cada espacio, esto evidentemente

permite obtener mejor calidad de color manteniendo la eficiencia del proceso.

La versión alpha de Thusnelda ya se encuentra disponible en diversos formatos: Código fuente de la librería libtheora, binarios para la línea de comandos de ffmpeg2theora e incluso componentes para QuickTime en Mac.

## **CAPITULO XI**

### **Experiencias Propuestas**

Tal y como se mencionó en la descripción de los objetivos de este trabajo, se pretende, luego de sopesar la relevancia de los diferentes tópicos expresados en él; proponer experiencias que puedan ser realizadas en los laboratorios de comunicaciones de la Escuela de Ingeniería Eléctrica de la U.C.V. y que den pie a la elaboración de futuras “Prácticas de laboratorio” donde se evidencie, no solo el proceso de codificación, sino las posibles alternativas gratuitas y libres para la implementación de soluciones de compresión y transmisión de video.

El proceso de codificación de video, aunque matemáticamente complejo, es conceptualmente simple y sus características principales pueden ser estudiadas en diversos niveles de educación. Con el fin de proponer experiencias generales que permitan desarrollar actividades académicas destinadas al estudio de los procesos de codificación, debemos disponer de cierta escala de complejidad que permita adaptarlas a las necesidades específicas del educando y a las herramientas teóricas y prácticas con las que se cuenta.

Con el fin de proponer un modelo con diferentes niveles de complejidad, se tomó la decisión de dividir el proceso de codificación, no en sus bloques fundamentales, sino de acuerdo al tipo de acercamiento que el estudiante en formación puede tener a él. Se plantean tres procesos fundamentales desde los cuales se espera lograr, una visión general, un análisis profundo y una motivación a futuras investigaciones:

1. Entender.
2. Caracterizar y Evaluar.
3. Optimizar y desarrollar.

La siguiente tabla espera representar los objetivos generales y específicos que debe buscar cada etapa en el estudio de la codificación de video:

**Tabla 15, Objetivos a Lograr, Autor**

<b>Etapa</b>	<b>Objetivos Generales</b>	<b>Objetivos Específicos</b>
<b>Entender</b>	<p>.- El objetivo fundamental debe estar dirigido a que el estudiante se familiarice con el concepto de compresión y las herramientas conceptuales básicas que se utilizan en él.</p>	<p>.- Entender el concepto de la codificación de datos con y sin pérdidas y especialmente de video.</p> <p>.- Capacitar al estudiante para definir y caracterizar los elementos básicos del proceso de codificación de video: División de la imagen en elementos fundamentales, predicción, cuantización, codificación.</p> <p>.- Familiarizar al estudiante con los conceptos de redundancia espacial y temporal y las herramientas básicas utilizadas para aprovechar estas redundancias en la compresión y codificación de video.</p>
<b>Caracterizar y Evaluar</b>	<p>.- Debe tener como principal objetivo capacitar al estudiante para caracterizar el proceso de codificación de video y analizar sus resultados.</p>	<p>.- Familiarizar al estudiante con las herramientas y tipos de métrica utilizadas para determinar la calidad de muestras de video codificadas.</p> <p>.- Dotar al estudiante con las herramientas teóricas y prácticas que le permitan desarrollar un procedimiento para comparar distintos <i>códecs</i> de video presentes y futuros y recomendar los mas adecuados según las características de la aplicación a implementar.</p>

	<b>Objetivos Generales</b>	<b>Objetivos Especificos</b>
<b>Optimizar</b>	.- El fin principal de esta etapa deberá ser el proporcionar al estudiante las herramientas teóricas y prácticas que le permitan optimizar los procesos de codificación y/o la integración de la etapa de codificación de video en diversas aplicaciones.	.- Dotar al estudiante de las herramientas teórico-prácticas necesarias para entender y ajustar los parámetros inherentes al proceso de codificación y realizar modelos que permitan representar la relación entre estos parámetros y la eficiencia del proceso de codificación. .- Familiarizar al estudiante con los mecanismos internos de decisión que realizan los codificadores durante el procesado de las muestras de video y proponer experiencias que permitan observar los efectos de estas decisiones en la eficiencia del proceso de codificación. .- Dar al estudiante las herramientas necesarias para el estudio, implementación y desarrollo de herramientas complejas del proceso de codificación como el filtrado y la corrección de errores y las técnicas de Escalabilidad Variable.

