UNIVERSIDAD AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA

INGENIERÍA ELÉCTRICA - TELECOMUNICACIONES

ESTUDIO DE PÉRDIDAS DE PAQUETES PARA LA TRANSMISIÓN DE VIDEO H.264/AVC

TESIS

QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
ING. GABINO BENÍTEZ PATRACA

TUTORES:
DR. VÍCTOR GARCÍA GARDUÑO
DEPARTAMENTO DE INGENIERÍA EN TELECOMUNICACIONES
DR. SUNIL KUMAR
ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT
SAN DIEGO STATE UNIVERSITY

MÉXICO, D.F. MARZO 2014

JURADO

Presidente: DR. FRANCISCO GARCÍA UGALDE

Secretario: DR. JAVIER GOMEZ CASTELLANOS

Vocal: DR. VÍCTOR GARCÍA GARDUÑO

Primer Suplente: DR. BOHUMIL PSENICKA

Segundo Suplente: DR. JOSÉ MARÍA MATÍAS MARURI

Lugares donde se realizó la tesis:

Universidad Nacional Autónoma de México, D.F.

San Diego State University, California E.U.A.

TUTOR DE TESIS

DR. VÍCTOR GARCÍA GARDUÑO

FIRMA

If we knew what it was we were doing, it would not be called research, would it?

Albert Einstein

Con todo mi cariño y amor para esas personas
que han dado todo por mi, porque con paciencia y comprensión
me han guiado para que yo pudiera realizar uno de mis sueños.

A mi Madre y a mi Padre

Agradecimientos

Quiero agradecer en primer lugar a Dios por acompañarme cada día, por estar en cada momento junto a mi. En los momentos difíciles me mostraste la luz y me diste fuerza para levantarme.

Gracias mamá por siempre creer y confiar en mi. Tus palabras me han motivado a seguir adelante. Gracias por el apoyo incondicional que siempre me brindas. Tu fuiste mi primera maestra y la mejor, nunca me has dejado de enseñar lo valioso de esta vida. Sin importar la distancia ni el tiempo eres mi mas grande tesoro.

También a ti papá, gracias por haberme enseñado el valor de la constancia. Gracias por haberme dado la oportunidad de crecer y luchar por mis objetivos. Sin ti nada de esto hubiese sido posible.

Gracias a mis hermanas Yesi, Leydi, Leyda y Mireya que me han brindado la motivación para estar donde estoy hoy. Son mi fortaleza y mi inspiración.

Gracias a mi asesor, el Dr. Víctor, por apoyarme en este trabajo, por darme todas esas oportunidades y la posibilidad de trabajar en algo que realmente disfruto. Gracias al Dr. Sunil por abrirme las puertas en la universidad de San Diego.

Quiero agradecer a todas las personas que hicieron de mi estancia en San Diego una colección de momentos inolvidables, gracias Griss, Gail, Angeliki y Fa. Siempre tendrán un lugar muy especial dentro de mi.

Abstrac

The video applications over wireless networks, such as video streaming are incresing rapidly. In order to efficiently utilize the wireless bandwidth, video data is compressed using sophisticated video coding techniques such as H.264. On this thesis different slices dropping schemes in terms on their effect on the quality of H.264/AVC encoded video are studied. In these schemes, It is taken into account the priority, slice groups, frame type and frame location. The main objective of this thesis is to understand slice dropping effects and propose a dropping policy that lead to minimum video distortion when it is required to drop slices or reduce the bit rate.

Resumen

Las aplicaciones de video sobre redes inalámbricas, como streaming de video, están creciendo a un ritmo acelerado. Con el fin de utilizar eficientemente el ancho de banda, la información de video es comprimida usando técnicas sofisticadas de codificación como lo es el estándar H.264. En esta tesis, diferentes esquemas de pérdidas de *slices* son estudiadas tomando en consideración sus efectos en la calidad del video codificado. Se toman en cuenta aspectos como la prioridad, grupos de *slices*, tipo de imagen y su localización. El principal objetivo de la tesis es comprender los efectos de pérdidas de *slices* y proponer una política de pérdidas que introduzca la menor distorsión cuando se requiera reducir el *bit rate*.

Índice general

Índice de figuras

Índice de tablas

Lista de Acrónimos

ASO Arbitrary Slice Order.

BER Bit Error Rate.

CABAC Context-adaptive Binary Arithmetic Coding.

CAVLC Context-Adaptive Variable Length Coding.

CIF Common Intermediate Format.

CMSE Cumulative Mean Squared Error.

DCT Discrete Cosine Transform.

DPB Decoding Picture Buffer.

FLC Fixed Length Coding.

FMO Flexible Macroblock Order.

GOP Group of Pictures.

HD High Definition.

IDR Instantaneous Decoding Refresh.

ITU-T International Telecommunication Union.

MB Macroblock.

MOS Mean Opinion Score.

MPEG Moving Picture Expert Group.

MSE Mean Squared Error.

NAL Network Abstraction Layer.

NALU Network Abstraction Layer Unit.

PPS Picture Parameter Set.

PSNR Peak Signal-to-Noise Ratio.

QP Quatization Parameter.

RBSP Raw Byte Sequence Payload.

RDO Rate-Distortion Optimization.

ROI Region of Interest.

RTP Real-time Transport Protocol.

SAE Sum of Absolute Differences.

SD Standard Definition.

SG Slice Group.

SPS Secuence Parameter Set.

SSIM Structural Similarity Index.

SVH Sistema Visual Humano.

UEP Unequal Error Protection.

UVLC Universal Variable Length Coding.

VBR Variable Bit Rate.

VCL Video Coding Layer.

VLC Variable Length Coding.

VQM Video Quality Metric.

Capítulo 1

Introducción

H.264/AVC es el estándar de codificación más reciente del Grupo de Expertos de Codificación de video de la International Telecommunication Union (ITU-T) y el Moving Picture Expert Group (MPEG). Las características principales del estándar de video son su excelente desempeño en compresión y la adaptación versátil de la representación del video a la red de transporte para aplicaciones de tiempo real (video telefonía) y otras donde los retardos no tienen alto impacto (almacenamiento, difusión o streaming). El estándar ha logrado mejoras significativas en la eficiencia de la tasa de distorsión en comparación con los estándares existentes. H.264/AVC logra esta compresión explotando la redundancia temporal y espacial bloque a bloque dentro del video. La calidad del video comprimido se degrada significativamente debido a las pérdidas en el canal cuando los datos son transmitidos en canales inalámbricos propensos a errores.

1.1. Motivación

Actualmente existe un crecimiento en la demanda de aplicaciones de video lo que se traduce en una creciente necesidad de transmitir video de alta calidad sobre canales inalámbricos. Se requieren esquemas protectores para empaquetar video y así distribuir los recursos del sistema con el fin de minimizar la distorsión en el receptor. Para minimizar el error sufrido en la transmisión de la secuencia codificada, técnicas de Unequal Error Protection (UEP) han sido usadas de tal forma que bajo ambientes ruidosos la calidad del video recibido sea relativamente aceptable para el usuario final.

El estándar de video H.264/AVC cuenta con varias herramientas de resistencia al error [1] que pueden ser explotadas para particionar el flujo de bits en diferentes capas de acuerdo a su importancia y dar protección por consiguiente a través de códigos correctores de errores. Actualmente los métodos de UEP soportados son la partición de datos, codificación escalable, diferenciación de

cuadros de video y diferenciación de posiciones de bloques de imágenes [1]. Cada método de UEP es único y combinándolos es posible mejorar su robustez ante errores.

En transmisiones inalámbricas de video, los datos de video comprimido son transmitidos en forma de paquetes. Cuando esos paquetes son perdidos aleatoriamente durante la transmisión o desechados debido a congestión en la red, la distorsión visual empieza a aparecer en el video, degradando la calidad del contenido e incluso haciéndolo irreconocible para el decodificador del receptor.

Debido al uso de codificación de longitud variable en el estándar H.264/AVC a fin de lograr altas tasas de compresión, incluso la pérdida de un único bit o un bit en error puede conducir a tener que descartar completamente el paquete de datos [2]. Durante el proceso de codificación, la información de un cuadro de video puede ser usada para predecir uno o más cuadros subsecuentes o anteriores, esto implica que en el proceso de decodificación el error de un cuadro se puede propagar a varios más.

Se propone un esquema que sea capaz de mejorar la robustez del video ante los problemas expuestos anteriormente combinando varias características de resistencia al error del estándar H.264/AVC.

1.2. Importancia del estándar H.264/AVC

El estándar H.264/AVC posee gran transcendencia para la Internet, servicios de broadcast, el mercado de consumidores de electrónicos y para las industrias de dispositivos móviles y de seguridad entre tantas más. H.264/AVC describe y define un método de codificar video que brinda el mejor desempeño que cualquier estándar anterior a él. H.264/AVC hace posible comprimir video para que ocupe menos espacio, por lo cual una secuencia de video comprimido emplea menor ancho de banda para su transmisión y/o menor espacio de almacenamiento comparado con estándares previos.

Una combinación de avances tecnológicos e incremento de la expectativa del usuario final conlleva demandas de video digital de mayor calidad. Como muestra de esto, se encuentran compañías de televisión ofreciendo contenidos en alta definición y las ampliamente usadas, video llamadas a través de Internet que introducen una alta carga de tráfico. La compresión estandarizada de video hace posible la inter-operabilidad para productos de diferentes fabricantes como codificadores, decodificadores y almacenamiento multimedia.

Toda la gama de beneficios de H.264/AVC evidentemente implica un costo. El estándar resulta ser una tarea compleja y desafiante para quienes buscan desarrollar implementaciones e interfaces con el estándar. Debido a su alto costo computacional, un codificador H.264 puede inducir tiempos de codificación y decodificación lentos o drenaje rápido de la batería en dispositivos móviles.

[1] También llamado Región de Interés (Region of Interest (ROI) por sus siglas en ingles)

1.3. Objetivo

En este documento se tiene como primer objetivo estudiar el estándar de codificación H.264/AVC y comprender de forma general las interdependencias existentes entre las etapas de la codificación y los parámetros de configuración para conocer cualitativamente el alcance de los pérdidas en una secuencia de video, independientemente de cual sea el origen de la pérdida.

Aunque en la actualidad son más confiables las redes de comunicación, no es posible garantizar la aparición de pérdidas de paquetes. Las pérdidas de paquetes de video tienen un diferente impacto en la calidad del video por lo que sería deseable minimizar la degradación del video por algún método.

El segundo objetivo principal de esta tesis se basa en la medición del impacto de pérdidas en la calidad del video con el fin de establecer niveles de prioridad a los paquetes de video codificado. Adicionalmente se busca diseñar un patrón de pérdidas de *slices* donde se establezca una clasificación de los *slices* de video que tiene como objetivo señalizar que tipo de paquetes minimizan la degradación de la calidad cuando se considera un porcentaje de pérdidas en la secuencia de video. Este patrón sera nombrado posteriormente como **política de pérdidas**.

1.4. Estructura de la tesis

Esta tesis está organizada de la siguiente forma:

En el segundo capítulo se describe el modelo a bloques del funcionamiento del estándar H.264. Aquí se muestran los principales procesos que componen el codificador y decodificador. La mayoría de las tareas desarrolladas en el codificador cuentan con su correspondiente tarea inversa en decodificador y ocurren en orden contrario.

El tercer capítulo está enfocado a explicar las características de resistencia al error con las que cuenta H.264, estas serán aprovechadas en el estudio de pérdidas. Se explica como estas herramientas mejorarían la robustez del video ante pérdidas durante su transmisión.

En el cuarto capítulo se muestran las configuraciones de codificación de video seleccionadas y las secuencias de prueba. Se describe la forma en que se introducen las pérdidas en las secuencias y la forma en que se asignan prioridades a los *slices* en función de la calidad medida.

En el quinto capítulo se proponen patrones de pérdidas, se evalúa el desempeño a través del progreso de la política e identifican las tendencias que surgen. Además se realiza una comparación del desempeño política con un esquema de pérdidas aleatorias para corroborar los beneficios de la política propuesta.

En el sexto y ultimo capitulo se exponen las conclusiones finales, la utilidad y el potencial de desarrollo de este trabajo. Así también, se habla de las directrices a futuro que se pueden derivar del mismo.

Capítulo 2

Modelo Básico de H.264/AVC

Este capítulo introduce el estándar H.264/AVC y una descripción funcional del mismo. El significado que tiene este estándar puede verse desde distintas perspectivas. Es un estándar industrial. Define una formato para datos de video comprimido. Provee un conjunto de herramientas que pueden ser usadas en una variedad de formas para comprimir y comunicar información visual. Además es una etapa de la evolución de serie de métodos estandarizados de compresión de video.

El video digital es una representación de escenas del mundo real muestreadas en puntos específicos en el tiempo para producir una secuencia continua de imágenes. La codificación de video es el proceso de compresión y descompresión de una señal de video digital, la compresión busca convertir la señal para que tenga un formato apropiado en su transmisión y almacenamiento. La compresión de la señal se logra eliminando la redundancia. El estándar H.264/AVC especifica como un video codificado debe ser representado y decodificado como lo muestra la figura 2.1.

2.1. Descripción Funcional

Cada estándar de codificación debe estar representado por un CODEC (par de codificador y decodificador) que represente la secuencia de video original por medio de un modelo, una eficiente representación codificada que puede ser usada para reconstruir una aproximación de los datos del video. Idealmente el modelo debe representar la secuencia usando la menor cantidad de bits como sea posible y con la mayor fidelidad permisible. Estos dos objetivos, eficiencia de compresión y alta calidad son generalmente contrapuestos.

El codificador de video H.264/AVC funcionalmente consiste de tres unidades conectadas: un *un modelo de predicción*, un *modelo espacial* y un *codificador*

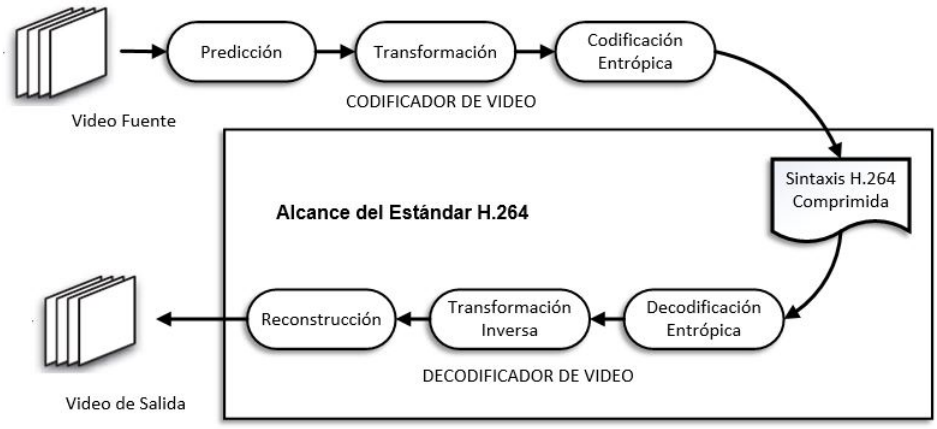


Figura 2.1: Procesos de codificación y decodificación de video H.264

entrópico. La entrada al modelo de predicción es una secuencia de video bruta o sin comprimir. El modelo de predicción trata de reducir la redundancia explotando las similitudes en cuadros de imagen vecinos y muestras vecinas de una imagen, frecuentemente a través de la construcción de una predicción formada de cuadros de video, tanto previos como futuros y bloques de muestras de la imagen actual. La predicción se crea por medio de la extrapolación espacial de muestras vecinas de una imagen en particular, proceso que es llamado predicción intra, o a través de compensar las diferencias entre imágenes cercanas, conocido como predicción inter o compensada en movimiento. La salida del modelo de predicción es un cuadro de imagen residual que se crea substrayendo los valores de predicción de los valores del cuadro actual, además también se generan parámetros que indican el uso predicción intra o inter y la descripción de como el movimiento fue compensado.

El cuadro de imagen residual forma la entrada al modelo espacial, que aprovecha las similitudes entre las muestras locales en la imagen residual para reducir la redundancia espacial. En el estándar se logra aplicando una transformación a las muestras residuales y cuantizando el resultado. La transformada convierte las muestras a otro dominio, en el cual las muestras son representadas por coeficientes de la transformada. Los coeficientes son cuantizados para remover valores insignificantes o despreciables, dejando solo una pequeña cantidad de coeficientes significativos que proveen una representación más compacta de la imagen residual. La salida del modelo espacial es el conjunto de los coeficientes de la transformada cuantizados.

Los parámetros de la primera y segunda unidad del codificador como los modos de predicción intra e inter, vectores de movimiento y coeficientes de transformada, entre otros, son comprimidos por el codificador entrópico. Esta ultima etapa remueve la redundancia estadística de los datos, por ejemplo representando vectores y coeficientes que ocurren con frecuencia por medio de códigos

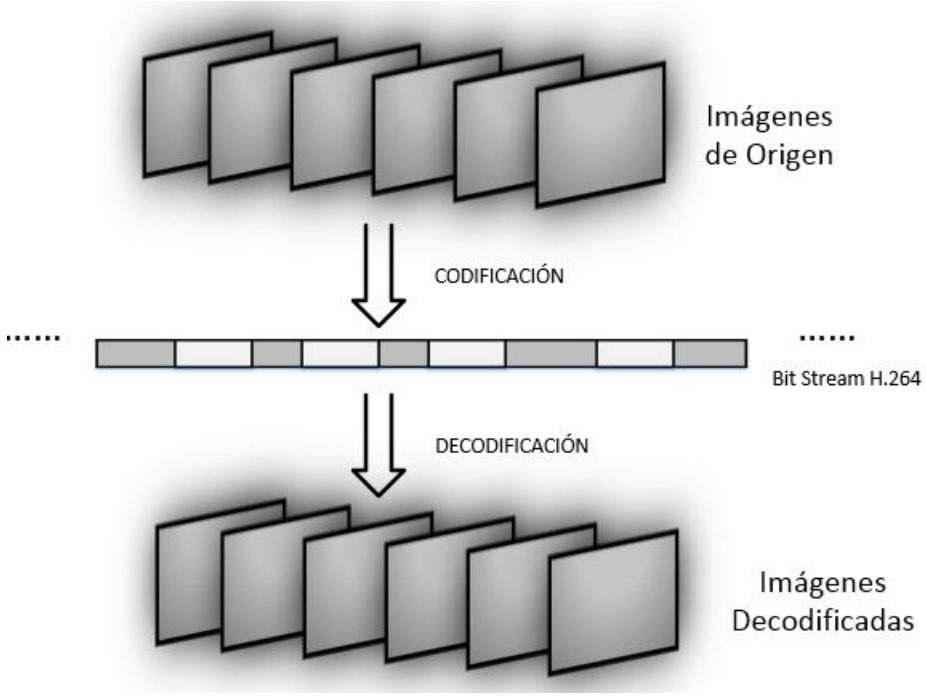


Figura 2.2: Codificación de video: Secuencia fuente, Bitstream codificado, Secuencia decodificada

binarios de corta longitud. La salida de codificador entrópico es un bit stream comprimido o un archivo que puede ser transmitido o almacenado. Finalmente una secuencia comprimida consiste de parámetros de predicción codificados, coeficientes residuales e información de cabecera.

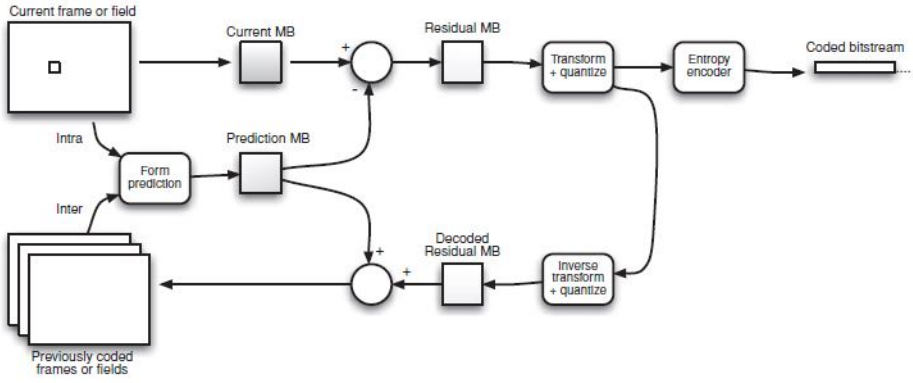
El decodificador de video reconstruye un cuadro de imagen del *bit stream* comprimido. Los coeficientes y parámetros de predicción son decodificados por un decodificador entrópico, posteriormente el modelo espacial se decodifica para reconstruir una versión residual de un cuadro de imagen. El decodificador usa los parámetros de predicción conjuntamente con las muestras antes decodificadas para crear la predicción. El cuadro de imagen final es formado sumando la imagen residual y la predicción.

Un codificador de video H.264 lleva a cabo procesos de predicción, transformación y codificación para producir la representación del video en la sintaxis de H.264 como se muestra en la Figura 2.1. El decodificador de video H.264 lleva a cabo los procesos análogos complementarios de decodificación inversa y reconstrucción para generar la secuencia de video codificado para su reproducción.

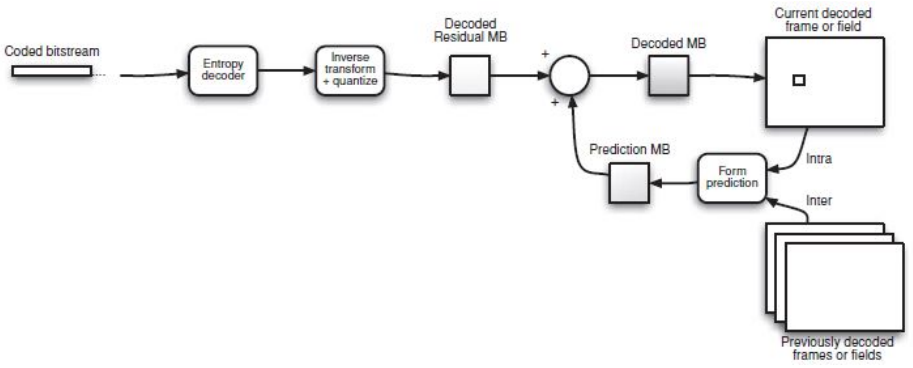
Como en la Figura 2.2 se muestra, cuando una secuencia de video original es codificada en formato H.264, una serie de bits representan al video de for-

ma comprimida. Este flujo de datos comprimido es almacenado o transmitido y puede se decodificado para reconstruir el video. En general, la versión decodificada no es idéntica a la original ya que H.264 es un estándar de compresión con pérdidas por lo que la calidad de algunas imágenes se degrada durante el proceso de compresión.

La versatilidad de la adaptación del estándar H.264/AVC se debe a la separación conceptual entre la Capa de Codificación de video (VLC), la cual provee el núcleo para la representación del video con altas tasas de compresión y la Capa de Abstracción de Red (NAL) por sus siglas en ingles, que empaqueta esa representación para su entrega sobre algún tipo de red en particular. Las características anteriores pueden ser traducidas a un gran número de ventajas para distintas aplicaciones de video. El enfoque del estándar H.264/AVC es similar a los adoptados en estándares previos como el H.263 y el MPEG-4, que consisten en los siguientes etapas principales:



(a) Codificador



(b) Decodificador

Figura 2.3: Diagrama a bloques del par complementario CODEC H.264.