Para que las experiencias a recomendar sean factibles de realizar en los laboratorios de la Escuela de Ingeniería Eléctrica de la UCV deben tener ciertas características que permitan su implementación:

- Deben poder ser realizadas con equipos comunes y con poder de procesamiento limitado.
- Deben utilizar, en lo posible, software que no requiera el pago de licencias o la adquisición de permisos especiales.

- Deben permitir el estudio del proceso, y no solo el de sus resultados, para ello, es altamente recomendable el trabajo con software de código libre.

#### Experiencias Recomendadas:

##### 1. Etapa: Entender.

Para cumplir con los objetivos de esta etapa debe poder realizarse el proceso completo de codificación de muestras de video, es deseable que después de entender los conceptos de división de bloques, de redundancia espacial y temporal y las herramientas que explotan esta redundancia, éstas puedan ser visualizadas por el estudiante. Debe poder realizarse una comparación, al menos a nivel subjetivo, entre la muestra original y los resultados del proceso de codificación.

- Conocimiento previo necesario:
  - Conceptos de: Formas de representar un video digital (Formatos, Espacios de Color, Cuadros, Pixels), codificación (con y sin pérdida).
- Herramientas necesarias:
  - Computador con sistema operativo Linux y las herramientas: VLC, *ffmpeg*, *Scilab* correctamente configuradas y con las librerías respectivas de los *códecs* a utilizar (Theora, H264).
- Actividades:
  - Visualizar, en una imagen estática los diferentes espacios de color (YUV, RGB) y los planos y pixels que la conforman representados por las diferentes matrices. (Scilab)
  - Realizar la decodificación (sin pérdida) de una muestra de video en cualquier formato a un formato intermedio CIF con representación YUV para obtener un video de referencia.
  - Realizar la codificación del video de referencia con diversos codificadores y con diferentes parámetros de codificación. (VLC).

- Comparar, perceptualmente, la calidad de los videos codificados en relación a la muestra de video de referencia.
- Visualizar, para cada cuadro del video codificado: el tipo de cuadro, la codificación utilizada y los vectores de movimiento. (*ffmpeg, ffplay*).

## 2.- Etapa: Caracterizar y Evaluar.

- Conocimiento previo necesario:
  - Además de los conocimientos adquiridos en la Etapa “Entender”, el estudiante debe familiarizarse con los conceptos de: Calidad de video, Parámetros que definen la calidad de video y algoritmos propuestos para realizar la métrica (subjetivos, objetivos, mixtos), degradación del video en la etapa de codificación y en el envío, calidades de video típicas esperadas (según aplicación y capacidad).
- Herramientas necesarias:
  - Además de las utilizadas en la etapa anterior, debe estar instalada alguna herramienta que permita realizar las mediciones objetivas de calidad de video. (se recomienda Evalvid para linux).
- Actividades:
  - Realizar la codificación de muestras de video con diferentes *códecs* y estableciendo diferentes parámetros en el proceso de codificación. (*ffmpeg, ffmpeg2theora, VLC*).
  - Realizar la decodificación SIN pérdida de los videos previamente codificados para obtener, para cada uno, un video en formato intermedio CIF (YUV).
  - Utilizar las herramientas de medición para obtener, para cada video, los parámetros de calidad objetivos (PSNR y SSIM) de acuerdo al video de referencia.
  - Realizar una medición de tipo subjetivo y basada en una escala pre-



establecida (DSIS)

- Comparar los resultados obtenidos en las diversas mediciones, establecer relaciones entre estos resultados y la variación de parámetros efectuada, comparar los resultados entre sí para verificar la validez de los procedimientos de medición.
- Realizar la medición de los parámetros inherentes al proceso de codificación: consumo de recursos de procesamiento, latencia, relación Calidad Vs. Tamaño del archivo codificado (o tasa de bits).
- Comparar los códecs utilizados.

### 3. Etapa: Optimizar.

La etapa Optimizar necesita un conocimiento más profundo del proceso de codificación y de las herramientas matemáticas utilizadas. Sin embargo, *códecs* como H.264 y Theora nos permiten realizar variaciones de muchos de los parámetros relacionados con estas herramientas para, en principio, poder caracterizarlas y evaluar su impacto en la eficiencia del proceso de codificación de video.