1. Dividir cada cuadro de video en bloques de píxeles, tal que el procesamiento de los cuadros de video pueda ser manejado a nivel de bloques usando una versión modificada de la transformada coseno discreta (Discrete Cosine Transform (DCT)) a la que se llamara por cuestiones practicas transformada entera.

2. Explotar las redundancias espaciales que existen dentro de un cuadro de video codificando algunos de los bloques originales a través de aplicar la DCT, cuantización y codificación entrópica o codificación de longitud variable.

3. Explotar las dependencias que existen entre bloques de cuadros de video sucesivos, tal que solo los cambios entre cuadros sucesivos necesiten ser codificados. Esto se logra usando estimación de movimiento y compresión. Para cualquier bloque se desarrolla una búsqueda en uno o varios cuadros de video previamente codificados para determinar los vectores de movimiento que serán usados por el decodificador en la predicción el bloque actual.

4. Explotar cualquier redundancia espacial restante que exista dentro de un cuadro de video por medio de la codificación de los bloques residuales, o dicho de otra forma, la diferencia entre el bloque original y el correspondiente bloque predicho, de igual forma a través de la transformada DCT, cuantización y codificación entrópica. Desde el punto de vista de codificación. Para la estimación y compensación de movimiento, H.264 emplea bloques de diferentes tamaños y formas, mayor resolución, estimación de movimiento de fracciones de pixel y selección de múltiples cuadros de referencia. Por otro lado H.264 usa transformada entera que aproxima la transformada coseno usada en estándares anteriores pero que no sufre el problema de desajuste de coeficientes en la transformada inversa. En H.264 la codificación entrópica puede ser realizada usando Codificación Universal de Longitud Variable (UVLC) y Codificación Aritmética Variable Adaptable al Contexto (CABAC).

En común con estándares de codificación previos, H.264 no define explicitamente un CODEC (la pareja codificador y decodificador) sino que define la sintaxis del *bitstream* de video codificado, conjuntamente con el método de decodificación de este *bitstream* como se observa en la Figura 2.1.

La estructura de un codificador es mostrada en la Figura 2.3. Los datos son procesados en unidades de *macrobloques* (MB) correspondientes a 16×16 píxeles. En el codificador, un macrobloque de predicción es generado y substraído del macrobloque actual para formar un macrobloque residual; para posteriormente transformarlo, cuantizarlo y codificarlo. En paralelo, los datos cuantizados son re-escalados, se les aplica la transformada inversa y son sumados a los macrobloques de predicción para reconstruir una versión codificada de la imagen que sera almacenada para futuros procesos de predicción. En el decodificador, un

macrobloque se decodifica, re-escala y aplica la transformada inversa para formar el macrobloque residual. El decodificador genera la misma predicción que fue creada en el codificador y la suma al residual para producir un macrobloque decodificado.

2.2. Procesos del Codificador

2.2.1. Predicción

H.264 soporta un amplio rango de opciones de predicción como la predicción intra, predicción inter, múltiples tamaños del bloque de predicción, múltiples imágenes de referencia y modos especiales de predicción como el modo directo y predicción ponderada.

A diferencia de estándares previos (H.263 y MPEG-4 Visual) donde la predicción intra se lleva a cabo en el dominio de la transformada, la predicción intra en H.264 siempre se realiza en el dominio espacial. La figura 2.4 muestra las fuentes de predicción de tres tipos distintos de macrobloques, tipos I, P y B.

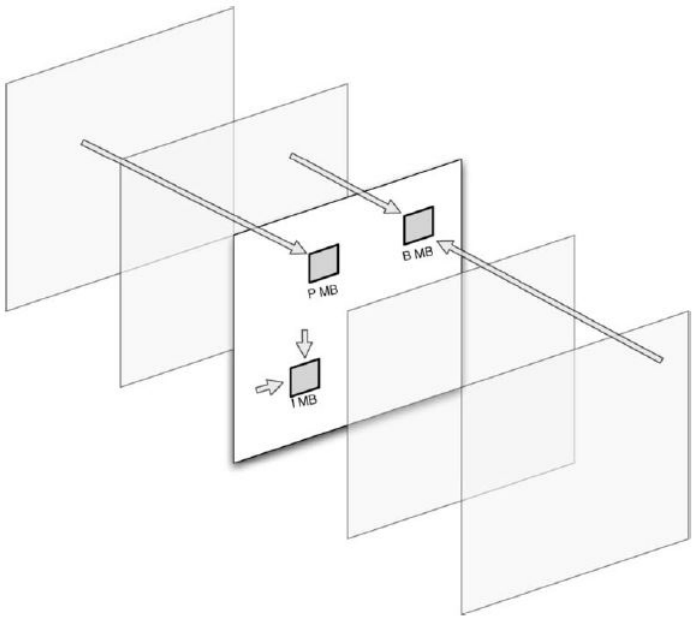


Figura 2.4: Tipos de predicción y sus fuentes de origen

El codificador forma una predicción del macrobloque actual basado en los datos previamente codificados, provenientes de la imagen actual usando predicción intra o de otras imágenes anteriormente codificadas usando predicción

inter. El codificador substrae la predicción del macrobloque actual para formar un bloque residual Figura 2.5. Encontrar una predicción inter apropiada es usualmente descrito como un proceso de *estimación de movimiento* y substraer una predicción inter de un macrobloque es llamado *compensación de movimiento.*

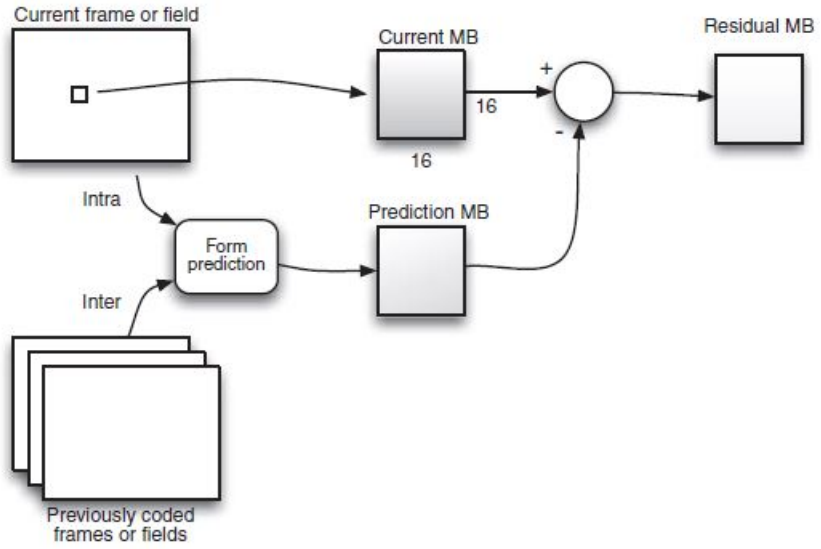


Figura 2.5: Diagrama a bloques de la Predicción

Predicción Intra

La predicción intra usa tamaños de bloques en luminancia de 16×16, 8×8 y 4×4 para predecir un macrobloque a partir de los píxeles circundantes previamente codificados en la misma imagen o dicho de otra forma este tipo de predicción los macrobloques son codificados sin hacer referencia a ningún dato afuera del slice actual. Los valores de píxeles vecinos son extrapolados para obtener una predicción del macrobloque actual. Cada macrobloque en un slice tipo I es un macrobloque tipo I. Para un bloque típico de luminancia o crominancia existe relativamente alta correlación entre muestras en el bloque y muestras inmediatamente adyacentes del bloque. Por lo tanto, la predicción intra usa muestras de bloques adyacentes para predecir los valores del bloque actual. Un bloque de predicción es generado por cada componente de croma. Cada bloque generado tiene diferentes número de opciones de predicción de acuerdo a su tamaño y componente (Tabla 2.1).

La predicción de un bloque es creada directamente de muestras superiores, izquierdas laterales o una combinación de ellas. Solo aquellas muestras que estén

Tamaño del bloque de predicción	*Modos de predicción*
16 x 16 (luma)	Cuatro posibles modos de predicción
8 x 8 (luma)	Nueve posibles modos de predicción. Solo High Profile.
4 x 4 (luma)	Nueve posibles modos de predicción
Croma	Un bloque de predicción es generado por cada componente. Cuatro modos de predicción. El mismo modo es usado para ambos componentes.

Tabla 2.1: Tipos de Predicción Intra

en realidad disponibles pueden ser usadas para formar la predicción. Por ejemplo, un bloque localizado en el margen izquierdo de la imagen o slice no cuenta con muestras vecinas del lado izquierdo, por lo que ciertos modos de predicción no estarán disponibles. El codificador selecciona algún otro modo de predicción disponible para el bloque actual.

La selección del tamaño del bloque de predicción para el componente de luma de los tres disponibles tiende a ser un compromiso entre eficiencia de la predicción y el costo implicado en la señalización del modo de predicción. Un bloque de predicción más pequeño (i.e. 4×4)tiende a dar más precisión en la predicción por lo que la predicción se aproxima más a los valores originales. Esto implica bloques residuales más pequeños tal que menos bits son requeridos para codificar los coeficientes transformados de los bloques residuales. No obstante la selección de predicción de bloques de 4×4 debe ser señalizada al decodificador por lo que se tienden a requerir más bits para codificar la selección realizada.

Un bloque de predicción de mayor tamaño tiende a dar una predicción menos precisa, por lo tanto más datos residuales son producidos pero menos bit son requeridos para codificar la selección de predicción. En conclusión el codificador en general elegirá un modo de predicción intra disponible para minimizar el total de bits de la predicción y los bloques residuales.

El codificador selecciona para cada bloque el mejor modo de predicción basándose en la función *suma de errores absolutos* (SAE). Esto quiere decir que un bloque sera predicho usando cada modo de predicción para posteriormente ser comparado pixel a pixel con el bloque a que corresponda dicha predicción. La SAE indicara la magnitud del error de predicción. De esta manera se elegirá la mejor modo para ese bloque porqué genera el menor SAE.

En la figura 2.7 están mostrados los modos de predicción para bloques de 8×8 píxeles, para los bloques de 4×4 se utilizan los mismos modos en una versión escalada. Las muestras de arriba y a la izquierda, etiquetadas con las letras A-Z han sido previamente codificadas y reconstruidas, por ende están disponibles para formar una predicción de referencia. Las flechas indican la

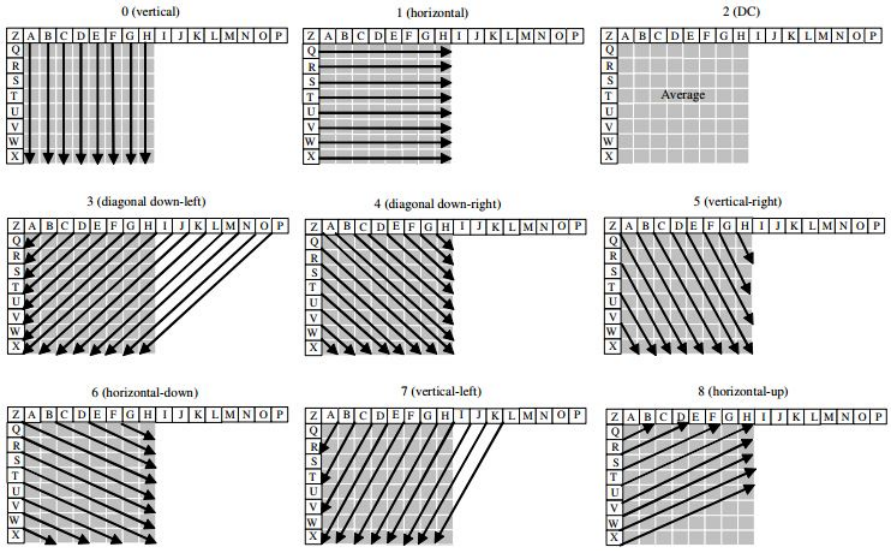


Figura 2.6: Modos de predicción Intra 8×8

dirección de predicción de cada modo. Para los modos 3-8 las muestras predichas son formadas a través de un promedio ponderado de los valores A-Z.

Los bloques de 16×16 solo cuentan cuatro posibles modos de predicción, que corresponden a los modos 0, 1 y 2 de los mostrados en la figura 2.7 más un modo planar. Los componentes de croma también usan los mismo cuatro modos de predicción. Cabe resaltar que ambos componentes de croma siempre usan el mismo modo.

Predicción Inter

La predicción Inter es el proceso de predecir un bloque de muestras a partir de una imagen previa que ya ha sido codificada y transmitida, una imagen de referencia. Este proceso involucra seleccionar una región de predicción, generar el bloque de predicción y sustraerlo del bloque original de muestras, así formar un bloque residual para su posterior codificación. El bloque de muestras a ser predicho, una partición de un macrobloque, puede oscilar desde el macrobloque completo (16×16 píxeles) hasta bloques de 4×4. Las imágenes ya codificadas son almacenadas en un *buffer*, que puede incluir imágenes anteriores o posteriores a la imagen actual en orden de reproducción. La diferencia de las posiciones entre el bloque actual y el bloque de predicción en la imagen de referencia es llamado vector de movimiento. Los vectores de movimiento pueden apuntar a un posición especifica hasta con una resolución de un cuarto de pixel[1], característica que representa una de las mayores mejoras de H.264 con los anteriores estándares.

[1]En el componente de luma se tiene una precisión de un cuarto de pixel

Cada vector de movimiento es codificado diferencialmente a partir de los vectores de los bloques vecinos.

De forma opcional el bloque de predicción puede ser ponderado de acuerdo a la distancia temporal entre la imagen de referencia y la imagen actual. En los macrobloques tipo B, un bloque puede ser predicho en modo directo, en tal caso ningún bloque residual o vector de movimiento es enviado y el decodificador infiere los vectores de movimiento de los vectores anteriormente recibidos.

Para resumir el proceso de codificar un macrobloque con predicción inter se pueden seguir los siguientes pasos, es necesario aclarar que los pasos no necesariamente pueden ocurrir en el siguiente orden:

1. Interpolar las imágenes en el *buffer* de imágenes decodificadas (DPB) para generar posiciones de un cuarto de pixel para el componente de luma y de un octavo de pixel para los componentes de croma.

2. Elegir un modo de predicción basado en las siguientes tareas:

   a) Elección de las imágenes de referencia disponibles en DPB.

   b) Elección las particiones de macrobloque (tamaño de los bloques de predicción).

3. Elegir tipos de predicción:

   a) Predicción de una imagen de referencia en la Lista 0 para macrobloques P o B, o Lista 1 solo para macrobloques B.

   b) Bi-predicción de dos imágenes de referencia, una de la Lista 0 y otra de la Lista 1, exclusivamente para macrobloques B. De forma opcional se puede hacer uso de la predicción ponderada.

4. Seleccionar el o los vectores de movimiento para cada partición de un macrobloque, uno o dos vectores de movimientos para cada bloque dependiendo si se usan una o dos imágenes de referencia.

5. Predecir los vectores de movimiento de los vectores previamente codificados y generar vectores diferenciales. Opcionalmente se puede usar predicción en modo directo solo para los macrobloques B.

6. Codificar el tipo macrobloque, la elección de referencias de predicción, los vectores de movimiento diferenciales y residuales.

7. Aplicar un filtro a la imagen para eliminar problemas de formación de bloques previo a su colocación en DPB para referencias posteriores.

Lista	*Descripción*
List0 (*slices* P)	Una sola lista de todas las imágenes de referencia. Por default, la primera imagen en la Lista es la imagen más recientemente decodificada.
List0 (*slices* B)	Una lista de las imágenes de referencia previas a la imagen actual en orden de reproducción.
List1 (*slices* B)	Lista de las imágenes de referencia posteriores a la imagen actual en orden de reproducción

Tabla 2.2: Listas de Predicción

Imágenes de Referencia

Los *slices* recibidos en el decodificador generan imágenes para ser desplegadas. Adicionalmente también son almacenadas en el DPB para ser usadas como referencia. Las imágenes en el DPB son indexadas (listadas en un orden definido) en las siguientes listas, dependiendo si el macrobloque actual está en un slice P o B.

Un macrobloque P siempre usará la lista List0 (*Slices* P). Dentro de un slice tipo B pueden existir macrobloques codificados tipo P o B de acuerdo a su tipo de predicción. Dentro del slice B si el macrobloque es de tipo P hará uso de la lista List0 (*Slices* B), si es de tipo B hará uso de ambas listas para *slices* B.

El orden en las listas es importante, las imágenes de referencia más cercanas temporalmente a una imagen actual aparecerán primero en la lista ya que es más probable que contengan la mejor aproximación de predicción.

Particiones de Macrobloques

Cada macrobloque tipo P o B puede ser predicho usando una variedad de tamaños de bloques. El MB es dividido en una, dos o cuatro particiones:

a) una partición de 16×16 píxeles, cubriendo el MB completo.

b) dos particiones de 8×16.

c) dos particiones de 16×8.

d) cuatro particiones de 8×8.

Si se elige un tamaño de partición de bloque de 8×8 muestras de luma y sus correspondientes muestras de croma se le llama sub-macrobloque, que a su vez puede ser sub-dividido.

Cada partición y sub-particiones de macrobloque tienen uno o dos vectores de movimiento (x, y), cada uno apuntando a un área del mismo tamaño en una imagen de referencia que es usada para predecir la partición actual. Una

partición en un MB tipo P tienen una imagen de referencia y un vector de movimiento asociado. Cuando se trata de un MB tipo B se tiene una o dos imágenes de referencia y uno o dos vectores correspondientemente.

La predicción compensada en movimiento tiende a ser más precisa cuando se hace uso de particiones de tamaño pequeño, especialmente cuando el movimiento es relativamente complejo. Sin embargo, más particiones en un MB significa que más bits deben ser utilizados para representar los vectores de movimiento y las particiones. Frecuentemente el codificador elige particiones grandes para áreas homogéneas de una imagen con textura suave o bajo movimiento y particiones pequeñas donde el movimiento es más complejo.

2.2.2. Transformación y Cuantización

El proceso de predicción en el estándar H.264 se realiza sin pérdida de información por lo que es un proceso totalmente reversible. Como ya se ha mencionado previamente H.264 es un estándar de compresión con pérdidas, estas pérdidas causan distorsión visual o decremento de la calidad. Esta distorsión ocurre en el proceso de transformación y cuantización. Después de los procesos de predicción, transformación y cuantización, la señal de video es representada como una serie de coeficientes de transformación cuantizados conjuntamente con los parámetros de predicción. Esos valores deben ser codificados en un s usando diferentes mecanismos.

La transformación inversa y el re-escalamiento o cuantización inversa están definidos en el estándar H.264. Esos procesos o sus equivalentes deben ser implementados en cada decodificador. Los procesos equivalentes para codificador como la transformación, no están estandarizados pero pueden ser derivados.

En un codificador H.264, un bloque de coeficientes residuales es transformado y cuantizado. La transformación central del estándar es una *transformada entera* de 4×4 u 8×8, una versión escalada de la Transformada Coseno Discreta DCT. En efecto la ecuación 2.1 define una DCT inversa para un bloque de muestras de tamaño $N \times N$, donde Y_{xy} son los coeficientes de entrada y X_{ij} son las muestras de la salida en la imagen.

$$X_{ij} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_x C_y Y_{xy} \cos \frac{(2j+1)y\pi}{2N} \cos \frac{(2j+1)x\pi}{2N} \tag{2.1}$$

La implementación de la ecuación 2.1 para un valor de $N > 2$ en un procesador requiere de aproximaciones a ciertos factores multiplicatorios irracionales. Distintas aproximaciones pueden alterar significativamente la salida de la transformada conduciendo a un problema de desajuste (salidas distintas) entre distintos codificadores o decodificadores. H.264 elimina el problema de desajuste recurriendo a la transformada entera, tal que cada implementación de H.264 produzca resultados idénticos.

La transformada entera involucra sumas, corrimientos de bits y es necesario el uso de multiplicatorias, esto con el fin de minimizar su complejidad computacional. La entrada de la transformación es una serie de valores residuales de píxeles $X = \{x_{00}, x_{01,...x_{33}}\}$ para obtener como resultado los coeficientes $Y = \{y_{00}, y_{01}, ...y_{33}\}$ definidos por la ecuación 2.2.

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \quad (2.2)$$

Esta matriz de transformación es usada en todas las transformaciones de bloques de 4×4, con excepción del bloque de coeficientes de DC en codificaciones intra de 16×16 donde se ocupa la transformada Walsh Hadamard. La transformada inversa de los coeficientes normalizados $Y' = \{y'_{00}, y'_{01}, ...y'_{33}\}$ para obtener valores en el dominio espacial está definida por la ecuación 2.3.

$$X' = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & \frac{-1}{2} & -1 & 1 \\ 1 & -1 & 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & \frac{-1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & \frac{-1}{2} \end{bmatrix} \quad (2.3)$$

Un bloque de muestras residuales es transformado usando la transformada entera. La salida de la transformada es un conjunto de coeficientes, cada uno de los cuales es un valor ponderado de un patrón base estándar. Cuando se combinan los patrones bases recrean el bloque de muestras residuales.

La salida de la transformada, el bloque de coeficientes es cuantizado, cada valor es dividido por un valor entero conocido como parámetro de cuantización (Quatization Parameter). La cuantización reduce la precisión de los coeficientes de transformación de acuerdo a la magnitud del QP. Usualmente el resultado es un bloque donde la mayoría de los coeficientes son cero, con solo pocos coeficientes distintos de cero. Configurar el QP a un valor alto significa que más coeficientes serán cero resultando en una mayor compresión a expensas de menor calidad en las imágenes codificadas. Configurar el QP a un valor bajo significa que más de coeficientes serán diferentes de cero posterior a la cuantización, resultando en mejor calidad de imagen pero menor eficiencia de compresión.

2.2.3. Codificación del Bit Stream

La codificación de un archivo con formato H.264 consiste en una serie de símbolos codificados, estos símbolos cumplen con la sintaxis enunciada en el estándar e incluyen parámetros, identificadores, códigos delimitadores, tipos de predicción, vectores de movimiento y coeficientes de transformación. H.264 permite convertir toda esta información en patrones binarios haciendo uso de diferentes métodos:

Fixed lenght Code: Es un código de longitud fija. Cada símbolo es convertido aun patrón binario con un longitud especifica de n bits.

Exponential-Golomb Variable Length Code: El símbolo es representado con un palabra código *Exp-Golomb* con un número variable de bits. Las palabras código de menor longitud son asignadas a los símbolos con mayor probabilidad de ocurrencia. Los códigos *Exponential-Golomb* son de longitud variable con las siguientes propiedades:

1. La longitud del código se incrementa con el indice *code_num*.
2. Cada código puede ser construido lógicamente y decodificado algoritmicamente sin la necesidad de tablas de búsqueda.

Una palabra código *Exp-Golomb* posee la siguiente estructura:

$$\langle PrefijodeCeros\rangle\langle 1\rangle\langle INFORMACION\rangle \tag{2.4}$$

La palabra código consiste en un prefijo de M ceros, un bit 1 y un campo de información de M bits (*INFO*). Cada palabra código puede ser generada a partir del parámetro *code_num*:

$$\begin{aligned} M &= floor[\log_2(code\_num + 1)] \\ INFO &= code\_num + 1 - 2^M \end{aligned} \tag{2.5}$$

De forma contraria *code_num* se decodifica siguiendo los pasos:

1: Leer una serie de ceros consecutivos hasta que un bit 1 sea detectado. El número de ceros consecutivos se igual a M.