- Conocimiento previo necesario: Además de los obtenidos en las etapas anteriores, los conceptos previos necesarios dependerán directamente del tipo de parámetro a optimizar o de la aplicación a implementar con los *códecs*.
- Herramientas: Las utilizadas en las etapas anteriores + las necesarias para el estudio específico planteado.
- Actividades Propuestas: (dada la complejidad de esta etapa, solo se proponen actividades de tipo general).
  - Visualizar y Evaluar parámetros del *códec* de video H.264 y determinar su influencia en la eficiencia del proceso de codificación.
  - Limitar, modificar y/o especificar, en diferentes pruebas, el número y tipo

de los Vectores de movimiento a utilizar en el proceso de codificación, determinar el impacto de estas variaciones en el proceso de codificación y en la calidad de los resultados obtenidos.

- Limitar, modificar y/o especificar los mecanismos sobre los cuales se asignan los tipos de cuadros a utilizar y las relaciones entre ellos.
- Limitar, modificar y/o especificar el tipo y las características de las matrices de cuantización a utilizar por el *códec*.
- Limitar, modificar y/o especificar la forma como el *códec* selecciona el tamaño, ubicación, forma y tipo de bloques y macrobloques y determinar su influencia en la eficiencia del proceso.
- Limitar, modificar y/o especificar el tipo de codificación variable a utilizar por el *códec* y otros parámetros relativos al proceso de codificación entrópica.
- Limitar, modificar y/o especificar los mecanismos de multiplexación y empaquetamiento post codificación, caracterizar el proceso según las diferentes aplicaciones a implementar.
- Implementar un proceso de Codificación Escalable de video para su posterior distribución bajo condiciones diversas de red, analizar la eficiencia de la codificación VSC y proponer soluciones específicas de desarrollo de aplicaciones que utilicen este concepto.
- Realizar una implementación de un *códec* de código libre y abierto (ej. Theora) y realizar modificaciones que optimicen su desempeño en aplicaciones específicas.

Como pudo observarse en la descripción de las actividades a realizar, se utilizan, en todos los casos, librerías pertenecientes a programas de software libre, gratuito y de código abierto. Las especificaciones de código y parámetros para cada una de las pruebas puede encontrarse generalmente en la ayuda de cada aplicación (\$ aplicación –help). Las Etapas de “Entender” y “Caracterizar y Evaluar” fueron desarrolladas teniendo en mente a estudiantes de pre-grado y de cursos básicos de

post-grado, las actividades recomendadas en la etapa “Optimización”, son abiertas y pretenden estimular la realización de trabajos de investigación o de estudio profundo de los procesos de codificación e implementación de soluciones para distribución masiva de contenido multimedia.

## CONCLUSIONES

El desarrollo de soluciones para la distribución masiva de contenido multimedia ha tenido un gran impacto en la constante actualización de los mecanismos relacionados con mejorar la eficiencia del proceso. En el competitivo mercado del video, la meta es, sin duda, reducir los costos de procesamiento y transmisión, aumentar la calidad entregada al cliente y hacer accesible la tecnología un mayor número de posibles usuarios.

Durante el desarrollo de este trabajo se estudió solo un eslabón de la larga cadena de tecnologías utilizadas para el manejo y distribución de material de video; sin embargo, el procesado del video antes de ser distribuido presenta relación directa con los factores más influyentes al momento de elegir una plataforma tecnológica determinada: calidad y costos. Con el fin de realizar un análisis completo de los resultados de este trabajo, es necesario dividir, momentáneamente, el estudio de estos factores, para luego concatenarlos en una conclusión final.

### *Calidad*

Como se ha mencionado reiteradamente, la calidad es un concepto subjetivo y directamente dependiente de las características propias del sistema visual humano. El modelo de calidad ideal sería, evidentemente, entregar al cliente muestras de video que correspondan con exactitud a la muestra original, sin embargo, y por las razones expuestas a lo largo de este trabajo, esto es tecnológica y económicamente inviable. Surge entonces la necesidad de codificar los videos originales de forma tal que, aunque se pierda información para reducir su tamaño, el video pueda ser reconstruido y el resultado satisfaga las necesidades del observador. Es, en la codificación, donde debe establecerse un equilibrio entre la reducción de los datos y los niveles de satisfacción del cliente.

Los codificadores de video, como Theora y H.264 basan su funcionamiento en las características propias y comunes de las muestras de video, como lo son la redundancia temporal y espacial, y en las características del sistema visual humano como la capacidad para percibir el movimiento y la precisión de los detalles. Las herramientas matemáticas que utilicen para tales fines y la optimización de los algoritmos utilizados serán, al final, los factores determinantes para lograr las metas de calidad de video deseadas.