2: Leer un bit 1, simplemente ignorarlo.

3: Leer M bits después del bit 1 del paso anterior. Estos bits corresponden a los bits de información, *INFO*.

4: Obtenemos $code\_num = 2^M + INFO - 1$.

La tabla 2.3 es un ejemplo de palabras código generadas por este método. En este ejemplo x_i toma el valor de uno o cero. En cada palabra dada que contenga un sufijo de n bits existen 2^n códigos posibles para representar.

Bits	*Valores*
1	0
0 1 x_0	1 - 2
0 0 1 x_1x_0	3 - 6
0 0 0 1 $x_2x_1x_0$	7 - 14
0 0 0 0 1 $x_3x_2x_1x_0$	15 - 31
...	...

Tabla 2.3: Ejemplos de códigos Exp-Golomb

CAVLC Es un método eficiente para la codificación de coeficientes ordenados de la transformada de bloques residuales, donde distintos grupos de códigos de longitud variable (tablas VLC) son elegidos dependiendo de las estadísticas de los coeficientes recientemente codificados, se basa adaptación al contexto.

Después de un escaneo de los coeficientes de la transformada, la distribución espacial de los coeficientes típicamente muestra valores de mayor magnitud para la parte de baja frecuencia decreciendo hacia los coeficientes de alta frecuencia. Existen dos tipos de escaneo de coeficientes, en zig-zag o de campo. Después del escaneo se realiza un conversión a una serie de códigos de longitud variable (VLC). Las tablas VLC se seleccionan basado en las estadísticas locales como el número de coeficientes cuantizados distintos de cero y su posición.

Basándose en el comportamiento estadístico de los coeficientes de la transformada ya cuantizados, los siguientes elementos de información son usados para representar la información contenida en el bloque:

- Número de coeficientes distintos de cero (N) y coeficientes iguales a uno (T1), este ultimo valor refleja el número de coeficientes con valor absoluto de uno al final del escaneo. Para un bloque de 4×4, N puede tomar un rango de valores desde 0 hasta 16 y T1 puede tomar valores desde 0 hasta 3. Si existen más de tres valores de $+/-1$, solo los últimos 3 valores son tratados como casos especiales. El resto son codificados como el resto de los coeficientes.

  Para un bloque de luma existen cuatro posibles opciones de tablas de búsqueda, tres tablas de códigos de longitud variable y una tabla de código de longitud fija (FLC). La elección de la tabla depende del valor de N de los bloques superior (N_s)e izquierdo (N_i) del bloque actual, previamente codificados. Un parámetro N_a se calcula como sigue:

  I Si ambos bloques están disponibles en el mismo slice, $N_a = (N_s + N_i * 1) >> 1$, donde $>>$ indica un corrimiento binario hacia la derecha.

  II Si solo el bloque superior está disponible, $N_a = N_s$.

III Si solo el bloque izquierdo está disponible, $N_a = N_s$.

IV Si ninguno de los bloques está disponible, $N_a = 0$.

N_a selecciona la tabla de búsqueda de acuerdo a la tabla 2.4 tal que la elección se adapte al número de coeficientes codificados en bloques vecinos, representando la adaptación al contexto. Cada tabla VLC está diseñada para un número bajo, medio o alto de coeficientes distintos de cero. La tabla FLC asigna un código de seis bits a cada valor.

N_a	*Tablas*
0, 1	Tabla VLC 1
2, 3	Tabla VLC 2
4, 5, 6 y 7	Tabla VLC 3
8 y más	FLC

Tabla 2.4: Opciones de tablas de búsqueda VLC

- Codificar el signo de cada T1. Para cada $+/-1$ el signo es codificad con un bit único, 0 para positivo y 1 para negativo. Se realiza en orden inverso, iniciando de la parte de alta frecuencia.
- Codificar la magnitud de los coeficientes restantes distintos de cero. La codificación se realiza en orden inverso a partir de los coeficientes de alta frecuencia hacia el coeficiente de DC. Generalmente los componentes de altas frecuencia tienden a tener menores amplitudes.
- Codificar el número total de ceros antes del ultimo coeficiente. Se considera todos los valores cero anteriores a los coeficientes distintos de cero.
- Codificar cada ráfaga de ceros. El número de ceros precedentes a cada coeficiente distinto de cero es codificado en orden inverso

CABAC Un método de codificación aritmética en el cual el modelo de probabilidad es actualizado basado en estadísticas de codificación previas. Es un modo de codificación opcional disponible en los perfiles Main y High. La codificación de un símbolo involucra las siguientes etapas:

Binarización CABAC usa codificación aritmética binaria. El valor de un símbolo no binario como un vector de movimiento es convertido a un código binario antes de codificarlo aritmeticamente.

Seleccionar el modelo del contexto. Se selecciona un modelo de probabilidad para cada uno o más de los símbolos y se elige de un grupo de modelos disponibles de acuerdo a las estadísticas recientes.

Codificación Aritmética. Un codificador aritmético codifica los datos de acuerdo al modelo de probabilidad seleccionado.

Actualización del modelo de probabilidad. Se actualiza el modelo basados en el valor actual codificado.

2.3. Procesos del Decodificador

2.3.1. Decodificación del Bit Stream

Un decodificador de video recibe un flujo de datos en formato H.264 para posteriormente decodificar cada elemento de sintaxis y extraer la información como coeficientes de la transformada entera ya cuantizados, información de predicción entre otros. Esta información es usada para revertir el proceso de codificación y recrear la secuencia de video con imágenes.

2.3.2. Re-escalamiento y Transformación inversa

Re-escalar los coeficientes significa multiplicarlos por un valor entero y así restaurarlos a su escala original. Es importante notar que el proceso de cuantización no es un proceso totalmente reversible, la información removida durante la cuantización en el codificador no puede ser restaurada durante el re-escalamiento.

La transformación inversa combina los patrones bases estándar, ponderandolos por coeficientes re-escalados para recrear cada bloque de datos residuales. Las diferencias entre el bloque residual original y el reconstruido son debidas al proceso de cuantización. Un paso de cuantización de mayor tamaño tiende a producir grandes diferencias.

2.3.3. Reconstrucción

Para cada macrobloque el decodificador forma una predicción idéntica a la creada en el codificador usando predicción inter de imágenes previamente decodificadas o predicción intra de macrobloques previamente decodificados de la imagen actual. El proceso de predicción es un proceso totalmente reversible pues no existe pérdida de información. El decodificador suma la predicción al bloque residual para la reconstrucción de un macrobloque, el cual puede ser desplegado como parte de la secuencia de video.

2.4. Estructura de H.264

2.4.1. Jerarquía en el Video Codificado

La estructura básica de codificación de H.264/AVC es similar a los estándares previos. La codificación del video se desarrolla imagen a imagen, cada imagen

que es codificada, primero es particionada en un numero de *slices* (es posible tener un *slice* por imagen). Un *slice* consiste de una secuencia de un numero entero de MB codificados o pares de MB. Para maximizar la eficiencia de codificación, se usa solo un *slice* por imagen. Para un ambiente propenso al error, múltiples *slices* por imagen son usados, tal que el impacto de un error en el *bit stream* sea confinado solo a una pequeña porción de la imagen correspondiente. La jerarquía de la organización de los datos de video es:

$$Video \Rightarrow Imagenes \Rightarrow Slices \Rightarrow MB \Rightarrow Sub-MB \Rightarrow Bloques \Rightarrow Pixeles$$

En H.264/AVC, los *slices* son codificados individualmente y representan unidades de codificación, mientras las imágenes son unidades de acceso y consisten de *slices* codificados con datos asociados.

Tipos de Imágenes Básicas

Hay tres diferentes tipos de imágenes básicas en el estándar que pueden ser usadas por el codificador: I, P y B. Las imágenes I (Intra-codificadas) contienen solo *slices* intra codificados y consisten de macrobloques que no usan ninguna referencia temporal. Dado que solo la predicción espacial es permitida, estas imágenes son lugares convenientes en el *bit stream* para desempeñar acceso aleatorio, conmutación de canal, etc. Además detienen la propagación del error hecha en las imágenes decodificadas en el pasado. Adicionalmente se define un tipo especial de imagen intra codificada llamada Instantaneous Decoding Refresh. Esta imagen tiene la restricción que las imágenes que aparezcan después a ella en el *bit stream* no pueden usar como referencia las imágenes que aparezcan antes.

Las imágenes P (*Predicted Picture*) consisten de MB o sub-MB que pueden usar solo un vector de movimiento para el proceso de predicción. Varias partes de las imágenes P pueden usar diferentes imágenes como referencia para estimar el movimiento. Las imágenes de referencia pueden aparecer en el pasado o en el futuro de la imagen actual. Las imágenes P pueden además contener MB tipo intra.

Las imágenes B (*Bi-predicted Pictures*) consisten de MB o sub-MB que usan hasta dos vectores de movimiento en el proceso de predicción. Las imágenes B pueden contener MB con uno o ningún vector de movimiento. Estas imágenes pueden usar diferentes imágenes como referencia, ambas pueden estar en el pasado o futuro o una antes y otra en el futuro.

El termino *Group of Pictures (GOP)* como su nombre siguiere, consiste de un grupo de imágenes que inicia con un imagen intra-codificada. No hay ninguna definición formal de GOP en el estándar H.264/AVC. Por lo tanto, aunque el término GOP en relación con el estándar es ampliamente usado, se mantiene definido vagamente. Puede existir un número arbitrario de imágenes intercodificadas (P y B) en el GOP.

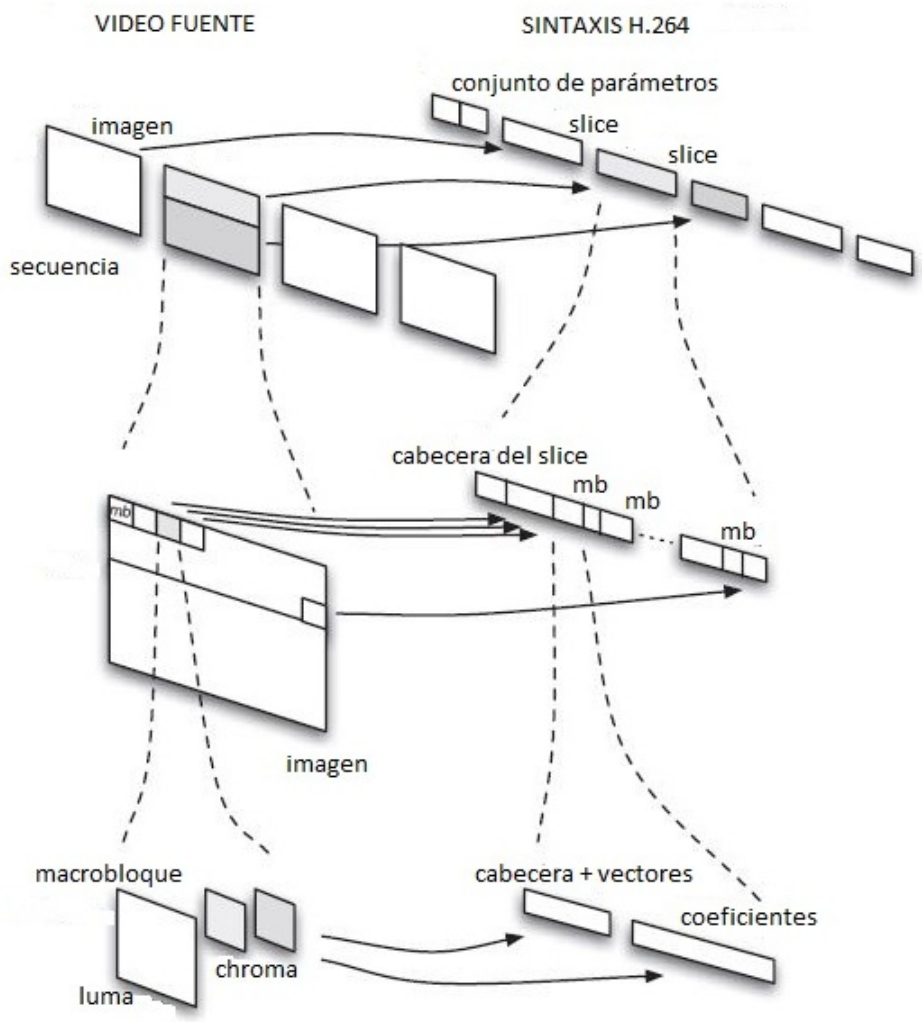


Figura 2.7: Resumen de la sintaxis en H.264/AVC

Cada imagen codificada esta compuesta de uno o más *slices*, cada uno conteniendo una cabecera seguida de un cierto numero de macrobloques que no necesariamente debe ser constante. Existen mínimas inter-dependencias entre cada *slice*, las cuales pueden ayudar a limitar la propagación del error. Los posibles escenarios para elegir su tamaño son:

- Un *slice* por imagen codificada, esta es una practica común en aplicaciones H.264.
- N *slices* por imagen, cada una conteniendo M macrobloques, N y M son enteros. El numero de bytes en cada *slice* tiende a variar dependiendo de

la cantidad de movimiento y detalle en el área de la imagen.

- N *slices* por imagen, conteniendo un número variable de macrobloques, tal que se mantenga el número de bytes por *slice* cuasi-constante. Esto es útil cuando cada *slice* es transmitido sobre una red de paquetes con tamaño fijo.

Los tipos de *slices* disponibles son listados en la tabla 2.6, conjuntamente con el tipo de MB que se permite incluir.

Tipo de *Slice*	Tipo de MB	Notas
I (incluye IDR)	Sólo I	Sólo predicción Intra (I)
P	I y/o P	Predicción Intra y/o predicción con una referencia por partición de MB (P)
B	I, P y/o B	Predicción Intra (I), predicción con una referencia(P) o dos referencias (B)
SP	P y/o I	Slices P-Switching
SI	I	Slices I-Switching

Tabla 2.5: Tipos de *Slices* en H.264/AVC

La cabecera de un *slice* comunica información común a todos los macrobloques contenidos en el, como el tipo de *slice*, que determina el tipo de MB están permitidos, el numero de imagen al que corresponde, configuración de la imagen de referencia y parámetro de cuantización entre otros.

Los datos de un *slice* consisten en una serie de macrobloques. Un macrobloque que no contiene datos es llamado *skip macroblock* y su ocurrencia es común en secuencias con bajo nivel de movimiento.

Un Slice Group (SG) es un subconjunto de MB de una imagen codificada y esta puede contener uno o mas *slices*. Dentro de cada *slice* en un SG, los MB son codificados en orden secuencial. Si se usa un sólo Slice Group por imagen, entonces todos los macrobloques son codificados secuencialmente a menos que se elija una configuración de ASO. Múltiples Slice Group hacen que sea posible trasladar los MB codificados de formas muy versátiles. La distribución de los macrobloques es determinada por medio de un mapa que indica a que SG pertenece cada MB. Uno de los usos mas importantes de multiples SG incluye esquemas de resistencia al error.

2.4.2. Perfiles

El estándar H.264 fue desarrollado para una amplia variedad de aplicaciones que se extienden desde video llamadas, televisión móvil, video conferencias y entretenimiento. Todo ese conjunto de aplicaciones tienen muy distintos requerimientos. Hacer que todos los decodificadores H.264 implementaran todas las

herramientas necesarias para la decodificación de tan distintas aplicaciones de video sería muy costoso y completamente innecesario. La forma en que se soluciono este dilema fue mediante la división de las herramientas de codificación en diferentes categorías, llamadas *Perfiles*. Cada perfil contiene un subconjunto de todas las herramientas de codificación especificadas en el estándar. Para un decodificador compatible con cierto perfil es mandatario implementar todas las herramientas especificadas en ese perfil. Para un codificador generando un *bitstream* H.264 para cierto perfil no es permitido usar ninguna herramienta o algoritmo que no este especificado en ese perfil pero si se permite que use sólo un subconjunto de las mismas.

Dentro de cada perfil hay un número de niveles, cada nivel especifica un conjunto de restricciones impuestas en los parámetros de configuración del video. De esta manera se simplifica un poco la complejidad de un decodificador si se conoce con anticipación la capacidades. Los tres principales perfiles y su área de aplicación principal aparecen en la tabla 2.6

Perfil	*Aplicación*
Baseline Profille	Video conferencias y Video Telefonía
Main Profile	Broadcast video
Extended Profile	Streaming Media

Tabla 2.6: Principales Perfiles y sus áreas de aplicación en H.264

Baseline Profile

Este perfil fue diseñado para aplicaciones relacionadas con dispositivos móviles, teléfonos celulares y video conferencias. En este tipo de aplicaciones el consumo de energía tiene un papel fundamental. Es por eso que la complejidad del codificador y decodificador se espera sea baja. Las imágenes tipo B y la codificación CABAC no son permitidas debido a su costo computacional. Se espera que la resolución espacial sea baja como los tamaños CIF y QVGA. Además de los bajos consumos de energía varias de las aplicaciones que usan este perfil operan en un ambiente con altas tasas de error. Buscando un transmisión de video más robusta algunas herramientas de resistencia al error han sido adicionadas como Ordenación Flexible de Macrobloques (Flexible Macroblock Order), Ordenamiento Arbitrario de *slices* (Arbitrary Slice Order) y *slices* Redundantes.

Es importante notar que este perfil no es un subconjunto de los perfiles *Main* y *High* ya que estos perfiles no soportan las herramientas de resistencia al error antes mencionadas. Por lo tanto este perfil no puede ser decodificado por decodificadores Main y *High* a menos que las herramientas de resistencia al error sean inhabilitadas.

Constrained Baseline Profile

Existen ambientes como *broadcast* de televisión, donde existen canales propensos al error y la necesidad de simplificar el codificador y decodificador pero también se desea compatibilidad con los perfiles *Main* y *High*. Con el propósito de lograr estos objetivos conjuntamente H.264 permite el uso de este perfil. En este perfil existe una opción que señaliza el posible uso de las herramientas de resistencia al error.

Extended Profile

Este perfil engloba las características del perfil *Baseline*. Básicamente está destinado para aplicaciones de video en *streaming*. Además de que incluye todas las herramientas de resistencia al error el perfil *Baseline* permite el uso de imágenes SP y SI, ya que si llegase a haber un repentino cambio en el ancho del banda del canal sea posible ajustar la tasa de transmisión del video. Además permite el uso de resoluciones SD y HD así como también el uso de imágenes B. No permite el uso de codificación CABAC.

Main Profile

Este perfil tiene a la televisión digital a una de sus mayores aplicaciones. Fue diseñado para proveer alta eficiencia de codificación. Incluye el uso de imágenes B y herramientas de codificación entrelazada. Soporta tanto codificación CABAC como CAVLC por lo que existe compatibilidad con el perfil *Constrained Baseline*. Para este perfil se esperan tasas de error bajas del orden de 10^{-6} las herramientas de resistencia al error fueron inhabilitadas.

High Profile

Este perfil provee la más alta eficiencia de codificación y la más alta flexibilidad con las herramientas de codificación. Engloba al perfil *Main*. Este perfil puede obtener hasta diez por ciento mayor eficiencia de codificación en comparación con el perfil *Main* cuando se codifica en la misma resolución. está destinado a aplicaciones de entretenimiento. Existen otros perfiles derivados a partir de este que habilitan características de codificación mejoradas, como mayor resolución de color, más bits por pixel y otros formatos de muestreo.

2.4.3. Niveles

sería lógico pensar que no es practico implementar un decodificador capaz de decodificar secuencias de video con resolución CIF con tan solo 176×144 píxeles hasta secuencias de video para cine con resolución de 4096. H.264 tiene

establecido niveles. Los niveles especifican algunos parámetros de codificación que tienen un impacto significativo en el tiempo de procesamiento para la decodificación y en las necesidades de memoria. El estándar H.264 especifica 17 niveles que muestran en la tabla A.3 del apéndice A. La compatibilidad de un decodificador es indicada a través del perfil y nivel.

Capítulo 3

Características de H.264/AVC enfocadas al Estudio de Pérdidas

Los requerimientos demandados al estándar H.264/AVC surgen de variadas aplicaciones de video. El objetivo es brindar soporte a aplicaciones como el *video streaming*, *video conferencing* sobre redes fijas e inalámbricas así como en diferentes protocolos de transporte. Las características del estándar apuntan en dirección a cumplir los requerimientos de tales aplicaciones.

Los canales inalámbricos tienen anchos de bandas cambiantes debido a factores como los desvanecimientos multitrayectoria, interferencia cocanal y ruido del canal. La capacidad del canal además varia con el movimiento del receptor móvil con respecto a la estación base en un ambiente celular. Todos esos factores generan altas tasas de Bit Error Rate (BER) en canales inalámbricos y pueden conducir a una devastadora degradación del video recibido, por lo que el problema de transmisiones eficientes de video sobre el ancho de banda variables y condiciones de canal adversas se torna un problema complejo. El video en bruto necesitaría un gran ancho de banda para su transmisión o almacenamiento, tanto que no sería costeable y casi imposible su transmisión en tiempo real. A través de explotar la redundancia espacial y temporal de las secuencias, el estándar H.264/AVC hace factible la transmisión y eficiente almacenamiento del video.

H.264 tienen un conjunto de nuevas características que lo hacen capaz de lograr alta compresión y excelente desempeño, algunas de las más importantes son:

1. *Capa de Abstracción de Red*: En H.264 los datos de video codificado son organizados en unidades NAL. Las unidades NAL contiene un formato genérico tanto para sistemas de transporte orientados a paquetes como

los orientados a *bit stream*. Cada unidad NAL contiene un byte de cabecera que indica el tipo de datos contenidos dentro del paquete. Los bytes restantes contienen carga de datos del tipo indicado en el byte de cabecera.

2. *Flexibilidad en la selección de tamaño de bloques*: H.264 permite mayor flexibilidad en términos de la selección tamaño de bloques para modelar eficazmente los detalles locales. El tamaño del bloque puede ser tan grande como de $16x16$ o tan pequeño como $4x4$.

3. *Múltiples cuadros de referencia*: A diferencia de los previos estándares que para el proceso de predicción solo usan un cuadro de video como referencia, H.264 permite el uso de hasta dieciséis cuadros de video para incrementar la eficiencia de codificación.

4. *Compensación de movimiento con precisión de un cuarto de pixel*: En compensación de movimiento, con precisión de hasta un cuarto de pixel es posible obtener muestras interpoladas de posiciones fraccionarias que se generan con vectores de movimientos fraccionarios. Basados en los vectores y muestras de posiciones enteras, las muestras fraccionarias son calculadas usando filtrado en dos dimensiones.

5. *Filtro de desbloqueo*: La codificación de video basada en bloque usualmente introduce distorsiones visuales en las secuencias de video. H.264 usa un filtro en lazo adaptable con el fin de remover la formación de bloques. El filtro además es incluido en el lazo de compensación de movimiento para mejorar la predicción temporal.

6. *Transformada Entera*: En H.264 la Transformada Discreta Coseno es reemplazada por la transformada entera para evitar problemas de desajuste generados con la transformación inversa coseno. La transformada entera es además eficiente para ser implementada en *hardware*.

7. *Ordenación flexible de macrobloques*: H.264 soporta siete diferentes modos inteligentes de agrupación de macrobloques en *slices*. Esto ayuda a dispersar los errores sobre un cuadro de video y prevenir su acumulación en una región en particular.

H.264/AVC se basa conceptualmente en una separación de dos capas: Video Coding Layer (VCL) y la Network Abstraction Layer (NAL). La capa VCL es el núcleo de codificación, su salida es una secuencia de bits representando los datos de video codificado. Esta capa se concentra en alcanzar la máxima eficiencia de codificación. La capa NAL abstrae los datos de la capa VCL en términos de los detalles requeridos por la capa de transporte y transporta esos datos sobre una variedad de redes. La capa NAL provee información de cabecera acerca del formato VCL usado. Una unidad tipo NAL (NALU) es un paquete que contiene un número entero de bytes, además, define un formato genérico para ser usado en las redes de transporte.

Los datos de salida de la capa VCL son encapsulados en unidades tipo NAL, previo a su transmisión o almacenamiento. Cada unidad NAL contiene una carga Raw Byte Sequence Payload (RBSP), una secuencia de datos de video codificado. La secuencia codificada se representa por un conjunto de unidades NAL que pueden ser almacenadas o transmitidas sobre redes basadas en paquetes o enlaces de transmisión orientados a *bit stream*. El propósito de especificar por separado la VCL y la NAL es distinguir entre las características especificas de codificación (dentro de la capa VCL) y las características especificas de transporte (en la capa NAL).

3.1. Capa de Abstracción de Red NAL

La Network Abstraction Layer (NAL) está diseñada para proveer adaptación de red y habilitar el uso simple y efectivo de la VCL para una amplia variedad de sistemas. La NAL facilita la habilidad para acoplar los datos VLC a capas de transporte, por ejemplo RTP/IP, Servicios de Internet, H.32X y sistemas MPEG-2 para servicios de difusión.

Como se observa en la Figura 3.1 [4] los datos de video se organizan en unidades NAL, cada una de las cuales es efectivamente un paquete que contiene un número entero de bytes. El primer byte de cada unidad NAL es un byte de cabecera que indica el tipo de dato dentro de la unidad NAL y los bytes restantes contienen carga de datos de mismo tipo del indicado en la cabecera. La carga de datos en la unidad NAL es intercalada con bytes de prevención se secuencia, son bytes insertados con un valor especifico para prevenir la generación aleatoria de un patrón llamado *prefijo de código de inicio* dentro del contenido de la unidad NAL. La definición de la estructura de la unidad NAL especifica un formato genérico para el uso en sistemas de transporte orientados a paquetes y orientados a *bitstream*.

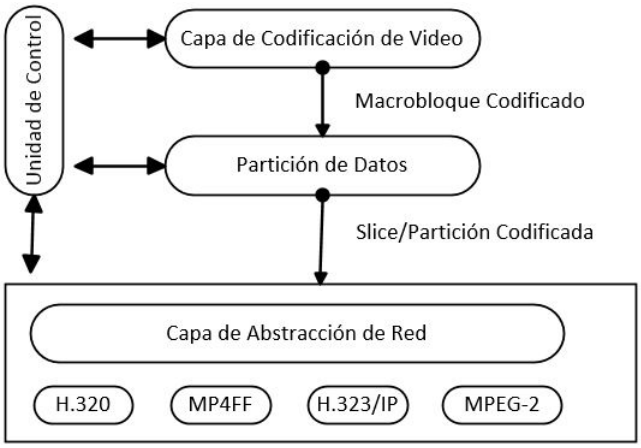


Figura 3.1: Estructura del codificador de video H.264/AVC

3.1.1. Uso del Formato Byte Stream en unidades NAL

Algunos sistemas (e.g. H.320 y MPEG-2) requieren entrega de unidades NAL enteras o parciales como un flujo ordenado de bytes o bits dentro de las cuales los limites de las unidades NAL necesitan ser identificables con patrones dentro de los datos codificados por si mismos. Para el uso en tales sistemas, la especificación H.264/AVC define un formato orientado a *bit stream*, donde a cada unidad NAL se le antepone un patrón de tres bytes como prefijo llamado prefijo de inicio de código. Los limites de la unidad NAL pueden ser identificables buscando en los datos codificados el patrón de código de inicio. Se puede hacer uso de uso de bytes de prevención de emulación para garantizar que el código de inicio sea una cadena única que no se repita en el flujo de bits.

Una pequeña cantidad de datos adicionales, solo un byte por cuadro de video, es agregada para permitir a el decodificador operar en sistemas que proveen un flujo de bits sin alineación de limites de bytes para recuperar la alineación necesaria de los datos en el flujo continuo de información. Opcionalmente datos adicionales pueden ser insertados en el flujo de datos para permitir extender la cantidad de datos a enviar, con el fin de lograr un recuperación más rápida de la alineación de trama.

3.1.2. Unidades NAL para sistemas orientados a transportación de paquete

En otros sistemas (i. e. sistemas con pila de protocolos de Internet y RTP), los datos son transportados mediante paquetes que son fragmentados por el protocolo de transporte del sistema. Identificación de los limites de las unidades NAL dentro de los paquetes puede realizarse sin el uso de patrones de códigos de inicio. Para este tipo de sistemas la inclusión de códigos de inicio en los datos sería un desperdicio de capacidad.

3.1.3. Unidades NAL con contenido VCL y no-VCL

Las unidades NAL se clasifican a su vez en unidades VCL y unidades no-VCL. Las unidades VCL contienen los datos de la representación de valores de las muestras de las imágenes que componen el video; las unidades no-VCL contienen cualquier información adicional asociada como conjuntos de datos de configuración de parámetros (i.e. importantes cabeceras de datos que pueden aplicarse a muchas unidades NAL de tipo VCL) e información de mejora suplementaria (i.e. información de temporización y otros datos complementarios que mejoran el uso del decodificador de la señal video pero no necesariamente para decodificar las muestras).

3.1.4. Conjunto de parámetros

Un conjunto de parámetros contiene información que se espera que no cambie constantemente y afecta la decodificación de un gran número de unidades VCL. Existen dos tipos de conjuntos de parámetros:

1. *Conjunto de Parámetros de Secuencia (SPS)*: Estos parámetros no cambian frecuentemente y su efecto cubre una serie consecutiva de cuadros de video codificado.

2. *Conjunto de Parámetros de Imagen(PPS)*: Estos parámetros ayudan a decodificar una o más imágenes individuales dentro de una secuencia de video.

Los mecanismos llamados SPS y PPS desacoplan la transmisión de información que no cambia frecuentemente de la información asociada con la representación de muestras de video codificado. Cada unidad Network Abstraction Layer con contenido Video Coding Layer contiene un identificador que hace referencia al contenido del conjunto de parámetros de imagen relevantes; cada conjunto de parámetros de imagen contiene un identificador que hace referencia al contenido de un conjunto de parámetros de secuencia. De esta forma, una pequeña cantidad de datos (el identificador) puede ser usado para referirse a una gran cantidad de información (el conjunto de parámetros) sin la necesidad de repetir la información dentro de cada unidad NAL de tipo VLC.

Los parámetros de secuencia e imagen pueden ser enviados con anticipación a las unidades NAL tipo VLC a las que serán aplicados y pueden ser repetidos para proporcionar robustez contra la pérdida de datos. En algunas aplicaciones, los parámetros de secuencia pueden ser enviados dentro del canal que transporta las unidades NAL tipo VLC (transmisión en banda). Para otras aplicaciones, se puede tomar ventaja de enviar el conjunto de parámetros fuera de banda usando mecanismos de transporte más confiables que el canal por donde se transmite el video.

3.1.5. Unidades de Acceso

Un conjunto de unidades NAL representan una unidad de acceso. La decodificación de cada unidad de acceso resulta en una imagen decodificada. Cada unidad de acceso contiene un conjunto de unidades VCL que conjuntamente componen una *imagen decodificada primaria*. Es posible prefijar con un delimitador de unidad de acceso para ayudar en la localización del inicio de la unidad de acceso. Información de mejora suplementaria conteniendo datos como temporización de imágenes puede preceder la imagen primaria codificada.

La imagen primaria codificada consiste de un conjunto de unidades NAL tipo VCL compuesta de *slices* o particiones de datos que representan las muestras muestras de la imagen de video. Seguido de la imagen primaria codificada

pueden transmitirse unidades NAL tipo VCL que contienen representaciones de áreas de la misma imagen de video. Estas son nombradas como *imágenes codificadas redundantes* y están disponibles en el decodificador para recuperación de pérdidas o datos corruptos en las imágenes primarias codificadas. Los decodificadores no necesitan decodificar las imágenes redundantes su las primarias están presentes.

Finalmente, si una imagen codificada es la ultima imagen de una secuencia de video codificada [1], una unidad NAL de fin de secuencia debe estar presente para señalizar el final de secuencia; si la imagen codificada es la ultima imagen de un flujo de unidades NAL, una unidad NAL de *end of stream* debe aparecer.

3.1.6. Secuencias de video Codificado.

Una secuencia de video codificado consiste de una serie de unidades de acceso que están ordenadas secuencialmente en el flujo de unidades NAL y usa solo un conjunto de parámetros de secuencia. Cada secuencia de video codificado puede ser decodificado independientemente de cualquier otra secuencia de video, dada la información del conjunto de parámetros necesarios. Al inicio de una secuencia de video codificado se puede encontrar una unidad de acceso Instantaneous Decoding Refresh (IDR). Una unidad IDR es una imagen intra-codificada que puede ser decodificada sin decodificar ninguna imagen previa. La presencia de una unidad de acceso IDR indica que ninguna imagen subsecuente en la secuencia requerirá de imágenes previas a la imagen intra para ser decodificada.

3.2. Ordenación Flexible de Macrobloques

La herramienta de FMO (Flexible Macroblock Ordering) permite el ordenamiento de macrobloques de una determinada imagen en dos o más grupos SG (*slice group*). Cada imagen puede ser dividida hasta en ocho diferentes SG, cada uno conteniendo al menos un macrobloque. De forma similar a anteriores estándares de codificación basados en codificación por bloques, H.264 codifica los macrobloques de izquierda a derecha y de arriba hacia abajo (*raster scan*). La asignación de macrobloques a diferentes SG hace que el FMO sea una herramienta poderosa para recuperación de errores. H.264/AVC define seis modos diferentes de FMO que se pueden apreciar en la Figura 3.2. Se ha usado el modo disperso dado su desempeño superior [5]. Dividiendo la imagen en SG ayuda a dar prioridad a partes de una imagen y a la ocultación del error [6].

Si algún macrobloque particular es reportado como perdido, los macrobloques vecinos pueden ser usados para reconstruirlo. Cada SG consiste de una

[1] Una secuencia de imágenes que es independientemente decodificable y usa solo un conjunto de parámetros de secuencia

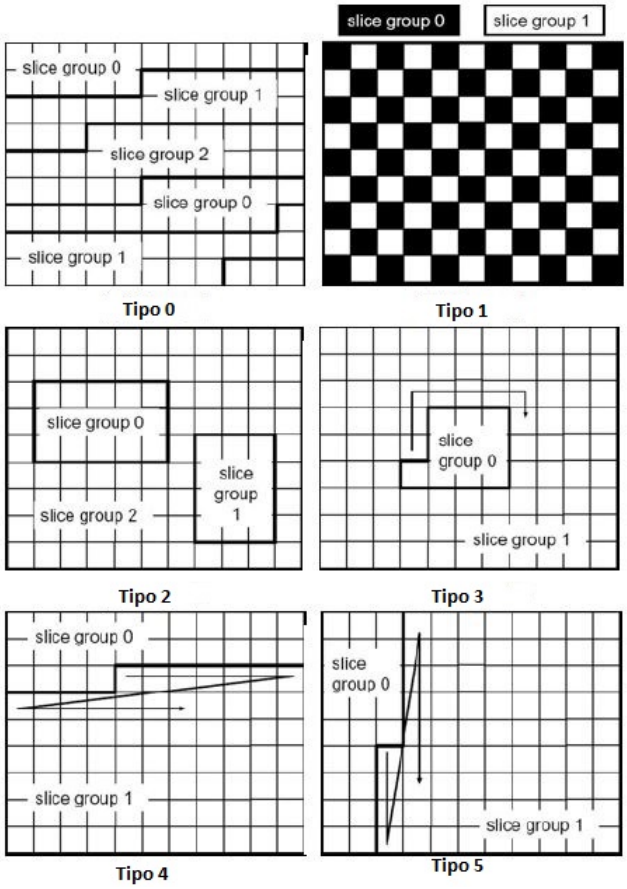


Figura 3.2: Tipos de FMO de H.264

secuencia de macrobloques. En el estándar H.264 cada slice es decodificable de forma independiente y por lo tanto un error en un slice no se propaga hace slices vecinos. Slices más pequeños proveen mejor desempeño al error porque tienen menor probabilidad de ser contaminado con ruido, sin embargo, el uso de tamaño de *slices* pequeños reduce la eficiencia de compresión [5].

Si el patrón FMO de la Figura 3.3 es usado en la codificación de video y un SG resulta que perdido por algún motivo durante su transmisión, en el cuadro de video actual los macrobloques corruptos dispondrán de información de macrobloques adyacentes o vecinos a la región donde se genera la pérdida para que se implementen métodos de cancelamiento de error. La mayoría de los macrobloques de la imagen actual poseerán cuatro macrobloques vecinos, con excepción de los macrobloques de los bordes. La independencia entre diferentes Slice Group (SG) dentro de la misma imagen asegura que los bloques vecinos

de otro(s) SG(s) sean decodificados sin inconvenientes. La especificación provee seis diferentes patrones para agrupar los macrobloques, algunos de esos patrones pueden ser usados para identificar fácilmente regiones rectangulares de interés dentro de una imagen (i.e. el rostro de una persona). Aunque es posible argumentar que esta región de interés debe ser mejor protegida que otras partes del video, esto no siempre se mantiene. Usualmente la redundancia temporal puede ser explotada por los algoritmos de cancelamiento de error para reconstruir regiones pérdidas, incluso si estas regiones están localizadas dentro de las regiones de interés.

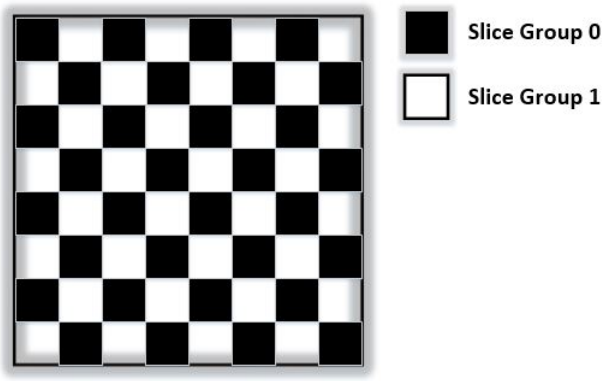


Figura 3.3: FMO en modo disperso del estándar H.264

Los parámetros que controlan la configuración FMO son incluidos en PPS. El tipo de FMO deseado puede ser seleccionado en el archivo de configuración del codificador usando el parámetro (*slice_group_map_type*).

H.264/AVC especifica siete tipos diferentes de mapeo FMO:

1. *FMO tipo 0 - Interleaved slices*: El número de macrobloques consecutivos en cada SG puede ser definido antes de que el siguiente SG inicie (longitud continua). Por lo cual, para configurar este tipo de mapeo FMO, la longitud continua y el número de SG's (*num_slice_groups_minus*1) son parámetros requeridos.

2. *FMO tipo 1 - Scattered slices*: Los macrobloques son asignados a un SG por medio de la ecuación 3.1

$$i \rightarrow ((i \; mod \; w) + ((i/w) * n)/2) mod \; n \tag{3.1}$$

Donde:

i: es el número de macrobloque a ser mapeado.

w: es el ancho de la imagen en término de macrobloques.

n: es el número de SG deseados (*num_slice_groups_minus1 + 1*).

Dado que la formula es conocida en codificador como en el decodificador, el único parámetro requerido para ser codificado dentro de los PPS es el número de SG. Con esta función de mapeo es posible definir el patrón de ajedrez.

3. *FMO tipo 2 - Foreground with left-over*: Cada SG es definido por las coordenadas de las esquinas superior-derecha e inferior-izquierda de un rectángulo contenido en la imagen. Los rectángulos pueden traslaparse, en este caso, los macrobloques dentro de las regiones de traslape son asociados con el SG con el menor número de identificador. Los parámetros necesitados en el decodificador son solo el número de SG y un par de coordinadas para cada rectángulo. Esta topología es particularmente útil en escenarios donde una región de interés pueda ser identificada, tal como video conferencias donde la región de interés es la rostro de los participantes. Como se mencionó anteriormente, es necesario enviar un conjunto de parámetros de imagen (PPS) para cambiar los parámetros que definen las regiones del primer plano.

4. *FMO Tipo 3 - Box out*: El mapeo de macrobloques a SG inicia desde el centro de la imagen y avanza de forma rotacional. La dirección puede ser en el sentido de las manecillas del reloj o de inversa, dependiendo del parámetro *SliceGroupChangeDirectionFlag*. El tamaño de cada SG puede cambiar en cada imagen dependiendo de valor de los parámetros *SliceGroupChangeRate* y *SliceGroupChangeCycle*. El primer parámetro especifica el múltiplo, en número de macrobloques por el cual el tamaño de SG puede cambiar de una imagen a la siguiente; solo se define un vez y no cambia. El valor de *SliceGroupChangeCycle* puede cambiar cada imagen. Por lo tanto , el número de macrobloques en el SG cero es $SliceGroupChangeCycle \times SliceGroupChangeRate$.

5. *FMO tipo 4 - Raster scan*: El mapeo de macrobloques a SG inicia del inicio de la imagen y avanza de izquierda a derecha y arriba hacia abajo. Como en el caso previo, el tamaño de los SG puede cambiar continuamente.

6. *FMO tipo 5 - Wipe scan*:El mapeo de macrobloques a SG comienza desde el inicio de la imagen y avanza verticalmente columna por columna. Como los FMO tipo 3 y 4, el tamaño de los SG puede cambiar cada imagen.

7. *FMO tipo 6 - EXplicit*: La función de mapeo es generada con una asignación explicita de macrobloques a los SG. En este caso, el mapa entero de macrobloques tiene que ser transmitido dentro de PPS.

La Figura 3.2 ilustra los diferentes tipos de FMO expuestos.

La flexibilidad de la herramienta FMO viene con cierto costo. Una primera observación es que el costo de usar FMO es mayor si hay pocos *slices* en una imagen (i.e. muchos macrobloques por slice). Otra observación es el hecho de que los parámetros de cuantización (QP) y por ende el tasa de transmisión tienen un claro impacto en el costo relativo del uso de FMO. Para ser más precisos, el costo relativo del FMO es mayor a bajas tasas de transmisión.

La herramienta de FMO introduce elementos de sintaxis adicionales. El costo del FMO depende de muchos parámetros. Codificando secuencias de video y variando los parámetros (i.e. con y sin FMO) es posible tener un noción respecto al costo atribuido al FMO. Esta información es útil cuando se requiere hacer un compromiso entre ahorrar bits adicionales en el video por FMO y mejorar la resistencia al error.

De acuerdo a [6], cuando se trata el costo de introducir FMO, tres aspectos fundamentales deben ser tomados en cuenta. Primeramente, FMO requiere que el codificador provea un mapa de macrobloques (*MBA_map*) al decodificador. Segundo, FMO estropea la predicción basada en macrobloques vecinos. Por ultimo, FMO fuerza a dividir la imagen en múltiples *slices*. En caso del FMO tipo 1, la transmisión del mapa de macrobloques puede ser un tanto innecesario; lo único que necesita señalizar el codificador es el número de SG's que son usados y que el FMO tipo 1 ha sido implementado durante la codificación del video. Esos parámetros son comunicados a por medio del conjunto de parámetros de secuencia (PPS). Dado que el FMO tipo 1 provee un patrón de distribución disperso para la mejor protección contra el error, es improbable que el patrón cambie durante la secuencia de video. Por lo tanto, FMO tipo 1 no requiere la transmisión de información PPS extra.

Dado que cada flujo de video codificado necesita al menos un PPS, es posible concluir que el costo de establecer el uso de FMO tipo 1 es insignificante. H.264/AVC es efectivo reduciendo las redundancias espaciales entre bloques vecinos gracias a muchas nuevas mejoras. Cuando se usa FMO tipo 1 ningún par de macrobloques vecinos pertenecen al mismo SG. Por consiguiente, es imposible par un macrobloque ser predicho por medio de sus vecinos. Esto no solo afecta la eficiencia de compresión de las imágenes intra si no también las imágenes inter codificadas.

Considerando los macrobloques de tipo P_SKIP, para lograr una buena compresión de datos, estos tipos de macrobloques son codificados usando un valor seguido por el número de macrobloques que tienen ese mismo tipo. Debido al uso del patrón disperso el número de macrobloques con el mismo tipo dentro de un SG casi siempre sera uno. En H.264/AVC los *slices* consisten de una cabecera y de los datos de los diferentes macrobloques dentro del slice. Para *slices* de la misma imagen, las cabeceras tienen muchos valores en común. Por lo que hay cierta cantidad de redundancia si una imagen es codificada usando más de un slice. En H.264/AVC cada SG necesita de al menos un slice, por lo que entre más SG's sean usados mayor sera la redundancia enviada al decodificador. En

conclusión se puede afirmar que el costo de introducir FMO tipo es doble, primeramente se necesitan más *slices* en comparación cuando no se usa FMO y segundo se reduce la posibilidad de explotar la redundancia espacial dentro de una imagen.

3.3. Detección del Error en H.264/AVC

Con la finalidad de reducir la tasa de transmisión o el tamaño de almacenamiento del video, el estándar H.264/AVC usa codificación entrópica. En este tipo de codificación, un error en la transmisión de una palabra código no solo afecta la misma, sino también las palabras código subsecuentes. Obviamente, este hecho resulta en degradación de la calidad del video. La detección del error ocupa un lugar fundamental en la robustez del decodificador para evitar su propagación.

El método más comúnmente usado para la detección del error se basa en la ubicar anomalías en la sintaxis de la representación del video. Como ya se ha mencionado antes, el estándar define explicitamente todos los elementos que componen la sintaxis y su orden adecuado en el *bitstream*. Básicamente si ocurre un error en *bitstream* de video se incurre en violaciones de sintaxis o de su interpretación. Debido al uso de codificación de longitud variable, los errores frecuentemente se propagan en el video. La detección del error bajo esta estrategia puede identificar situaciones anormales como:

- Elementos de sintaxis no definidos.
- más de 16 coeficientes codificados en un bloque de 4×4.
- Cabeceras de sincronización no validas.
- Un número incorrecto de bits de relleno. Puede ocurrir cuando quedan bits restantes después de decodificar todos los coeficientes del ultimo bloque en un paquete de video.
- Imposibilidad de decodificar paquetes de video.

Posterior a la detección del error se reporta la localización del macrobloque donde ocurrió la violación y los siguientes macrobloques son considerados como corruptos. Adicionalmente se realiza la búsqueda de un código de inicio (palabra de código única) para re-sincronizar el *bitstream*.

Cuando se detecta presencia de error en el *bitstream* H.264, al menos un bit en el slice debe estar erróneo. No obstante, no todos los casos de bits en error causaran problemas cuando se lleva a cabo la decodificación. De acuerdo a [13] la probabilidad de detectar un error acertadamente es bastante alta. En general, menos del 1 % de los errores posibles en una secuencia de video no son detectados. Por lo que la probabilidad de detectar el error correctamente mayor

que el 0.99. Los errores que ocurren y no son detectados son casi imperceptibles cuando se despliega el video.