Aunque situados en la misma clase, debido al tipo de procedimientos y herramientas matemáticas que integran, y, a pesar de que a simple vista el esquema de codificación utilizado es, por decir menos, similar, Theora y H.264 son codificadores con orígenes y objetivos diferentes. H.264 es un *códec* de alta complejidad, su definición en base a diferentes perfiles lo hace capaz de entregar material adaptado a las necesidades propias de cada aplicación y las características de escalabilidad le permiten, además, adaptar las características de codificación de forma dinámica a los diferentes canales de distribución. Theora, por su parte, tiene como propósito fundamental adaptar contenidos de video al transporte sobre redes IP, y por ello, su desarrollo va encaminado a satisfacer necesidades específicas en este campo.

Dado el carácter general de las pruebas realizadas, debemos reconocer que, aunque válidas, son insuficientes para determinar cual de los códecs, Theora o H.264 sería el mas adecuado; de hecho, podemos sostener la idea de que no existe un códec “universalmente” adecuado. Sin embargo, es importante destacar que, dentro de este contexto general, Theora demostró ser capaz de entregar resultados con calidades muy cercanas a las ofrecidas por el códec de MPEG y consecuentes con los niveles de calidad exigidos por el mercado. La adopción de un codificador como Theora, si tomamos como único patrón la “calidad de video entregada”, tendrá que ser definida por el tipo de aplicación a utilizar y el mercado al que este dirigido. Los análisis comparativos efectuados demostraron que, la complejidad y capacidad de configuración de H.264 le otorgan ventajas en cuanto a la calidad de video que puede

entregar; sin embargo, la diferencia con los resultados obtenidos con Theora fue, a todas luces, pequeña, y la decisión final dependerá, únicamente, del desarrollador de la aplicación y de las necesidades del cliente.

### *Costos*

Al definir los costos de adoptar una tecnología específica frecuentemente se comete el error de asociarlos únicamente con la adquisición de los equipos o permisos (Licencias) necesarios, dejando a un lado los costos relativos a la integración con las plataformas existentes, el mantenimiento, la relación de ganancias, entre otros factores. Es necesario destacar que, no solo los derechos de uso son parte importante en la comparación económica. Un codificador puede entregar una calidad de video igual o superior que otro, sin embargo, la relación entre la Calidad y la tasa de compresión es un factor económico importante, a fin de cuentas, el ancho de banda es también un valor a considerar. Además, otros factores como el consumo de recursos de procesamiento y los costes de posibles actualizaciones pueden llegar a ser determinantes en muchos casos. Respecto al uso eficiente del ancho de banda, como en la mayoría de los casos estudiados, Theora 1.0 se mantuvo ligeramente por debajo de los resultados obtenidos por H.264, sin embargo, y a la vista del estudio teórico sobre las capacidades de escalabilidad del códec de MPEG, estos resultados podrían cambiar de forma drástica con el uso eficiente de la tecnología SVC. Respecto al uso de recursos de computo, fue evidente que la complejidad de H.264 trae consigo un aumento en las necesidades de recursos que necesita para desempeñarse. Theora, por su parte, y fiel a la idea de la “simplicidad”, mostró tener un consumo de recursos bastante menor, al menos en las condiciones de las pruebas realizadas.

A pesar de que, como se dijo, no es el único factor, los detalles sobre los costos asociados a los derechos de autor y licencias han colocado a Theora en una posición ventajosa sobre el estándar de MPEG, a menos en cuanto al mercado de Exploradores de Internet se refiere. Las diversas compañías desarrolladoras de este tipo de programas se debaten aún ante la elección de un codificador que se convierta en estándar para las emisiones de video y que pueda ser integrado de forma nativa en sus productos, y, para muchas, la inversión en licencias requeridas para la utilización

de H.264 ha llevado a rechazar de plano este estándar. Evidentemente, siendo tal vez uno de los mercados con mayor crecimiento, la distribución de video sobre Internet y las decisiones que se tomen a este respecto por el grupo de desarrollo de HTML5 marcarán la diferencia en cuanto a cuál códec se convertirá en el más difundido y utilizado.

A nivel institucional, las ventajas económicas de Theora son más que evidentes y sería difícil encontrar excusas para utilizar tecnologías costosas como H.264 en aplicaciones con fines educativos o con cualquier fin no comercial.