El algoritmo de cancelamiento implementado en H.264 asume que en un ambiente inalámbrico, los *slices* erróneos son descartados antes de decodificarlos. Todos los *slices* que forman parte de un una imagen y sean correctamente recibidos son decodificados primero y después los *slices* perdidos son cancelados o reconstruidos. En el decodificador se mantiene un registro del estatus de cada macrobloque, los estatus pueden ser: Correctamente Recibido, Perdido y Cancelado.

3.4. Cancelación de Error en el Decodificador

Dado que las transmisiones de video están limitadas por el retardo impuesto por el canal, no es posible retransmitir los paquetes erróneos o perdidos. Por lo tanto existe la necesidad de implementar métodos para tratar de reducir el impacto visual de los errores después de su localización en el *bit stream*. Las *técnicas de cancelamiento* de error son implementadas en el decodificador para restaurar la parte del video corrompida con base en la información correctamente recibida. El cancelamiento de error basa su fortaleza en la correlación espacial y temporal presente en los bloques de video, por lo que se clasifican como métodos de cancelamiento espaciales o temporales. Cabe aclarar que el cancelamiento de error es una sección del estándar exclusivamente informativa.

3.4.1. Cancelamiento Espacial

El cancelamiento espacial aprovecha la característica de suavidad de las señales de video. Esta se refiere a que generalmente se presentan transiciones lentas o suaves entre bloques adyacentes de una imagen, por lo tanto es muy probablemente que los coeficientes de bloques vecinos posean valores cercanos en el dominio espacial. Los métodos de cancelamiento espacial usan la información espacial de de los bloques circundantes para interpolar el área pérdida.

El método de cancelamiento de error espacial en H.264 se basa en la interpolación ponderada de píxeles. Se estima cada píxel del macrobloque corrupto con base en los cuatro macrobloques adyacentes pero solo se usan los píxeles de los bordes como se muestra en la figura 3.4. Los valores de los cuatros píxeles vecinos son ponderados de acuerdo al inverso de la distancia $d_{(i,j)\rightarrow(x,y)}$ entre cada píxel fuente y el píxel destino, para luego ser dividido por la suma de las distancias. Los valores de los píxeles $Y(x, y)$ para un macrobloque se obtienen como se muestra en la ecuación 3.2

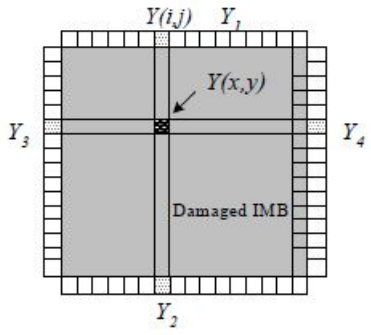


Figura 3.4: Cancelamiento Espacial en H.264

$$\widetilde{Y}(x,y) = \frac{\sum_{(i,j)\epsilon N} Y(i,j) \times [15 - d_{(i,j)\to(x,y)}]}{\sum_{(i,j)\epsilon N} d_{(i,j)\to(x,y)}} \tag{3.2}$$

Donde:
$N = \{Y_1, Y_2, Y_3, Y_4\}$, (x, y) ϵ MB intra pérdido

Solo los macrobloques correctamente recibidos son usados para el calculo, si al menos dos de tales macrobloques están disponibles. De otra forma, el cancelamiento también ocupa macrobloques vecinos previamente cancelados. Esta técnica de cancelamiento tiene un buen desempeño cuando se aplica a zonas planas, donde predominan las bajas frecuencias espaciales. Sin embargo no toma en cuenta la dirección de bordes de objetos. Existen otras técnicas espaciales de cancelación más eficientes a esta, que hacen uso de mayor cantidad de procesamiento e información disponible para restaurar eficazmente los bordes de objetos. Estas técnicas regularmente incrementan el gasto computacional en el decodificador por lo que es importante mantener un equilibrio entre eficacia de restauración y complejidad.

Si el macrobloque perdido pertenece a la ultima fila de la imagen, el cancelamiento se realiza a partir de los macrobloques vecinos localizados en la parte superior y lateral. Si el macrobloque perdido es la primera fila de la imagen, el cancelamiento se realiza a partir de los bloques de la parte inferior y lateral. Esto se cumple de forma análoga para los macrobloques de los bordes izquierdo y derecho. Si el macrobloque pertenece a otra posición distinta a las mencionadas la interpolación es posible con todos los bloques vecinos de estar disponibles.

3.4.2. Cancelamiento Temporal

Para el cancelamiento temporal propuesto en el estándar H.264 inicialmente se investiga la actividad de movimiento de los *slices* correctamente recibidos para

la imagen actual. Si el promedio de los vectores de movimiento es menor que un umbral predefinido, todos los *slices* perdidos son reconstruidos a través de la sustitución del área espacialmente correspondiente de la imagen de referencia. Cada píxel de una imagen reconstruida es una copia del píxel correspondiente de la imagen de referencia previamente decodificada. No se hace uso de la técnica de compensación de movimiento. La imagen reconstruida es usada tanto en la reproducción del video así como referencia, por lo que también se coloca en el *buffer* de imágenes de referencia.

Si el promedio de los vectores de movimiento es mayor al umbral se emplea cancelacion de error compensada en movimiento. Los vectores de movimiento de los macrobloques perdidos son estimados de la información de movimiento de los cuatro vecinos espaciales, generándose distintas estimaciones posibles. Cada vector de movimiento estimado es usado para ubicar un macrobloque en la imagen de referencia. Los valores de los píxeles en la imagen de referencia son copiados al área pérdida de la imagen actual.

Durante este proceso de prueba, los valores de los píxeles reconstruidos son formados usando cada vector de movimiento estimado. La selección de que vector de movimiento se debe elegir se hace con base en la suavidad del área reconstruida. La suavidad es medida por medio distorsión de luminancia d_{luma} calculada de acuerdo a la ecuación 3.3, que representa el promedio de las diferencias de los valores de luminancia entre el bloque predicho y los vecinos del bloque perdido. Solo se consideran los valores de los bordes de los bloques como se muestra en la figura 3.5.

$$d_{luma} = \frac{1}{N} \sum_{i=1}^{N} |\widetilde{Y}(\overrightarrow{mv})_i^{IN} - Y_i^{OUT}| \tag{3.3}$$

Donde:

$\widetilde{Y}(\overrightarrow{mv})_i^{IN}$: Es el i-esimo valor de luminancia reconstruido del bloque predicho en función a un vector de movimiento.

$\overrightarrow{mv}$: Representa el vector de movimiento estimado.

Y_i^{OUT}: Es el i-esimo valor de luminancia de los bloques vecinos.

N: Es el total de píxeles en los limites de los bloques.

La imagen o imágenes de referencia pueden ser cualquiera disponible en el *buffer* del decodificador que cumpla con la condición de incluir información de movimiento. Incluso una imagen tipo I puede ser reconstruida por este método, siempre y cuando no sea la primera imagen de una secuencia de video.

Experimentos prueban que la reconstrucción con estimación movimiento produce mejores resultados que la reconstrucción sin estimación ya que considera el flujo de movimiento del video [11]. Cuando el cuadro entero se pierde, la copia del cuadro es llamada por default.

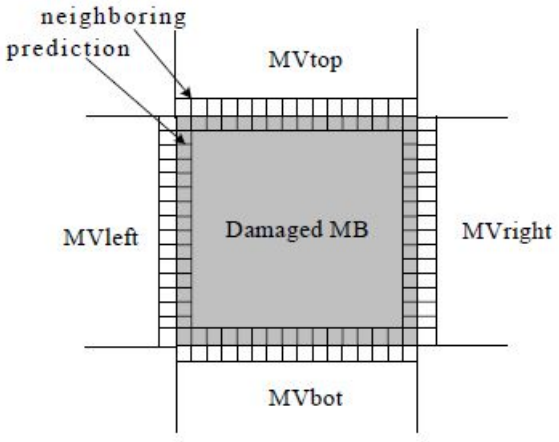


Figura 3.5: Cancelamiento Temporal en H.264

La implementación del esquema de cancelamiento de error en el estándar H.264 se realizó en el software JM[2] version 14.1 [3].

3.4.3. Pérdida total de la imagen

Cuando una imagen I o P se pierde o corrompe, el decodificador la identifica como pérdida a partir del conteo de orden de imagen (POC[3]) En caso de perder una imagen de referencia, el cancelamiento se lleva a cabo a través de la ultima imagen de referencia decodificada usando copia de cuadro. La propagación del error puede ser evitada si el cuadro de referencia reconstruido no es usado para decodificar otras imágenes; de otra manera el error se propagara hasta el siguiente cuadro IDR. En el caso de que no se use el cancelamiento de error, el cuadro es completamente perdido y por lo tanto hay perdida de sincronización en el video.

3.4.4. Pérdida parcial de la imagen

Si se pierde una partición de un slice, cualquiera que sea, el decodificador marca los macrobloques del slice y todos los demás bloques dependientes como perdidos y el cancelamiento de error es implementado de acuerdo a la configuración del decodificador.

[2]Implementación oficial del estándar H.264/AVC, existen diferentes versiones. A la fecha realización de este documento la ultima versión es la 18.5

[3]POC determina el orden de reproducción de las imágenes decodificadas, es derivado de la cabera del slice.

3.5. Esquemas de Resistencia al Error en H.264

H.264 hace disponible varios mecanismos de resistencia al error [1] tanto en el lado del codificador como del lado de el decodificador. En el codificador encontramos una gran cantidad de parámetros que se pueden ser sintonizados. Existe un compromiso entre la tasa de compresión y la redundancia agregada para los mecanismos de resistencia al error [6] que puede ser focalizado hacia diferentes problemas específicos presentes en ambientes heterogéneos. Los métodos más comunes de detener la propagación temporal de errores cuando no se dispone de canal de retroalimentación son la inserción aleatoria de macrobloques de actualización tipo intra y la inserción de imágenes con codificación tipo intra (IDR).

Mientras el uso de imágenes IDR resetea el proceso de predicción evitando la propagación del error, su uso viene con un alto costo en términos de ancho de banda que provoca severas variaciones en la tasa de transmisión. El uso de macrobloques tipo intra como reset de la propagación del error es más efectivo que el uso de imágenes intra porque no solo ayuda a lograr una tasa de transmisión constante sino que también provee mejores resultados reseteando estadisticamente el error para cada uno de los macrobloque. La actualizacion de una linea completa de macrobloques es otra opcion donde cada un grupo de macrobloques son intra codificados N imagenes. Se le considera solo una variante más.

El uso de *slices* es otro método comúnmente usado detener la propagación del error mejorar la robustez. Los macrobloques pertenecientes a un slice pueden ser decodificados independientemente de otros *slices* basado en que las dependencias inter-slice no son permitidas. Slices tipo SP hacen uso de la codificación predictiva compensada en movimiento para explotar la redundancia temporal en la secuencia, como lo hacen los tipo P. La codificación *slices* SP [6] permite reconstrucción idéntica del slice incluso cuando diferentes imágenes de referencia han sido usadas. Ellos esencialmente ayudan al flujo de datos conmutando, empalmando el acceso aleatorio y las herramientas de resistencia de error.

Otra herramienta en el codificador H.264 es la optimización de tasa de distorsión (Rate-Distortion Optimization (RDO)). La distorsión puede surgir debidos a errores de cuantización o predicción de bloques reconstruidos. Si la predicción no provee buena compresión, se permite la codificación intra para macrobloques. Respecto a la sintonización del codificador, RDO puede ser desactivado o activado si la optimización es deseada. Sin embargo, tales valores solo serán óptimos en la ausencia de errores en la red. Por tal razón, un tercer modo está disponible donde el codificador toma en cuenta la tasa de pérdidas de paquetes esperada de la red así como también los métodos del decodificador para hacer frente a los errores con el fin de decidir codificar un macrobloque en tipo intra o inter.

La opción de restricción de predicción intra es ligada al modo de predicción intra. Cuando está activa, el codificador evita usar píxeles de macrobloques codificados en tipo inter, para predecir macrobloques intra.

Adicionalmente el decodificador juega un importante rol en la resistencia al error dado que es responsable del cancelamiento de error[8]. Mantiene un mapa

del estatus de macrobloques, el cual indica para cada imagen siendo decodificada si cierto macrobloque ha sido correctamente recibido, perdido o si fue reconstruido. Los métodos varían para imágenes tipo intra de los de tipo inter. Para imágenes intra, la principal tarea consiste en desempeñar el promediado ponderado de píxeles de cada macrobloque perdido para reconstruir el área perdida. Para imágenes inter la tarea consiste principalmente en predecir adecuadamente los vectores de movimiento para los macrobloques perdidos, aunque los métodos intra también pueden ser utilizados. Para una más completa descripción de los métodos revisar [9]. El decodificador además cumple con otras tareas como manejar múltiples imágenes de referencia o pérdidas de imágenes completas. Como se menciona en [10], el decodificador de referencia H.264 no incorpora ninguna característica de recuperación de errores porque incrementa significativamente la complejidad del decodificador, con solo pequeñas modificaciones. Por lo que la detección de errores y su manejo deben ser procesados externamente.

3.5.1. Partición de Datos, Predicción Limitada Inter e Intra

La especificación H.264/AVC hace una clara distinción entre la parte de codificación de video y la interfaz de adaptación de red. Como se ve en la Figura 3.1 se expresa la presencia de dos capas en el estándar: VCL y NAL. Estas dos capas están interconectadas por medio de unidades NAL (NALU), que son estructuras sintácticas que consisten de un byte de cabecera y la carga de datos en una unidad NALU. Cuando se usa partición de datos [16] los datos codificados de un slice es dividido hasta en tres parte de acuerdo a su importancia. Cada parte es llamada *partición de datos* (*data partition*) del slice codificado y es puesta como carga de una NALU por separado. Las siguientes particiones están definidas en la especificación:

1. *Partición A*: Contiene la cabecera, los tipos de macrobloques, parámetros de cuantización, modos de predicción y vectores de movimiento.
2. *Partición B*: Contiene información residual de los intra macrobloques codificados.
3. *Partición C*: Contiene información residual de los inter macrobloques codificados.

Las imágenes IDR son siempre puestas en una única NALU; aun si la partición de datos está habilitada. Con la finalidad de saber a que slice, una partición pertenece, un identificador de *slices* se define un elemento en la sintaxis. Cada slice dentro de una imagen codificada tiene un identificador único. A pesar de que la herramienta orden arbitrario de *slices* este activa, el primer slice de una imagen codificada tendrá un identificador de slice igual a cero y el valor de identificador se incrementara en uno para cada slice subsecuente de la imagen

codificada en el orden de decodificación. El propósito de la herramienta de partición de datos es tener la capacidad de usar la información de las particiones correctamente recibidas aun cuando una de las particiones se pierde. Para usar esto, es importante conocer las dependencias entre las diferentes particiones. Para iniciar, es obvio que DPB y DPC [4] no pueden ser decodificadas cuando la correspondiente DPA no está presente. Es más, el decodificador posiblemente no podrá decodificar macrobloques si no conoce los modos de predicción y los vectores de movimientos usados. En otro tema, la información contenida en la partición A, especialmente los vectores de movimiento, pueden ser útiles al decodificador para cancelamiento de pérdidas de las correspondientes particiones B y C, pero esto no significa que funcione de forma inversa. Estas cuestiones se complican cuando se consideran las dependencias mutuas entre las particiones B y C.

En primera instancia, ninguna información contenida en la partición B se necesita para analizar y decodificar la partición C y viceversa. Sin embargo, hay que tener precaución debido a las múltiples dependencias entre macrobloques en H.264/AVC. Dos de esas dependencias tienen un impacto en las dependencias mutuas entre las particiones B y C. Una primera dependencia es causada por el concepto de predicción intra. Muchos de los modos de predicción intra usan píxeles de los macrobloques vecinos para predecir los macrobloques actuales. Esto posiblemente hace a los macrobloques intra codificados depender de los macrobloques inter codificados. Una segunda dependencia puede ser encontrada en la CAVLC, esta codificación usa el número de coeficientes distintos de cero en macrobloques vecinos para analizar el número de coeficientes distintos de cero en el macrobloque actual; esto posiblemente introduzca un posible dependencia entre los macrobloques intra e inter codificados.

Para evitar tales dependencias, H.164/AVC define una opción especial llamada Predicción Intra Limitada. Este modo de predicción restringe al codificador a usar solo datos residuales y muestras decodificadas de los macrobloques intra para la predicción intra. Esto elimina la primera dependencia discutida anteriormente.

3.5.2. Ventajas de la Partición de Datos

Cuando la partición de datos es usada, el comportamiento del decodificador depende del contenido de la NALU perdida:

1. *Partición A perdida*: En este caso las particiones B y C son descartadas dado que son inservibles debido a su dependencia a la partición A. El slice completo es reemplazado por una copia de la imagen anterior. Por lo que perder la partición A tiene el mismo efecto que perder el slice completo en caso de que no se use partición de datos.

2. *Partición B perdida*: Cuando el decodificador detecta que la partición B está perdida, descarta la partición C correspondiente (incluso si está pre-

[4] Partición de Datos B y C respectivamente.

sente), dado que no se puede decodificar sin la información de la partición B. Los macrobloques inter codificados en el slice pueden ser predichos usando los vectores de movimiento de la partición A. Los macrobloques intra codificados sera reemplazados por una copia de la imagen previa.

3. *Partición C*: Dado que las particiones A y B son independientes de la partición C, podemos hacer uso de toda la información contenida en esas particiones para decodificar el slice como si no se hubiese perdido algún paquete. En particular la compensación de movimiento y las predicciones inter e intra pueden ser desarrolladas, la predicción de los macrobloques intra puede ser corregida con la información residual de macrobloques inter, que fueron perdidos con la partición C, se asume sea cero.

3.5.3. Costo de la Partición de Datos

Cuando se usa partición de datos, necesitamos tres NALU's por slice en lugar de uno solo. Cada partición debe especificar un identificador de slice. Además, cada NALU tiene un byte de cabecera NAL y es precedido (Anexo B) en el flujo de bits por un código de inicio de cuatro bytes. Cuando no se usa partición de datos, solo se necesita una NALU por slice y el identificador de slice no se necesita ya que el slice de datos está completo y contenido en la misma NALU. Esto significa que la partición de datos genera un costo de $(3 * (1 + 1 + 4) - (1 + 4)) = 13 bytes$ por slice [17]. Esto es cerca de 5 extra en comparación con el video de baja calidad y menos del 3 para video de alta calidad. Este costo adicional extra es aceptable.

Capítulo 4

Estudio de Pérdidas en video H.264/AVC

4.1. Secuencias de video de Prueba

La selección de las secuencias de video es un componente importante para la evaluación de la calidad de video y el estudio de pérdidas. Es necesario seleccionar secuencias de video que por sus características abarquen un rango suficiente en complejidad de codificación. Seleccionar solo secuencias fácil de codificar haría que se marcara una tendencia favorable (pero no deseable, ya que sería imparcial) en los resultados que se buscan en este capítulo. A continuación se mencionan las secuencias seleccionadas.

Bus

Bus es una secuencia donde se aprecia un autobús en un corto recorrido a través de la calle. Durante el recorrido aparecen en escena otros vehículos, una camioneta rebasa al autobús por la izquierda. El segundo plano está cubierto por gran cantidad de arboles y un monumento, en la parte inferior del primer plano aparece una cerca metálica. Estos elementos representan un desafió para la compresión de video, ya que el nivel de detalles en la escena es sumamente alto, es muy rico en texturas. El movimiento presente en la secuencia es continuo con un pequeño grado de agitación. Por las propiedades del video es una secuencia compleja para codificar en el estándar H.264, ya que para lograr buena calidad de imagen se necesita incrementar el *bit rate*.

Foreman

Esta secuencia contiene la cara de una persona realizando expresiones por lo que es muy rica en mímica. Por otro lado el movimiento en la secuencia no es muy intenso durante la primera mitad pero si un tanto desordenado y sin

(a) Cuadro 22

(b) Cuadro 138

Figura 4.1: Capturas de cuadros de la secuencian de video Bus

un flujo continuo. Durante la segunda mitad tiene un carácter de movimiento intricado que crea problemas para el proceso de compensación de movimiento. Adicionalmente la cámara con la que se grabo la secuencia hace un movimiento tembloroso, lo que hace que la imagen sea inestable. Al final de la secuencia la cámara repentinamente hace un giro hacia la construcción de un edificio continuando con una escena casi con ausencia de movimiento. Esta secuencia es útil para mostrar el comportamiento del codificador en un escena estática posterior a un movimiento intenso. Debido a la cantidad de detalles y nivel de movimiento se puede considerar a Foreman como una secuencia de complejidad media.

(a) Cuadro 17

(b) Cuadro 98

Figura 4.2: Capturas de cuadros de la secuencian de video Foreman bbbbk

Akiyo

Akiyo se trata de una secuencia de video típica de prueba donde aparece una mujer de ascendencia asiática en un programa de noticias. El video se puede considerar plano en cuestión de detalles. El nivel de movimiento en la secuencia es relativamente muy bajo, los detalles más importantes aparecen en el rostro de persona. Es una secuencia donde se encuentra bien definida una región de interés ya que todo espectador que vea el video centrara su atención al centro de la imagen, dejando de lado lo que suceda en los contornos. La tasa de compresión de codificador H.264 tiene un excelente desempeño debido a las características ya mencionadas. En la tabla 4.1 se resumen aspectos de las secuencias de video de prueba.

(a) Cuadro 17

(b) Cuadro 98

Figura 4.3: Capturas de cuadros de la secuencian de video Akiyo

video	*Longitud en Cuadros*	*Duración en Seg.*	*Nivel de Movimiento*	*Textura*
Bus	150	5	Alto	Compleja
Foreman	300	10	Medio	Media
Akiyo	300	10	Bajo	Simple

Tabla 4.1: Características de videos de Prueba

4.2. Codificación de video en el Estándar H.264

Antes de poder codificar un video en formato H.264 es necesario obtener el software de referencia del estándar, el cual puede ser obtenido a través de [14].

Este paquete de software contiene una breve descripción en archivo de texto y el código fuente para poder generar la aplicación de software de H.264.

En una plataforma Windows, la aplicación se ejecuta a través de la linea de comando con instrucciones sencillas que pueden incluir o no diversas opciones de configuración, como el nombre de los archivos de video de entrada y salida por citar un ejemplo. Existen gran cantidad de parámetros de configuración para la codificación de un video, sin embargo, no es imperativo asignar valores a todos ellos. De no establecer valores de configuración se asumirán valores predeterminados. Para la modificación de parámetros en este trabajo se uso un archivo de configuración donde se listan todos los parámetros disponibles y su valor asignado.

Como se mencionó anteriormente H.264 usa distintos parámetros y herramientas de codificación de acuerdo a perfiles y niveles. En la codificación de video se eligió el perfil *Extended* ya que cuenta con relativamente altas capacidades de compresión y sobre todo herramientas de resistencia al error, característica fundamental para realizar el estudio de pérdidas en video. Recordemos que el parámetro *Nivel* de H.264 establece restricciones de las propiedades del video, el nivel seleccionado fue el 4,0.