### *Final*

Como se mencionó al inicio, la realización de este trabajo no buscaba establecer una jerarquía entre los codificadores en estudio, por el contrario, tuvo como objetivo fundamental exponer la ventajas y desventajas de ambos codificadores, difundir las características del códec “menos conocido” y dar las herramientas a futuros desarrolladores de aplicaciones para elegir, con propiedad, la tecnología de codificación de video que más se adapte a sus necesidades tecnológicas y económicas.

Pese a que este proyecto estuvo más orientado hacia la aplicación en el mercado que a la investigación, es importante destacar las ventajas que en el ámbito de la Investigación y Desarrollo tienen las soluciones como Theora. Sus características de código libre y abierto lo hacen ideal para su uso en aplicaciones científicas y académicas y su gratuidad hacen factible su uso en cualquier tipo de soluciones, comerciales o no, a desarrollar por instituciones de formación.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Study Group XV, Unión Internacional De Telecomunicaciones. *CÓDEC video PARA SERVICIOS ,AUDIOVISUALES A  $p \times 64$  kbit/s - Recomendación UIT-T H.261* . UIT 1994
- [2] Unión Internacional De Telecomunicaciones. *H.262: Information technology - Generic coding of moving pictures and associated audio information: Video.*
- [3] Unión Internacional De Telecomunicaciones. *Infraestructura de los servicios audiovisuales, Codificación de imágenes video en movimiento - Recomendación UIT-T H.263.* Ginebra, 2005.
- [4] MPEG. *MPEG Standards. Full list of standards developed or under development.* - EN: <http://www.chiariglione.org/mpeg/standards.htm> [Consulta: 2009, Octubre]
- [5] Ochoa-Domínguez, J.Mireles-García. *Descripción del nuevo estándar de video H.264 y comparación de su eficiencia de codificación con otros estándares.* Ingeniería. Investigación y Tecnología, enero - marzo, año/vol. VIII, número 001.
- [6] Tektronix, Inc. *A Guide to MPEG Fundamentals and Protocol Analysis Revision 2 Includes Next Generation códecs, High Definition Video and Mobile RF Digital Modulation.* September 2006
- [7] Xiph.Org Foundation – EN: <http://xiph.org> [Consulta: 2009, Octubre]



- [8] Melanson, Mike. *VP3 Bitstream Format and Decoding Process*. December 8, 2004.
- [9] Xiph.org Foundation. *Theora Specification*. April 16, 2008.
- [10] Joskowicz, José. *Estudio de la Medida de la Calidad Perceptual de Video*. 2008.
- [11] Unión Internacional De Telecomunicaciones. *Methodology for the subjective assessment of the quality of television pictures - Recommendation ITU-R BT.500-11*. 2002
- [12] Psannis, Kostas. *Enhanced H.264/AVC stream switching over varying bandwidth networks*. University of Macedonia, Department of Technology Management. 2008.
- [13] MPEGLA. *SUMMARY OF AVC/H.264 LICENSE TERMS*. -  
**EN:** <http://www.mpegla.com>. [Consulta: 2009, Octubre]
- [14] MPEGLA. *AVC Patent Portfolio License - Section 3.1.5*.  
**EN:** <http://www.mpegla.com>. [Consulta: 2009, Octubre]
- [15] MPEGLA. *AVC Patent Portfolio License - Section 3.1.2, 3.1.3, 3.1.4 and 3.1.7*. **EN:** <http://www.mpegla.com>. [Consulta: 2009, Octubre]

## BIBLIOGRAFÍAS

Giles, Ralph. *Ogg Theora a Free Video códec and Multimedia Platform* . Xiph.org Foundation, Vancouver Canada.

Richardson, Iain. *An Overview of H.264 Advanced Video Coding*. Vcodex White Paper. VCODEX 2007.

ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVT- E022, Septiembre 2002 .

Ruiz, Rosalba. *El H.264, Un Nuevo Estándar para Compresión de Video y su Aplicación a la Videoconferencia*. (Paper). 2007

Diamantopoulos, Georgios. *Video Coding Past, Present & Future* . Digital Systems and Vision Processing Group. - EN: <http://postgrad.eee.bham.ac.uk/gxd186/teaching.html>. 2006.

Halbach, Till. *Comparison Of Open and Free Video Compression Systems*. (Paper). Norwegian Computing Center.