H.264 permite segmentar cada imagen en varias partes llamadas *slices*, la ventaja de usar *slices* es que pueden ser decodificados independientemente y en paralelo. Sin embargo, usar múltiples *slices* perjudica la eficiencia de compresión, cuanto más *slices* haya por imagen menor sera la eficiencia. Por otro lado si los *slices* son de mayor tamaño y uno de ellos llegase a perderse y no estuviese disponible en la decodificación del video la afectación sería mayor que si el slice fuese de menor tamaño. En la configuración de parámetros del video se eligió que los *slices* fuesen de un tamaño fijo en número de bytes, este valor se fijo a 400 bytes como tamaño máximo. El tamaño de los *slices* está limitado a este valor e implica que puede haber *slices* de menor tamaño. Este parámetro es fundamental para la evaluación de transmisiones de video; ya que en las redes de comunicaciones el tamaño de un paquete está directamente relacionado con su probabilidad de error.

Otra característica a destacar en el archivo de configuración es la activación del control de tasa de transmisión en la codificación. Fundamentalmente es un mecanismo del estándar que permite alcanzar tasas de transmisión cuasicontantes durante toda la longitud de la secuencia.

Para H.264 se puede mantener un umbral predefinido de calidad de imagen independientemente del nivel de movimiento o detalles de una escena. Esto significaría que el ancho de banda necesario para transmitir el video aumentara cuando haya mucho movimiento y disminuirá cuando no. A esta característica se le conoce como tasa de transmisión variable (VBR). Ya que la tasa de transmisión puede variar demasiado, la red de transmisión puede que no sea capaz de soportar estas variaciones, ya que generalmente se cuentan con anchos de banda limitados.

Cuando se activa el control de tasa de transmisión se monitorea la velocidad de transmisión a la que el video necesitara enviarse, en bits por segundo. Si esta

velocidad supera la tasa de transmisión predefinida en el archivo de configuración, entonces el codificador modificara algunos parámetros gradualmente para disminuir la velocidad en esa sección. El principal elemento que modifica la tasa de transmisión es el valor del parámetro de cuantización (QP), para incrementos en la velocidad, el QP aumenta su valor para generar un descenso en la cantidad de bits generados, implicando también que la calidad del video disminuya. Para descensos en la velocidad, el QP disminuye y así también la calidad del video mejora. Básicamente se trata de ajustar que en promedio el video cumpla con la tasa de transmisión deseada (solo permitiendo variaciones muy ligeras) pero también buscando obtener la mejor calidad posible.

El tamaño de Group of Pictures (GOP) del video se establece a diferentes valores para analizar como está relacionado en el desempeño de pérdidas. Los valores elegidos fueron 10, 30 y 60, valores típicos usados en la literatura. El valor del GOP indica al codificador cada cuantas imágenes deberá codificar una imagen de intra.

El número de grupos de *slices* en que se agrupan los MB de cada imagen son dos, ya que con este valor se busca establecer una partición de imagen tipo ajedrez de la herramienta FMO. Finalmente la tasa de transmisión se vario entre distintos valores como 256 Kbps, 512 Kbps y 1 Mbps. Con el fin de evaluar de que manera afectan las pérdidas cuando el video aumenta o disminuye su calidad. En la tabla se resumen parámetros

Parámetro	*Valor(es) asignado(s)*
Frame Rate	30 fps
Resolution	CIF 352 × 288
Frame Pattern	IDR-P-B-P-B ... IDR
ProfileIDC	88=extended
LevelIDC	4.0
IDR Period	10, 30, 60
SlcieMode	Fixed in Bytes
Slice size	400 Bytes
Slice Group	2 SG
FMO	Dispersed
RateControl	Enable
BitRate	256Kbps, 512Kbps, 1Mbps

Tabla 4.2: Configuración del codificador H.264

4.3. Evaluación de la Calidad de video

Las imágenes [1] digitales están sujetas a una gran variedad de distorsiones en su procesamiento, compresión, almacenamiento, transmisión y reproducción que pueden resultar en una degradación visual de la calidad. Para la gran mayoría de las aplicaciones, que tienen como usuario final al ser humano el único método para cuantificar la calidad visual de una imagen que se puede considerar correcto es una evaluación subjetiva ya que existen una serie de fenómenos en el Sistema Visual Humano (SVH) que hacen a la evaluación de la calidad una tarea compleja.

Las pruebas subjetivas, conocidas como Mean Opinion Score (MOS) básicamente consisten en mostrar una serie de videos a espectadores, su opinión es registrada en una escala que va del uno al cinco, donde cinco significa una excelente calidad y uno una pésima calidad. Posteriormente se calcula un promedio para evaluar cada secuencia de video y así obtener una medida cuantitativa de la calidad. Sin embargo, este tipo de evaluación suele ser poco practica de realizar por ser costosa y consumir tiempo. La principal meta de los métodos objetivos de evaluación de imagen es desarrollar métricas que puedan automáticamente predecir la calidad de imagen percibida. Aunque se desearía que las métricas objetivas fueran lo más parecidas en su desempeño a las subjetivas, no todas correlacionan adecuadamente.

4.3.1. PSNR

La Relación Señal a Ruido de Pico PSNR es ampliamente usada como una métrica de calidad o indicador de desempeño de sistemas de procesamiento de imagen y video en el terreno de la investigación y la industria. Como ejemplo la estandarización de codificadores de video aun depende en gran medida del uso del Peak Signal-to-Noise Ratio (PSNR) como indicador de desempeño, como medida de ganancia en calidad para una tasa de codificación especifica.

Algunos estudios han indicado que el PSNR correlaciona escasamente con pruebas subjetivas de calidad. La literatura existente no ofrece evidencia definitiva acerca de la precisión del PSNR como métrica de calidad. Sin embargo, es una métrica de calidad muy popular en el procesamiento digital de imagen y video. La popularidad del PSNR como métrica de calidad surge principalmente del hecho que no representa alto gasto computacional y matemáticamente es fácil de comprender.

Para una secuencia de video el Error Cuadrático Medio MSE entre cada par i de imágenes correspondientes (original y reconstruida)de video se obtiene mediante la ecuación 4.1.

[1] Desde un punto de vista del procesamiento digital puede considerarse a una secuencia de video una sucesión continua de imágenes.

$$MSE(i) = \frac{1}{WH} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} [Y_r(x, y, i) - Y_p(x, y, i)] \tag{4.1}$$

Donde W y H son el ancho y la altura de la imagen respectivamente en píxeles, Y_r y Y_p son los valores de luminancia de las imágenes original y reconstruida respectivamente. El PSNR es expresado expresado en decibeles para cada par i es definido en la ecuación 4.2

$$PSNR(i) = 10log_{10} \frac{I^2}{MSE(i)} \tag{4.2}$$

Donde I es el máximo valor de luminancia de un píxel, para una representación de luminancia de 8 bits por píxel I toma un valor de 255. Un valor general del PSNR para el video es usualmente expresado como un promedio lineal de valores sobre cada imagen de una secuencia, como se expresa en la ecuación 4.3.

$$PSNR = \frac{1}{N} \sum_{i=1}^{N} PSNR(i) \tag{4.3}$$

Donde N representa el número de imágenes en el video.

4.3.2. SSIM

Una señal de imagen a la que se quiere evaluar su calidad puede ser conceptuada como una suma de una señal de referencia original y una señal de error. Una hipótesis de percepción del error adoptada en la literatura afirma que la pérdida de calidad perceptible está directamente relacionada con la visibilidad de la señal de error en la imagen, esta filosofía está reflejada en métricas como el PSNR y MSE. La métrica de calidad de Similitud Estructural SSIM considera la degradación de la imagen como cambios percibidos en la variación de la información estructural de la imagen.

La principal función de la visión humana es extraer la información estructural del campo de visión. El SVH está muy adaptado a este propósito, por lo que medir la distorsión estructural de una imagen ha sido considerado como una buena aproximación de la distorsión percibida. Por lo tanto, al menos para imágenes, la conservación de la estructura de la señal es fundamental. Básicamente la idea es cambiar la perspectiva con la que se aborda la estimación de la calidad entre medir el error y medir la distorsión estructural.

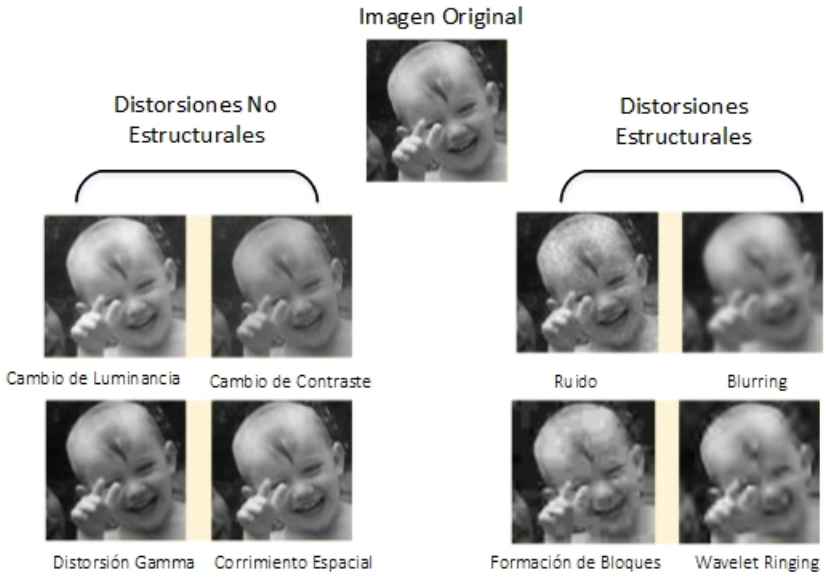


Figura 4.4: Distorsión estructural y no estructural

De acuerdo a [15] las señales de imágenes poseen estructura, implicando que sus píxeles tengan fuertes dependencias, principalmente si son espacialmente cercanos es cuando estas dependencias conllevan información importante acerca de la estructura del objeto en la escena. Con base en esta afirmación, un algoritmo puede buscar medir la distorsión estructural para lograr una medida de la fidelidad de la imagen. La figura 4.4 muestra la distinción entre distorsión estructural y no estructural. Las distorsiones no estructurales como un cambio en la luminancia o brillantez, en el contraste, distorsión gamma y corrimiento espacial son causados por condiciones ambientales ocurridas probablemente durante la adquisición o reproducción de la imagen. Estas distorsiones no alteran la estructura de los objetos en las imágenes. Sin embargo otras distorsiones como ruido aditivo, un aspecto borroso y pérdidas de compresión significan distorsión de la estructura de objetos.

La forma básica del SSIM es relativamente simple de analizar. Si consideramos x y y como dos secciones de imagen tomadas en la misma localización espacial para ser evaluadas. El SSIM mide las similitudes en tres elementos: la brillantes $l(x, y)$, el contraste $c(x, y)$ y la similitud de estructura $s(x, y)$. Estos elementos son obtenidos a través de medidas estadísticas simples como se muestra en la ecuación 4.4-4.6, para ser combinadas conjuntamente y formar el SSIM.

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{4.4}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{4.5}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{4.6}$$

Donde μ_x y μ_y son respectivamente las medias de x y y, σ_x y σ_y son respectivamente la desviación estándar de las muestras de x y y y σ_{xy} es la correlación cruzada de x y y después de haber restado su media. Finalmente se combinan las ecuaciones 4.4-4.6 para formar el SSIM.

$$SSIM(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [s(x,y)]^{\gamma} \tag{4.7}$$

Donde $\alpha > 0,\ \beta > 0$ y $\gamma > 0$ son parámetros usados para ajustar la importancia relativa de los tres componentes. Con el fin de simplificar la expresión, se establece $\alpha = \beta = 1$ y $C_3 = \frac{C_2}{2}$. El resultado es la expresión final del indice SSIM es como lo muestra la ecuación 4.8.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.8}$$

El indice SSIM es simétrico: $SSIM(x,y) = SSIM(y,x)$, de tal forma que dos imágenes que son comparadas dan el mismo resultado sin importar su orden. Esta métrica está limitada a un conjunto de valores posibles: $-1 < SSIM(x,y) \leq 1$, obteniendo un valor máximo de 1 solo cuando las imágenes son idénticas $x = y$. Para la evaluación de calidad realizada a las secuencias de video en este trabajo se busca el valor del SSIM para cada cuadro de video pero también un solo valor único del SSIM sobre todos los cuadros M de X y Y, para tal caso se realiza un promedio lineal como se indica en la ecuación 4.9.

$$MSSIM(X,Y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_i, y_i) \tag{4.9}$$

4.3.3. VQM

Video Quality Metric (VQM) es una herramienta para medir la calidad del video muy representativa que está basada en transformaciones lineales y no lineales de métricas de distorsión. El calculo de VQM involucra la obtención de características basadas en la percepción, calculo de parámetros de calidad de video para posteriormente combinar los parámetros y obtener un resultado final. Deben llevarse a cabo una serie de procesos de evaluación del video antes de poder combinar los resultados. Se debe tomar como referencia el video original para evaluar el video procesado o corrupto. Los siguientes etapas tienen como objetivo hacer más confiable el VQM como métrica de estimación de la calidad con medio de un pre-procesamiento de las señales.

Alineación Espacial: El proceso de alineación espacial determina el corrimiento vertical y horizontal del video corrupto. La precisión del alineamiento espacial es de medio píxel. Después de que el corrimiento es determinado se remueve del video, por ejemplo si se obtiene un corrimiento de dos píxeles hacia abajo, esta misma cantidad de píxeles se recorrerá el video hacia el sentido contrario.

Región Valida Procesada: Este proceso tiene el objetivo de prevenir áreas de imagen que contengan secciones no validas (como bordes negros de píxeles en los contornos del cuadro de imagen), ya que estas áreas influencian el calculo de VQM, por lo que deben ser excluidas. El comportamiento de muchos sistemas de video depende de la escena actual por lo que no es conveniente fijar la región valida para toda una secuencia de video, idealmente la región valida procesada debería ser calculada imagen a imagen. Después del calculo de la región valida, los píxeles son descartados del video original y corrupto.

Compensación de Ganancia: En este proceso se lleva a cabo una calibración de ganancia de los componentes de crominancia y luminancia. Es necesario que las secuencias a comparar estén espacialmente acopladas y temporalmente sincronizadas previamente. Un ajuste lineal, de primer orden, se usa para calcular la ganancia relativa entre la secuencia original y procesada.

Alineación Temporal: Esta sección presenta una técnica para estimar el retardo del video basada en las imágenes de la secuencia original y procesada. La técnica funciona correlacionando versiones de baja resolución de los cuadros de la secuencia original y procesada, que son sub-muestreados espacialmente. El retardo del video puede depender de atributos dinámicos como el nivel de detalle espacial y movimiento. Por ejemplo, escenas con gran cantidad de movimiento pueden son mayormente afectadas que las escenas con poco movimiento.

Para el calculo del VQM se emplea un modelo general que contiene siete parámetros independientes. Cuatro de ellos están basados en características ex-

traídas de gradientes espaciales de los componentes de luminancia, dos parámetros están basados en las características de vectores formados por los dos componentes de crominancia ($C_{B,\,C_R}$), y el ultimo está basado en un producto de las características que miden contraste y movimiento, ambos obtenidos solo del componente de luminancia. A continuación un breve lista de los parámetros:

- **si_loss:** Este parámetro detecta pérdida de información espacial, un claro ejemplo es el que la imagen se vea borrosa (blurring). Se usa un filtro espacial especialmente diseñado para medir distorsiones de bordes.
- **hv_loss:** Sirve para detectar corrimientos diagonales de los bordes horizontales y verticales de objetos.
- **hv_gain:** Tiene la función de detectar corrimientos de bordes diagonales en dirección horizontal y vertical.
- **chroma_spread:** Detecta cambios en la dispersión de la distribución de los componentes de color.
- **si_gain:** Es el único parámetro que mejora la calidad en el modelo. Mide las mejoras en la calidad que resulta de la definición de bordes.
- **ct_ati_gain:** La percepción de las distorsiones espaciales pueden ser influenciadas por la cantidad de movimiento presente. De igual forma, la percepción de distorsiones temporales puede ser influencia por la cantidad de detalles espaciales. Este parámetro se deriva del producto de información de contraste e información temporal.
- **chroma_extreme:** Detecta distorsiones severas de color localizadas, como aquellas producidas por secciones completas corrompidas.

El modelo general para la obtención del VQM está optimizado para lograr máxima correlación con los modelos de calidad subjetivos dentro de un amplio rango de tasas de transmisión. Este modelo consiste en una combinación lineal de los parámetros enumerados anteriormente. Este modelo produce valores de salida en el rango de cero a uno donde cero representa la mejor calidad o sin distorsión y uno representa la máxima distorsión percibida. El VQM se obtiene a través de la combinación lineal de los siete parámetros como se muestra en la ecuación 4.10.

$$\begin{aligned} VQM &= -0{,}2097 * \boldsymbol{si\_loss} + 0{,}5969 * \boldsymbol{hv\_loss} \\ &+ 0{,}2483 * \boldsymbol{hv\_gain} + 0{,}0192 * \boldsymbol{chroma\_spread} \\ &- 2{,}3416 * \boldsymbol{si\_gain} + 0{,}0431 * \boldsymbol{ct\_ati\_gain} \\ &+ 0{,}0076 * \boldsymbol{chroma\_extreme} \end{aligned} \tag{4.10}$$

4.4. Esquema de Pérdidas

Una vez codificados los videos de prueba son comprimidos y cuentan con un formato H.264, el video está dividido en una cabecera principal y *slices*. Esta cabecera contiene parámetros de configuración de secuencia, el objetivo de estos es fundamentalmente comunicar al receptor información útil para decodificar el video, que no cambia frecuentemente con el propósito de desacoplar el intercambio de información constante.

La cabecera del archivo bajo cualquier circunstancia debe mantenerse inalterada ya que la falta de esta sección dentro del archivo H.264 haría que no fuese posible la decodificación. Por este motivo el en el estudio de pérdidas no se considera el caso donde la cabecera este ausente en la decodificación. Cabe señalar que de ahora en adelante cuando se muestre una lista de *slices* es posible que no se muestre las secciones de cabecera, solo por motivos prácticos.

Independientemente de cualquier patrón de codificación de imagen seleccionado como *IDR-B-P-B- ... -IDR* o *IDR-B-B-P-B-B- ... -IDR* la primera imagen de cualquier secuencia se codificara como una imagen tipo IDR , por lo que su codificación es de tipo intra (no se incluyen dependencias de otros cuadros para su decodificación). Las imágenes IDR que se codifiquen posteriores al inicio, tendrán la función de evitar la propagación del error. Posterior a la primera imagen IDR, todas las demás imágenes dentro del GOP poseen dependencias de codificación. Estas dependencias se deben a la explotación de las redundancias temporales y espaciales, gracias a las cuales es posible obtener grandes tasas de compresión.

Ya se han explicado algunas de las principales configuraciones del decodificador, sin embargo, en la tabla 4.3 se enumeran los videos que serán utilizados y las características que los diferencian. Una de estas características es la tasa de transmisión, que está directamente relacionada con la calidad de salida, con la intensión de cubrir aplicaciones de baja y alta capacidad de ancho de banda. El segundo parámetro es el tamaño del GOP que está relacionado con la propagación del error.

El resto de columnas de la tabla 4.3 representan la composición del video en sus diferentes tipos de *slices* (IDR, P y B) así como la cantidad total de *slices* del mismo. Una observación sobre estos datos es que los videos que pertenecen a la misma secuencia y mismo *bit rate* se caracterizan por tener un total de *slices* similar. Sin embargo, su distribución en tipo es distinta. Como ejemplo tomemos el caso de la secuencia Foreman con *bit rate* de 512 Kbps donde el total de *slices* para los GOP de 10 y 60 son 1896 y 1879 respectivamente, pero el número de *slices* tipo IDR es radicalmente diferente con 892 y 142 respectivamente. Este comportamiento a su vez se presenta en los *slices* tipo P. Para los *slices* tipo B no se observan estas discrepancias.

Antes de abordar directamente las pérdidas en el video, es fundamental analizar como está compuesto el archivo en formato *.264 para tomar consideraciones pertinentes. Dentro del archivo está contenida toda la información necesaria que el decodificador requiere para ser capaz de decodificar el video y reproducirlo al

video	Tasa de Transmisión	Tamaño GOP	*Slices* IDR	*Slices* P	*Slices* B	*Slices* Totales
1.- Akiyo	128 Kbps	10	422	240	297	959
2.- Akiyo	128 Kbps	60	104	403	297	804
3.- Akiyo	256 Kbps	10	697	252	297	1246
4.- Akiyo	256 Kbps	60	104	776	297	1177
5.- Foreman	512 Kbps	10	892	687	317	1896
6.- Foreman	512 Kbps	60	142	1352	385	1879
7.- Foreman	1 Mbps	10	1556	1428	599	3583
8.- Foreman	1 Mbps	60	295	2509	751	3555
9.- Bus	512 Kbps	10	409	349	167	925
10.- Bus	512 Kbps	60	100	652	231	983
11.- Bus	1 Mbps	10	687	718	378	1783
12.- Bus	1 Mbps	60	196	1179	437	1812

Tabla 4.3: Indice de videos para esquema de pérdidas

usuario final. Para ver la composición del video se genero una indice de *slices*, al que nombraremos **tracefile**, que enumera los *slices* generados por cada imagen y algunas de sus características asociadas, como se muestra en la tabla 4.4. En la primera columna tenemos el número de slice, es único y su numeración es ascendente solo para fines de identificación. En la segunda columna está el *timestamp* que indica el instante de tiempo al que pertenece el *slice*. La columna siguiente es el tamaño del slice en bytes, en este ejemplo vemos distintos valores desde muy bajos hasta cercanos a los 400 bytes, ya que precisamente este es el valor que se asigno al parámetro *slice_size* en la codificación. La cuarta columna nos dice el tipo de imagen al que pertenece el slice. La combinación $5-7$ es para una imagen tipo IDR, $1-5$ es para una imagen tipo P, la combinación $1-6$ es para una imagen tipo B y todas las demás combinaciones señalizan cabeceras. La ultima columna de la tabla indica el número de imagen del que forma parte el slice.

La Capa de Abstracción de Red (Network Abstraction Layer) define la interfaz entre el la codificación de video y el mundo externo. Opera con Unidades de Abstracción de Red (NALUs), las cuales dan soporte a transmisiones basadas en paquetes. La interfaz NAL del decodificador asume que las NALU son recibidas en el orden en que fueron enviadas y que los paquetes están libre de error, perdidos o con una bandera activa señalizando presencia de error.

Una NALU consiste de un byte de cabecera y una cadena de bits que son

Número de Slice	Timestamp	Tamaño en Bytes	Tipo de Frame	Frame
0	0.000000	13	7 -1	0
1	0.000000	9	8 -1	0
2	0.033333	393	5 7	1
3	0.033333	396	5 7	1
4	0.033333	380	5 7	1
5	0.033333	387	5 7	1
6	0.033333	391	5 7	1
7	0.033333	401	5 7	1
8	0.033333	388	5 7	1
9	0.033333	113	5 7	1
10	0.033333	399	5 7	1
11	0.033333	403	5 7	1
12	0.033333	387	5 7	1
13	0.033333	404	5 7	1
14	0.033333	394	5 7	1
15	0.033333	401	5 7	1
16	0.033333	398	5 7	1
17	0.033333	50	5 7	1
18	0.066667	45	1 5	2
19	0.066667	33	1 5	2
20	0.100000	12	1 6	3
21	0.100000	12	1 6	3
22	0.133333	147	1 5	4
23	0.133333	115	1 5	4
24	0.166667	18	1 6	5
25	0.166667	20	1 6	5

Tabla 4.4: *Tracefile* de Salida en H.264

(en la mayoría de los casos) bits representando los MB de un slice. El byte de cabecera está compuesto por la antes mencionada bandera de error de un bit de longitud (*forbidden_zero_bit*), una bandera NAL disponible de dos bits (*nal_ref_idc*) y el tipo de NALU (*Type*) que ocupa los restantes cinco bits. La bandera *nal_ref_idc* NAL puede ser usada para señalizar si una NALU contiene información para reconstruir imágenes de referencia en la predicción inter, y así poder evaluar si su falta generaría propagación de error en imágenes subsecuentes.