Nan, Li. *Mesh Network Testbed and Video Stream Measurement* . (Paper). 2007

Psannis, Kostas. *Enhanced H.264/AVC stream switching over varying bandwidth networks* . IEICE Electronics Express, Vol.5, No.19. 2008.

Zhao, Shuang., Liu, Wenyu. *The Channel Rate-Distortion Model of Video Transmission Based on Bit Error*. 2007 International Symposium on Intelligent Signal Processing and Communication Systems , 2007.

García, Bonifacio. *Desarrollo de un servicio de acceso y presentación de contenidos multimedia personales e interactivos* . (Tesis) - Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación. 2005.

Xiph.org Foundation. *Theora Specification*. April 16, 2008.

Joskowicz, José. *Estudio de la Medida de la Calidad Perceptual de Video*. (Presentación). 2008.

J.Klaue, B.Rathke, and A.Wolisz, *EvalVid - A Framework for Video Transmission and Quality Evaluation*. 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation. 2003.

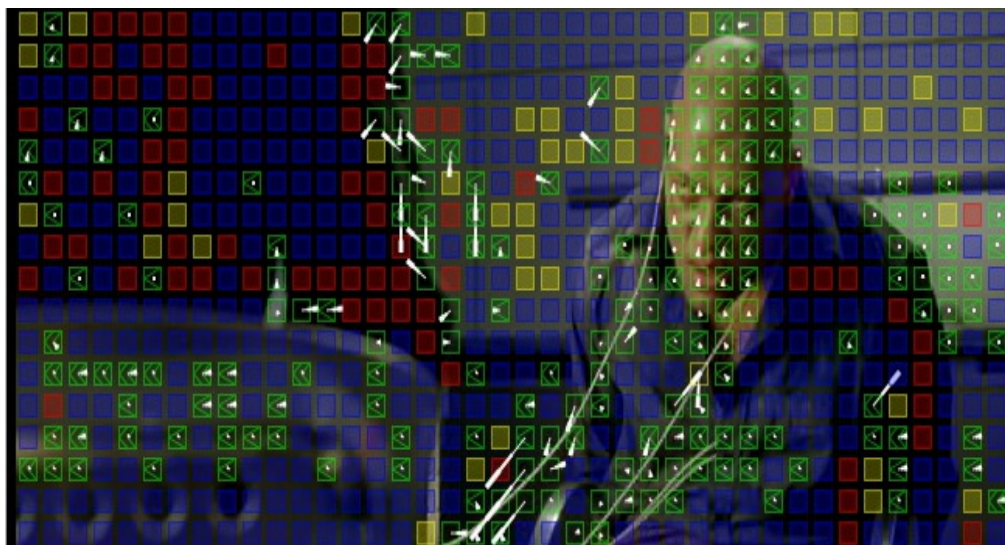
Lie, Arne. *Evalvid-RA: Trace Driven Simulation of Rate Adaptive MPEG-4 VBR Video*. Multimedia Systems, Volumen 14, Número 1, 13. Noviembre 2007.

Keith, Jack. *Video Demystified – A Handbook for the Digital Engineer*. Quinta Edición. Editorial Elsevier. Oxford, UK. 2007

Centro de Electrónica Industrial. *Aplicaciones Reconfigurables para Video Escalable. (Presentación)*. Universidad Politecnica de Madrid. 2o Seminario Anual, mayo 2009

Wang, Z. Bovik, A. Simonelly, E.. *Image quality assessment: From error visibility to structural similarity*. IEEE Transactions on Image Processing, vol. 13. 2004.

**[ANEXO 1]**  
**[División del Cuadro en Macrobloques]**



La imagen muestra un cuadro dividido en Macrobloques, los distintos colores en los que se ha representado cada bloque dependerán del tipo de codificación elegido para el mismo.

**[ANEXO 2]**  
**[Vectores de Movimiento]**



La imagen muestra un cuadro de video codificado en el cual (gracias a la modificación de los parámetros) se han codificado todos los bloques en modo INTRA. Pueden observarse los vectores de movimiento de cada macrobloque.

[<http://web.mit.edu/xiphmont/Public/theora/>]

### [ANEXO 3]

#### [Comparación H.264 – Theora 1.0]



Cuadro Original



Cuadro codificado con H.264



Cuadro codificado con Theora 1.0

[<http://web.mit.edu/xiphmont/Public/theora/>]

## **[ANEXO 4]**

### **[Herramientas – Aplicaciones Utilizadas]**

#### **Evalvid**

Evalvid consiste en una serie de herramientas escritas y desarrolladas por el Dr. Jirka Klaue y cuya función principal es la de realizar evaluaciones sobre la calidad de video transmitido sobre una red (real o simulada). Evalvid está dirigida específicamente a investigadores que necesiten evaluar la calidad de sus redes de datos en términos de la calidad de video que puede ser transmitida por ellas.