El campo *nal_ref_idc* tiene el potencial para ser usado con fines más prácticos por lo cual surgió la idea de utilizar este parámetro de dos bits como una forma de identificar la importancia de una NALU y aprovechar este conocimiento en las redes de transporte. Esta idea va rodeada de muchas otras cuestiones complementarias, como por ejemplo, con base en que característica podría medirse la importancia de un slice y como asignar un nivel de prioridad de forma sencilla y eficiente.

Para responder estas cuestiones se analizó la forma en que un slice podría impactar en la experiencia del usuario final, que finalmente es lo más primordial. Cada slice codificado conlleva información de parámetros de imagen y MBs codificados, estos *slices* pueden pertenecer a imágenes IDR, P o B. Aquí precisamente surge una afirmación que luce evidentemente como obvia: por las dependencias de codificación se puede suponer que siempre serán más importantes los *slices* que pertenezcan a imágenes de referencia. Esta hipótesis tiene una excelente fundamentación lógica, sin embargo es interesante y necesario probarla.

Una forma acertada de abordar la hipótesis anterior acerca de la importancia de un slice es midiendo su impacto directo en la calidad del video cuando se decodifica y este no está disponible o simplemente está corrompido por alguna razón ajena al decodificador. La magnitud del descenso de calidad reflejara de forma más objetiva la veracidad de esta afirmación.

La estrategia que se uso consiste en descartar cada uno de los *slices* que forman un video en la decodificación. Cuando el decodificador detecta la falta de un slice es capaz de realizar la decodificación sin inconveniente alguno. Evidentemente existirá un impacto en la secuencia, la magnitud y distribución de este impacto es de sumo interés. La forma de cuantificar el impacto es midiendo la calidad del video.

Para la medición de la calidad del video se han expuesto distintos enfoques de métricas. Es de radical importancia seleccionar la más adecuada según el contexto que se maneje, en algunas ocasiones es más valioso la confiabilidad de la métrica y en otras su eficiencia.

El VQM es una métrica muy confiable ya que correlaciona muy bien con la percepción del ser humano, sin embargo, es compleja, consume recursos y emplea bastante tiempo de procesamiento. De acuerdo al proceso de calculo del VQM, esta métrica arroja un solo valor de salida por cada secuencia completa de video evaluada y no es posible obtener la métrica a nivel de cuadros de imagen. Estas cuestiones hacen del VQM una métrica no factible para la medición de la calidad. Por otro lado el SSIM es menos compleja que el VQM y tiene un enfoque distinto para evaluar la calidad. Recordemos que el SSIM evalúa una la secuencia en función de medidas estadísticas de los elementos de imagen como la media, covarianza y coeficiente de correlación. Estas medidas reflejan la similitud estructural. Sin embargo, en pruebas realizadas se midió la calidad de imágenes con distintos grados de degradación oscilando desde ligeramente perceptibles hasta muy notorios; se observo que la distribución de los resultados de calidad poseían diferencias mínimas que no denotaban claramente las diferentes degradaciones introducidas, por lo que se decidió no usar esta métrica en esta sección del estudio de pérdidas.

Finalmente la métrica seleccionada fue el MSE, esta métrica tiene características muy atractivas: es muy sencilla, rápida y eficiente de calcular, emplea bajos recursos computacionales y adicionalmente el rango de valores de salida es bastante amplio. Un punto no favorable es que no se ajusta tan bien como el VQM al Sistema Visual Humano pero todas las demás características favorables

la hacen muy practica. El MSE está libre de parámetros de entrada, no tiene memoria y satisface adecuadamente las siguientes condiciones:

- **No negatividad:** $MSE(x, y) \geq 0$
- **Identidad:** $MSE(x, y) = 0$ si y solo si $x = y$
- **Desigualdad Triangular:** $MSE(x, z) \leq MSE(x, y) + MSE(y, z)$

El MSE cuenta con un significado físico, de forma natural se puede definir como la energía de la señal de error, de acuerdo con el teorema de Parseval esta energía se preserva después de alguna transformación lineal como la transformada coseno.

Retomando la estrategia de pérdidas, se elimino cada slice en la decodificación posteriormente se midió la calidad de cada imagen a lo largo de la toda la secuencia. Si solo se mide la distorsión en la imagen a la que pertenece el slice se tiene una medición del impacto en forma parcial. Evidentemente el slice perdido afectara la calidad de la imagen a la que pertenece pero existe la posibilidad de que la afectación continue. Razón por la cual se cuantifica el MSE en cada imagen y se acumula, el resultado es el Cumulative Mean Squared Error (CMSE). El valor del CMSE representa cuantitativamente el nivel de afectación del slice. La acumulación del MSE solo debe realizarse a partir de la imagen donde ocurre la pérdida y hasta el final del GOP al que pertenece la imagen. Posterior a la terminación de todo GOP a excepción del final de secuencia está presente una imagen tipo IDR, que por la forma en que está codificada detiene la propagación del error.

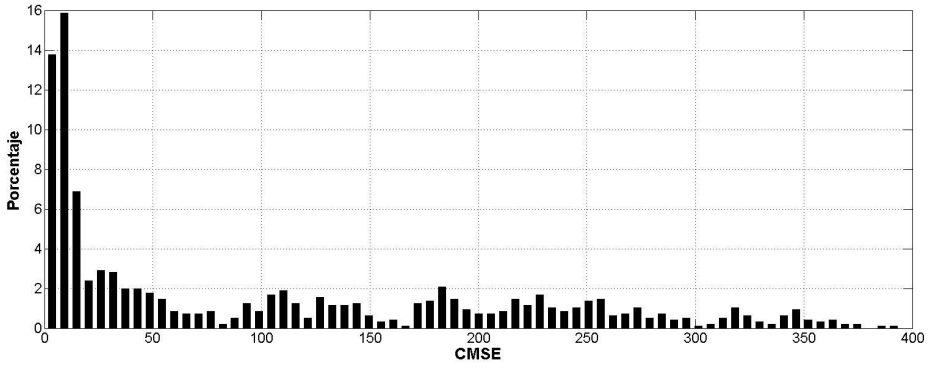
4.5. Estadísticas de Pérdidas

Habiendo realizado las pérdidas expuestas en el tema anterior, cada *slice* de toda secuencia tiene asociado un valor de CMSE. Este valor representa el grado de afectación que provocaría en el video su pérdida. La distribución y magnitud de las pérdidas son valiosos parámetros que ayudaran a comprender la inter-relación de codificación y las pérdidas en el estándar H.264.

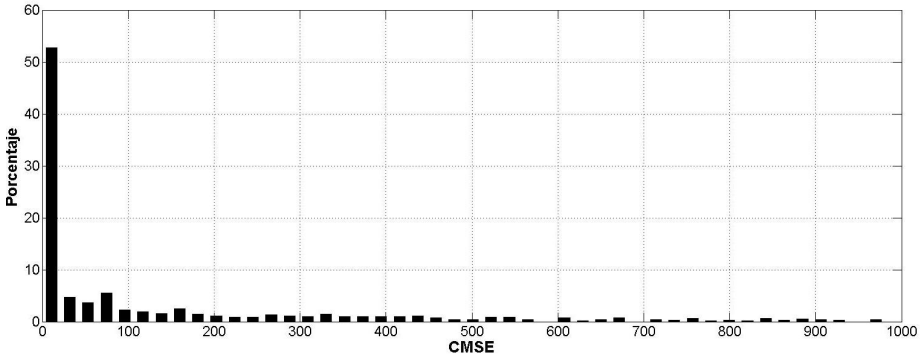
4.5.1. Histogramas del CMSE

En esta sección se presentan histogramas de la distribución del CMSE para algunas secuencias. En el eje horizontal se coloca el CMSE; el rango de los valores varia de acuerdo a la secuencia mostrada. Los histogramas muestran la distribución de los valores de CMSE obtenidos por los *slices*, la distribución se muestra en forma de barras con una altura que indica el porcentaje de *slices* situados en el intervalo de valores de CMSE que abarca el ancho de la barra. En el eje vertical representa el porcentaje de los *slices* que se sitúan en cada barra.

En la figura 4.5 se muestran las gráficas correspondientes a los histogramas del CMSE para los *slices* de la secuencia Akiyo con *bit rate* de 128 Kbps. Es valioso notar las distribuciones que tienen, para la gráfica con $GOP = 10$ los *slices* tienden a tener valores de CMSE bajos, inferior a 50 para ser más específicos. Así mismo, el resto de los *slices* cuentan con CMSE de más de 50 y menor a 400 pero con una distribución más dispersa. Para la gráfica con $GOP = 60$, la tendencia es sumamente marcada, más de la mitad de los *slices* están contenidos en la zona con CMSE inferior a 50 pero a diferencia de la gráfica anterior existen *slices* con valores de CMSE mucho más alto, algunos toman valores cercanos a 1000, se puede intuir que los *slices* con mayor CMSE muy probablemente serán de tipo IDR, ya que estos se ocupan de referencia para todo el GOP, y si a esto sumamos que la longitud del GOP sea extensa, la propagación del error sería muy larga provocando que el CMSE atribuido al *slices* IDR crezca considerablemente. Cabe recordar las características de la secuencia a la que pertenecen estas gráficas son una textura de imagen suave y bajo nivel de movimiento.

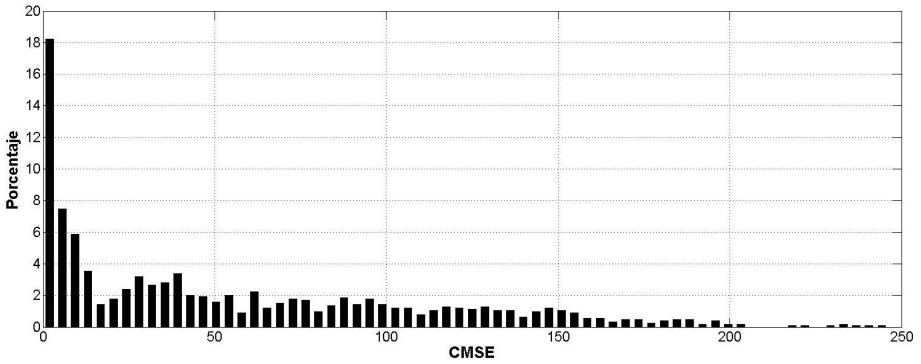


(a) Akiyo GOP 10

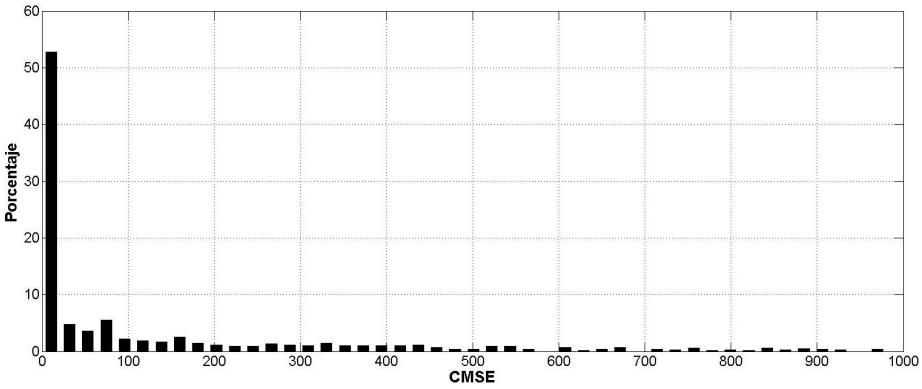


(b) Akiyo GOP 60

Figura 4.5: Histogramas CMSE de la secuencia Akiyo *Bit Rate* de 128 Kbps



(a) Akiyo GOP 10

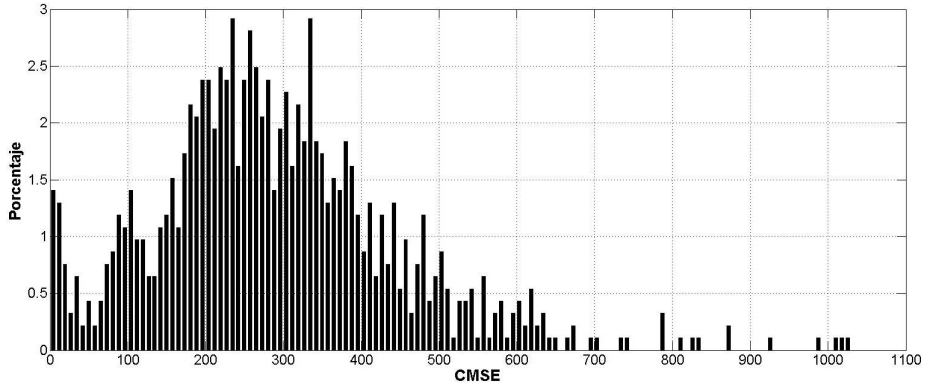


(b) Akiyo GOP 60

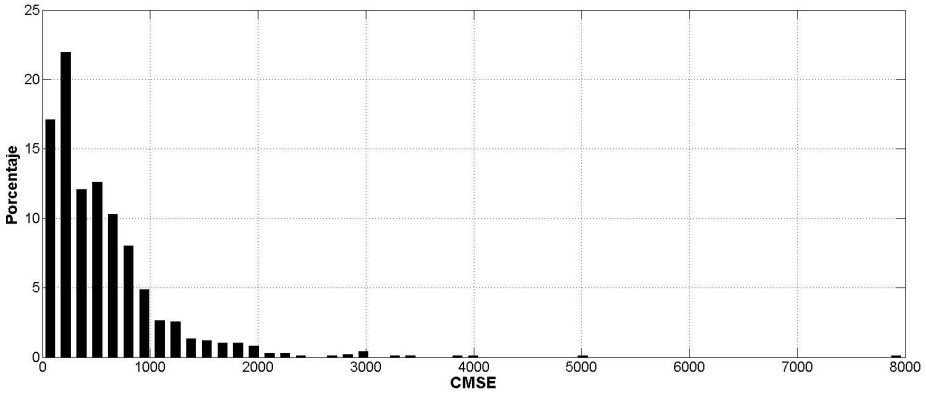
Figura 4.6: Histogramas CMSE de la secuencia Akiyo *Bit Rate* 256 Kbps

El gráfico correspondiente a la figura 4.6 es similar a la gráfica de la figura 4.5. La más importante diferencia entre este par de figuras radica en el rango de valores que toman los histogramas, más que en que en las distribuciones mismas. El gráfico de esta figura pertenece a una secuencia que se codifico con un *bit rate* del doble con respecto a la referencia anterior, esto implica una mejor calidad de salida en la codificación, por lo que cuando se mide la calidad con el CMSE el rango de valores que esta métrica tome, disminuirá. Respecto a las distribuciones de los histogramas cuentan con bastantes similitudes.

En la figura 4.7 se aprecian los histogramas de la secuencia Bus para un *bit rate* de 256 Kbps. Para un $GOP = 10$, a diferencia de la secuencia Akiyo la distribución de *slices* no se centra en valores cercanos a cero. El histograma posee la forma de una distribución de binomial. La mayor cantidad de *slices* están situados en valores de alrededor de 200 y 300, siendo esta una diferencia importante atribuida a las características propias de las secuencias.



(a) Bus GOP 10

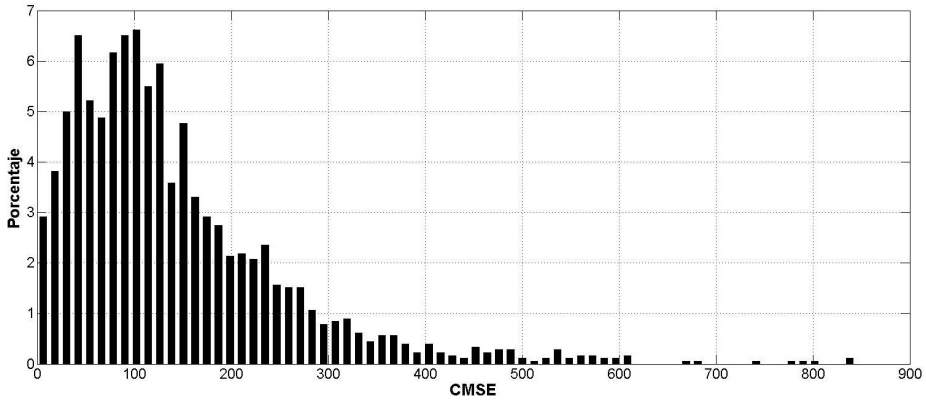


(b) Bus GOP 60

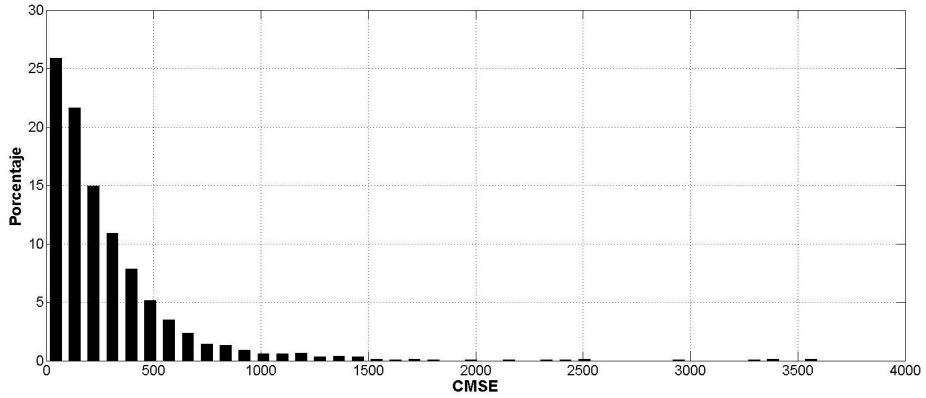
Figura 4.7: Histogramas CMSE de la secuencia Bus *Bit Rate* de 512 Kbps

Las gráficas de la figura 4.8 contienen los histogramas de la secuencia Bus con *bit rate* de 1 Mbps. Se percibe una distribución similar a las gráficas de la misma secuencia con *bit rate* de 512 Kbps, como ya se sabe, esto repercute directamente en la calidad del video. Así, el rango de valores CMSE donde se distribuyen los *slices* se ve solamente recorrido hacia un intervalo inferior respecto a las gráficas de 512 Kbps.

Con base en los histogramas presentados se corroboro que existen altas dependencias de codificación y las pérdidas de *slices* degrada en gran medida la calidad. También se observa que existe un porcentaje variable de *slices* en toda secuencia con valores relativamente bajos de CMSE, precisamente aquí es donde se considera factible asignarles a esos *slices* el menor nivel de importancia o prioridad. Para complementar la idea, conforme se incrementan gradualmente



(a) Bus GOP 10



(b) Bus GOP 60

Figura 4.8: Histogramas CMSE de la secuencia Bus *Bit Rate* 1 Mbps

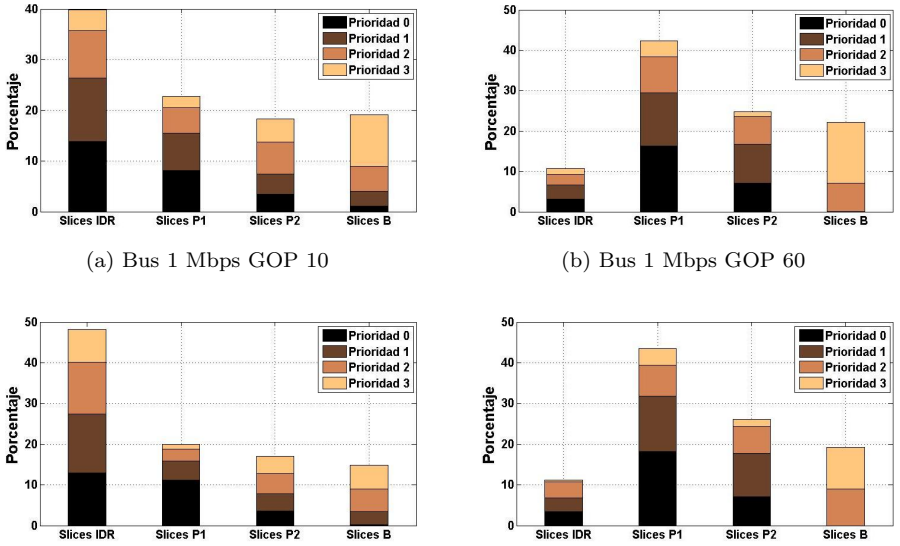
los valores de CMSE también incrementar el nivel de prioridad para señalizar que esos *slices* deben ser menos susceptibles a ser perdidos.

4.5.2. Asignación de Prioridades

Con el conocimiento de la distribución del CMSE que introducen los *slices* en la calidad, se propone dividir la totalidad de los *slices* que componen una secuencia de video en cuatro niveles de prioridad. Se ha argumentado con anterioridad que son cuatro niveles de prioridad ya que a nivel de la Capa de Abstracción de Red se maneja un campo de dos bit con capacidad de asignarlo como prioridad de un paquete de video codificado. Por consiguiente, el veinticinco por ciento de los *slices* totales estarán contenidos en cada prioridad. Los niveles de prioridad

asignados son numerados del cero al tres, donde cero representa el nivel de prioridad más alto o dicho de otra forma, a este nivel de prioridad serán asignados los *slices* con mayor CMSE y que introducen mayor degradación. El nivel de prioridad tres es el más bajo, a este nivel de prioridad serán asignados los *slices* con menor CMSE y que introducen menor degradación. Los niveles uno y dos de prioridad son niveles intermedios que siguen la idea expuesta.

La asignación de prioridades a los *slices* se realiza de forma transparente al tipo de slice. No importa de que tipo de slice se trate, en la asignación de la prioridad se ordenan los *slices* en forma ascendente en función del valor del CMSE que posean. Al primer 25 % de *slices* se les asigna la prioridad tres, la menos importante; al siguiente 25 % de *slices* se les asigna la prioridad dos y así consecuentemente.