Además de incluir herramientas relacionadas con parámetros de QoS de la red (BER, delays, jitter), Evalvid tiene la capacidad de determinar diversos parámetros objetivos de calidad de video como el PSNR y el SSIM.

El código fuente de Evalvid, así como la documentación necesaria para su uso se encuentran disponibles bajo licencia GNU en la pagina WEB del autor.

[<http://www.tkn.tu-berlin.de/research/evalvid/>]

#### **FFMPEG**

FFmpeg es una colección de software libre que puede grabar, convertir y hace streaming de audio y video. Incluye libavcodec, una biblioteca de códecs. FFmpeg está desarrollado en GNU/Linux, pero puede ser compilado en la mayoría de los sistemas operativos, incluyendo Windows.

FFMPEG incluye las instrucciones necesarias para convertir un video de un formato a otro (tanto en su codificación como en su contenedor), permite difundir y

capturar flujos de video bajo redes de datos y gracias a su integración con la librería “libacódec” permite la modificación de una gran cantidad de parámetros de codificación que no son accesibles a herramientas convencionales.

Una de las características resaltantes de ffmpeg es que puede ser compilado junto con diferentes librerías de códecs de video para que estas puedan ser incluidas dentro las funcionalidades típicas del programa. Gracias a esta característica pudo incluirse la codificación con los códecs Theora y Thusnelda para ser analizadas.

El código fuente de ffmpeg, así como la documentación relacionada, puede ser encontrado en la página WEB del proyecto: [<http://ffmpeg.org/>]

## **GPAC**

GPAC es un conjunto de herramientas para el estudio y manejo de contenido multimedia, incluye un reproductor de medios, un encapsulador y algunas subrutinas de servidor de contenido. GPAC es un proyecto bajo los principios de Código Libre y licencia GNU, su código fuente y documentación pueden obtenerse en la página WEB oficial del proyecto:[ <http://gpac.sourceforge.net/>]

## **FFMPEG2THEORA**

FFMPEG2THEORA es la aplicación oficial para convertir y manipular flujos de video en formato OGG y codificados con Theora. La librería se encuentra registrada bajo licencia GNU y su código fuente y documentación puede ser encontrada en: [<http://v2v.cc/~j/ffmpeg2theora/>]



## **VLC MEDIA PLAYER**

VLC Media Player es una aplicación multiplataforma y licenciada bajo GNU. El VLC Media player consiste en un reproductor de diversos formatos multimedia, un conversor (códecs) y empaquetador (contenedores) y un servidor de flujos de video capaz de realizar difusiones unicast/multicast en redes Ipv4 e Ipv6.

El código fuente, así como diversa documentación sobre su compilación, modificación y uso, se encuentra disponible en la página WEB oficial del proyecto: [<http://www.videolan.org/vlc/>]

## **Pyshaper**

Pyshaper es una herramienta escrita en python que permite modificar el ancho de banda disponible en un enlace, incluso entre puertos específicos. Pyshaper es un proyecto GNU, su código fuente y documentación puede ser encontrado en: [<http://www.freenet.org.nz/python/pyshaper/>]

## **[ANEXO 5]**

### **[Equipos Utilizados en las Pruebas]**

#### **Configuración de los equipos (PC) utilizados en las pruebas.**

##### **EQUIPO I**

LAPTOP Toshiba A205

Procesador: Pentium Core 2 Duo 5500

Tarjeta gráfica: Intel X3100m

Memoria: 2Gb, DDR2

##### **EQUIPO II**

PC

Procesador: Pentium Dual Core 3Ghz

Tarjeta gráfica: Intel X4100

Memoria: 4Gb, DDR2

##### **SOFTWARE (AMBOS EQUIPOS)**

Sistema Operativo I: Ubuntu 8.10

Librerías: EvalVid 2.7, GPAC 0.4.1, ffmpeg 4.05, ffmpeg2theora 0.2,  
libtheora 1.01, VLC 1.02

Sistema Operativo II: Windows Vista Ultimate Edition

Aplicaciones: VLC Media Player 1.0