(a) Bus 1 Mbps GOP 10

(b) Bus 1 Mbps GOP 60

(c) Bus 512 Kbps GOP 10

(d) Bus 512 Kbps GOP 60

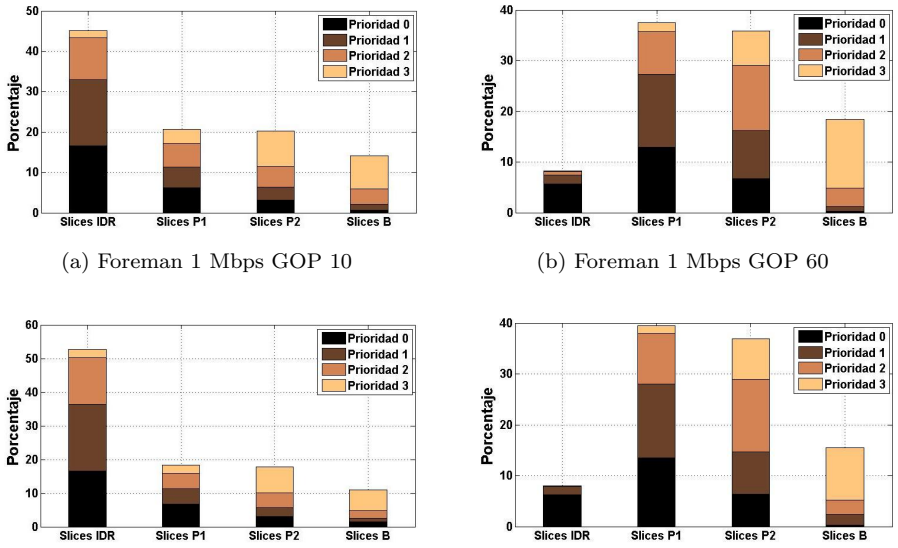
Figura 4.9: Distribución de Bits a través de tipo de imagen y prioridad en la secuencia Bus

Surge el interés de saber como esta compuesto cada bloque de prioridad, se puede suponer que la prioridad tres está constituida mayoritariamente de *slices* tipo B pero surge la interrogante si existen *slices* tipo IDR dentro de esta categoría. De forma análoga en la prioridad cero se supone anticipadamente que los *slices* que conforman este bloque son mayormente de tipo IDR pero queda la duda si aparecen *slices* tipo B en esta categoría.

Como ya vimos algunas secuencias codificadas están compuestas con gran mayoría de *slices* P, este tipo de *slices* toman especial consideración por su gran cantidad. Se decidió realizar una distinción de los *slices* P que pertenecen a

la primera mitad del GOP y los que pertenecen a la segunda, para analizar de mejor forma los datos. Los tamaños de GOP empleados son 10 y 60 por lo que la separación se hace en los cuadros de imagen 5 y 30, y las imágenes con número de secuencia que sean múltiplos de estos valores.

El tamaño del *slice* en la codificación se limitó a 400 bytes, pero existen muchos *slices* con tamaño muy inferior a este valor. Cuanta mayor información se pierda por un *slice* el grado de afectación crecerá. Otra recurso útil para el estudio de pérdidas es el conocimiento de la forma en que se distribuye el tamaño total de una secuencia codificada en las prioridades. La presentación de la distribución de prioridades a través de gráficas se realiza con base en el porcentaje de bytes que representa cada prioridad respecto del tamaño total de la secuencia completa y también el tipo de *slice* se considera . A continuación se vera como están repartidas las prioridades en este tipo de *slices*.



(a) Foreman 1 Mbps GOP 10

(b) Foreman 1 Mbps GOP 60

(c) Foreman 512 Kbps GOP 10

(d) Foreman 512 Kbps GOP 60

Figura 4.10: Distribución de Bits a través de tipo de imagen y prioridad en la secuencia Foreman

En figura 4.9 están las distribuciones de tamaño que ocupan los *slices* en bits para la secuencia Bus, la clasificación se hizo a partir del tipo de imagen al que pertenecen los *slices* y los cuatro niveles de prioridades asociados. Cada barra completa simboliza el porcentaje de todos los *slices* de un tipo en especifico (IDR, P y B). A su vez cada barra está subdividida en las contribuciones que tiene cada nivel de prioridad. Existen casos donde las barras contienen todas las prioridades pero también hay otros donde solo contienen una sola. Existen cuatro barras ya que los *slices* de tipo P se subdividieron en dos categorías:

Slices P1 es la primera mitad del GOP y *Slices P2* la segunda mitad.

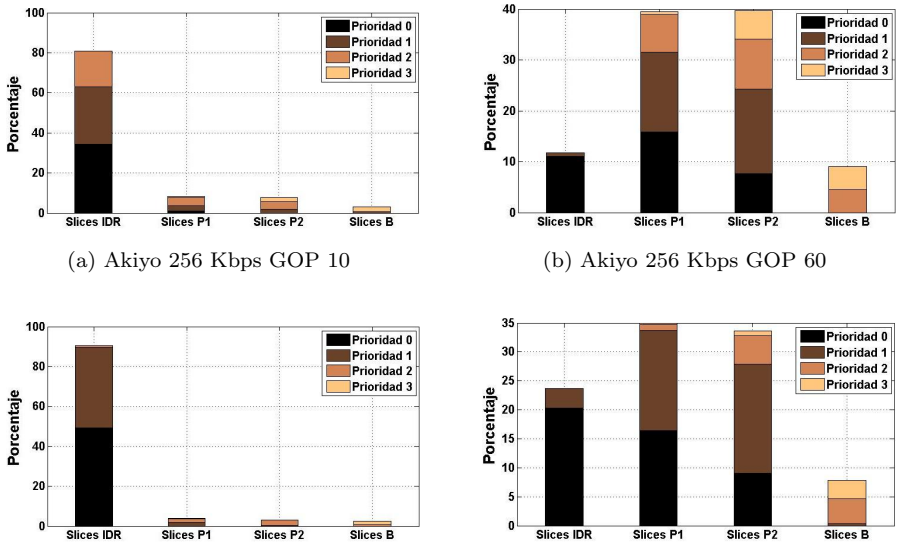
En la figura 4.9c se observa que los *slices* tipo IDR representan el cuarenta por ciento del total del video, dentro de este porcentaje están presentes las cuatro prioridades pero cuenta con mayor contribución la prioridad cero. En los *slices* de las demás barras de la misma forma, están presentes las cuatro prioridades pero en distinta proporción. Si se suman los porcentajes de las barras de *slices* tipo P1 y *slices* tipo P2 su tamaño se asemeja al de los *slices* tipo IDR, sin embargo, para la barra de *slices* tipo B el porcentaje es alrededor del quince por ciento debido a las fuertes tasas de compresión para este tipo de imágenes en el estándar. Cada tipo de imagen tiene funciones especificas, como las imágenes IDR que por su codificación intra detienen la propagación de error en caso de pérdidas anteriores a su aparición y las imágenes tipo B son las que aumentan la eficiencia de compresión pero son mas intensamente afectadas por la propagación del error.

Entre la figura 4.9c y 4.9d la única diferencia es el tamaño del GOP, con valores de 10 y 60 respectivamente. Para el video con $GOP = 60$ si se suman las contribuciones de los *slices* P de la primera y segunda parte, el resultado es que representan la mayor parte del video, alrededor del setenta por ciento. Aquí las imágenes IDR son menos frecuentes (tan solo cinco imágenes IDR) si se compara con el video con $GOP = 10$ (que cuenta con 29 imágenes IDR). Las imágenes IDR ocupan un tamaño considerablemente más grande si se les compara con las tipo P y las tipo B.

En la secuencia Akiyo de la figura 4.11 para las secuencias con GOP 10 en ambos *bit rates*, es abrumador el porcentaje de bits que ocupan los *slices*-IDR, con un porcentaje alrededor del noventa por ciento, significa que para una secuencia de 300 cuadros de video, 29 cuadros ocupan el noventa por ciento del tamaño en bits y 270 cuadros solo ocupan el diez por ciento. Esto nos da una idea clara de la enorme diferencia de compresión que existe entre imágenes IDR y las demás.

En la sección anterior surgió la hipótesis de que en la prioridad cero estarían contenidos todos los *slices* tipo IDR y que en la prioridad tres solo estarían los *slices* tipo B. Ahora con las gráficas de las distribuciones de prioridad y tipo de *slice* se puede responder objetivamente a estas interrogantes. A partir de observar las gráficas podemos concluir que efectivamente los *slices* tipo IDR mayoritariamente cuentan con prioridad cero, pero además también en muchos casos están presentes las demás prioridades e inclusive algunos *slices* tipo IDR pertenecen a la prioridad más baja, la prioridad tres. La justificación para este fenómeno es que estos *slices* IDR con baja prioridad contienen áreas de imagen con características planas en textura y movimiento. Por lo que ante su pérdida las herramientas de cancelamiento de error de H.264, pueden restaurar fácil y eficientemente la zona sin introducir degradaciones visibles. Es obvio que impactara en la métrica de calidad pero de forma ligera.

En el caso de los *slices* tipo B ciertamente cumplen con la suposición de que estarían en la prioridad más baja pero también se presentan casos en donde este tipo de *slices* contienen la prioridad cero, algo poco esperado. Este hecho



(a) Akiyo 256 Kbps GOP 10

(b) Akiyo 256 Kbps GOP 60

(c) Akiyo 128 Kbps GOP 10

(d) Akiyo 128 Kbps GOP 60

Figura 4.11: Distribución de Bits a través de tipo de imagen y prioridad en la secuencia Akiyo

se atribuye a que en este tipo de imágenes los *slices* prioridad cero están áreas codificadas con alto contenido de información que no pueden ser interpoladas o ser reconstruidas por dependencias de codificación.

A partir de la observación de estas gráficas se deduce que las barras con mayor porcentaje de bytes contienen la mayor cantidad de *slices* con altas prioridades (prioridades cero y uno), estas barras son la de *slices* IDR y P1 (primera mitad del GOP). Así mismo se cumple que en las barras con menor porcentaje de bytes estan mayoritariamente ocupadas por *slices* de prioridades bajas (prioridades dos y tres), las barras son de *slices* tipo P2 y B.

Capítulo 5

Política de Pérdidas

La asignación de prioridad a *slices* puede ser útil en muchas aplicaciones como redes de servicios diferenciados, *packet sheduling* en redes de transmisión inalámbricas, sistemas de protección desigual contra errores para canales propensos a distorsiones y control de tráfico para aplicaciones de *streaming*.

La calidad del video es influenciada por varios factores que dependen de la red de transmisión y otros relacionados con la aplicación, como las técnicas de recuperación ante errores y la configuración de la codificación. Es de vital importancia entender las relaciones entre errores en ráfaga y descenso de la calidad.

En este capítulo se estudia como las pérdidas de *slices* afectarían la calidad cuando no son aisladas; por lo cual se busca desarrollar una política de pérdidas que minimice la afectación cuando se requiera perder cierta cantidad de *slices* o reducir el *bit rate* del video en algún porcentaje.

5.1. Consideraciones para un Patrón de Pérdidas

Antes de pensar en estructurar un patrón de pérdidas es vital realizar algunas consideraciones previas para mejorar el desempeño. En el capítulo anterior se evaluó la importancia de cada una de las partes que forman un video, en esta sección se partirá de las consideraciones del capítulo anterior. Se continuara con la premisa de que la prioridad de un *slice* no depende de su tipo, sino solamente del valor de CMSE atribuido a su pérdida cuando se mide la calidad posterior a la decodificación.

Una herramienta del estándar H.264 diseñada para mejorar el desempeño ante errores es la ordenación flexible de MB, durante la codificación de las secuencias de video se eligió la opción de ordenación en modo disperso, la cual permite con una separación de los MB en grupos de *slices* SG en forma de una cuadricula de ajedrez, donde dos MB adyacentes siempre pertenecen a grupos

distintos. La principal desventaja de esta separación es el decrecimiento de la eficiencia de compresión; ya que pierde su fuerza el proceso de predicción porque depende de la redundancia espacial. Por otro lado, esta forma de codificación hace que cada *slice* sea una unidad de decodificación independiente y que si se llega a perder solo un slice de alguna imagen, se tendrá información espacial de los MB vecinos para la reconstrucción del área faltante, por medio de las herramientas de cancelamiento de error del estándar.

En la tabla 4.4 se presento en forma de lista, un ejemplo de los *slices* que conforman el video con sus características, sin embargo, no se encuentra ninguna información asociada a que SG pertenece cada *slice*. Para generar un patrón de pérdidas es fundamental considerar este factor previamente. Con el fin de conocer a que grupo pertenecen, se analizaron los *slices* para anexar la información a la tabla del *tracefile* de salida.

Número de Slice	Timestamp	Tamaño en Bytes	Tipo de Frame	Frame	CMSE	SG	Prioridad
2	0.033333	393	5 7	1	249.23	0	0
3	0.033333	396	5 7	1	344.85	0	0
4	0.033333	380	5 7	1	436.64	0	0
5	0.033333	387	5 7	1	795.34	0	0
6	0.033333	391	5 7	1	868.26	0	0
7	0.033333	401	5 7	1	1204.40	0	0
8	0.033333	388	5 7	1	759.08	0	0
9	0.033333	113	5 7	1	605.27	0	0
10	0.033333	399	5 7	1	172.87	1	1
11	0.033333	403	5 7	1	281.16	1	0
12	0.033333	387	5 7	1	544.80	1	0
13	0.033333	404	5 7	1	543.51	1	0
14	0.033333	394	5 7	1	1194.13	1	0
15	0.033333	401	5 7	1	767.78	1	0
16	0.033333	398	5 7	1	1430.08	1	0
17	0.033333	50	5 7	1	411.00	1	0
18	0.066667	45	1 5	2	86.96	0	1
19	0.066667	33	1 5	2	79.84	1	1
20	0.100000	12	1 6	3	0.64	0	3
21	0.100000	12	1 6	3	0.60	1	3
22	0.133333	147	1 5	4	124.31	0	1
23	0.133333	115	1 5	4	65.82	1	3
24	0.166667	18	1 6	5	0.76	0	3
25	0.166667	20	1 6	5	0.42	1	3

Tabla 5.1: Tracefile de Salida en H.264 modificada

En la tabla 5.1 se muestra una tabla similar a la anterior pero con nuevas columnas de información anexada a cada *slice*, se incorporan los valores de

CMSE, a que grupo pertenece cada *slice* y la prioridad asignada. Con estos datos se tienen suficientes bases para elaborar una política de pérdidas para secuencias de video H.264.

Para ilustrar la división de *slices* en grupos se presenta la figura 5.1. Aquí se muestra un cuadro de imagen de la secuencia Foreman, en formato CIF con resolución es de 352×288 píxeles. Cada MB dentro del estándar tiene un tamaño de 16×16. En la figura estos bloques de píxeles son delimitados con cuadros de colores, todos los bloques que estén enmarcados por el mismo color pertenecen al mismo *slice*. Aunque se observa la presencia de varios *slices*, solo existen dos grupos. Como ya vimos la aparición de los SG es alternada.

Figura 5.1: División en *slices* de Foreman cuadro 15

No es fácilmente reconocible el inicio y termino del *slice* debido a que en cada vecindad de dos MB están presentes dos colores de SG distintos. Se observa la alternancia de colores en MB consecutivos. La asignación de MB a los grupos siempre inicia desde la esquina superior izquierda, desplazándose hacia la derecha y posteriormente hacia abajo para alcanzar finalmente la esquina inferior derecha de la imagen. El número de MB que estén contenidos en cada *slice* depende de la cantidad de información asociada a los mismos. En zonas con textura suave probablemente los *slices* contendrán mayor número de MB. Cabe añadir que el número de *slices* en cada imagen depende mucho del tipo de codificación que se use: intra o inter. Una imagen con codificación intra en general posee muchos más *slices* que una imagen en codificación inter. Las imágenes

IDR contienen la mayor cantidad de *slices*, seguido de las P y finalmente las B, que son imágenes con el mayor grado de compresión.

Ya conocida la distribución de los MB en grupos de *slices*, otro factor a tomar en cuenta es que la pérdida de dos *slices* de grupos distintos pero espacialmente adyacentes generaría pérdidas de secciones completas de imagen que aun con el empleo de técnicas de cancelación de error no sería posible reconstruirlas de manera aceptable. Por lo que una regla que se debe tener en mente durante la concepción de un patrón de pérdidas es la siguiente:

Habiendo descartado un slice sin importar tipo y prioridad no se debe descartar el slice espacialmente adyacente con el propósito de que la reconstrucción sea eficiente.

Es posible verificar a partir de la tabla 5.1 cuando dos *slices* son adyacentes o no. Como ejemplo, consideremos el caso del cuadro de imagen número uno, está formado en total por 16 *slices*. Los primeros ocho *slices* pertenecen al $SG \rightarrow 0$ y los ocho restantes al $SG \rightarrow 1$. El primer *slice* del grupo cero es adyacente con el primer *slice* del grupo uno, el segundo con el segundo y así sucesivamente. Por lo que el *slice* con número 2 es adyacente con el *slice* número 10, el 3 con el 11, etc.

En las tablas de distribución de bits por tipo de imagen y prioridad del capítulo anterior se realizó la división de los *slices* que pertenecen a imágenes tipo P entre la primera y segunda mitad del GOP, de nueva cuenta se utilizará esta consideración para el patrón de pérdidas porque como veremos más adelante, las pérdidas en una sección u otra impactan con diferente magnitud. Las pérdidas de *slices* P generadas en la primera mitad del GOP impactaran con mayor magnitud que los *slices* P de la segunda mitad.

5.2. Formulación de Política de Pérdidas

Una política de pérdidas indica el orden en que los *slices* de un video deben seleccionarse para ser descartados con el fin de minimizar al máximo el decaimiento de la calidad. La estrategia que sigue una política es clasificar los *slices* de acuerdo a tipo, al SG, nivel de prioridad y posición dentro del GOP y posteriormente se ordenan con base en estas características. Los *slices* que sean más factibles de descartar serán considerados al inicio de la política de pérdidas. En cambio los *slices* que tengan alto impacto en la calidad no serán tomados en cuenta en lo absoluto.

Los criterios en los que basa la propuesta de políticas de pérdidas ya se han expuesto previamente de forma implícita. Una de los principales criterios son las dependencia de codificación, por lo que se puede establecer un orden de pérdidas jerárquico en base al tipo de imagen. Cuando se genera una pérdida, las dependencias de codificación implican propagación de error, la magnitud de esta propagación esta determinada por el tipo de imagen donde ocurre y la

posición que ocupa la imagen dentro del GOP. Otro criterio fundamental son las herramientas de recuperación al error. Estas funcionan mucho mejor cuando se pierden secciones parciales de imagen, de ahí surge la propuesta de evaluar una condición de adyacencia y dispersar las pérdidas en una imagen y a lo largo del GOP.

POLÍTICA A

1 *Slices* de prioridad 3 del primer grupo (SG0) en imágenes B. **B_P3SG0**

2 *Slices* de prioridad 3 del primer grupo (SG0) en imágenes P de la segunda mitad del GOP.**P_2hP3SG0**

3 *Slices* de prioridad 3 del primer grupo (SG0) en imágenes P de la primera mitad del GOP. **P_1hP3SG0**

4 *Slices* de prioridad 3 del primer grupo (SG0) en imágenes IDR. **I_P3SG0**

5 *Slices* de prioridad 3 del segundo grupo (SG1) que no sean adyacentes en imágenes B. **B_P3SG1**

6 *Slices* de prioridad 3 del segundo grupo (SG1) que no sean adyacentes y estén en la segunda mitad del GOP en imágenes P. **P_2hP3SG1**

7 *Slices* de prioridad 3 del segundo grupo (SG1) que no sean adyacentes y estén en la primera mitad del GOP en imágenes P. **P_1hP3SG1**

8 *Slices* de prioridad 3 del segundo grupo (SG1) que no sean adyacentes en imágenes IDR. **I_P3SG1**

9 *Slices* de prioridad 2 del primer grupo (SG0) en imágenes B. **B_P2SG0**

10 *Slices* de prioridad 2 del segundo grupo (SG1) que no sean adyacentes en imágenes B. **B_P2SG1**

11 *Slices* de prioridad 2 del primer grupo (SG1) en imágenes P. **P_P2SG0**

12 *Slices* de prioridad 2 del primer grupo (SG0) en imágenes IDR. **I_P2SG0**

La política de pérdidas especifica la forma u orden de descartar *slices* pero no hace alusión a la cantidad de *slices* que hay en cada paso de la misma. El número de *slices* en cada paso depende de cada secuencia en particular a la

que se aplique. La cantidad de *slices* a descartar depende directamente de la aplicación que hace uso de esta herramienta.

POLÍTICA B

1 *Slices* de prioridad 3 del primer grupo (SG0) en imágenes B.

2 *Slices* de prioridad 3 del segundo grupo (SG1) que no sean adyacentes en imágenes B.

3 *Slices* de prioridad 3 del primer grupo (SG0) en imágenes P de la segunda mitad del GOP.

4 *Slices* de prioridad 3 del segundo grupo (SG1) que no sean adyacentes y estén en la segunda mitad del GOP en imágenes P.

5 *Slices* de prioridad 3 del primer grupo (SG0) en imágenes P de la primera mitad del GOP.

6 *Slices* de prioridad 3 del segundo grupo (SG1) que no sean adyacentes y estén en la primera mitad del GOP en imágenes P.

7 *Slices* de prioridad 3 del primer grupo (SG0) en imágenes IDR.

8 *Slices* de prioridad 3 del segundo grupo (SG1) que no sean adyacentes en imágenes IDR.

9 *Slices* de prioridad 2 del primer grupo (SG0) en imágenes B.

10 *Slices* de prioridad 2 del segundo grupo (SG1) que no sean adyacentes en imágenes B.

11 *Slices* de prioridad 2 del primer grupo (SG0) y estén en la segunda mitad del GOP en imágenes P.

12 *Slices* de prioridad 2 del segundo grupo (SG1) que no sean adyacentes y estén la segunda mitad del GOP en imágenes P.

Como ejemplo, en un nodo de red como un *router*, que experimenta congestión y aumentos en los retardos de transmisión de paquetes de video, busca tomar medidas que alivien la situación. Una medida usada muy comúnmente es eliminar paquetes de su *buffer* cuando está lleno. Si los paquetes que se eliminan no se diferencian unos de otros y corresponden a la parte final del *buffer*, a este

método se le como conoce como *Drop Tail*. Una política de pérdidas dictaría el mejor criterio para discriminar que *slices* eliminar y en que orden. La cantidad de *slices* a eliminar depende directamente del *router*, este decide cuando es suficiente para dejar de descartar de acuerdo a su percepción de la situación.

Los *slices* de más baja importancia en términos de CMSE (prioridad 3) son los primeros que debe ser descartados, posteriormente los de prioridad 2, etc. Pero esta aseveración debe ser combinada con otras características para que se obtenga una esquema más robusto. Se ha considerado tipo de imagen y SG. De esta manera se puede enunciar que tipo de *slices* deben aparecer en primer lugar de la política de pérdidas: *slices* prioridad 3, que aparezcan en imágenes B y solo pertenezcan al SG0. Posteriormente de imágenes P y finalmente de imágenes IDR. La descripción más especifica y ordenada se muestra en los cuadros siguientes con las políticas A y B.

En el cuadro de la Política A se usan solo *slices* de prioridad 3, los ocho pasos por los que está compuesta, especifican combinaciones de características de los *slices* que han sido consideradas. El orden asumido con respecto al tipo de imagen (IDR, P y B) se debe a las dependencias de codificación, las tipo B no tienen ninguna dependencia por lo que aparecen al principio, le siguen las tipo P, donde las imágenes de la segunda mitad del GOP tienen dependencias menos fuertes que las de la primera parte y finalmente las tipo IDR que poseen las mayores dependencias. En los cuatro primeros pasos solo se usan *slices* del $SG \rightarrow 0$ y en los cuatro siguientes del $SG \rightarrow 1$ pero que cumplan con la condición de no ser adyacentes de otro slice previamente perdido.

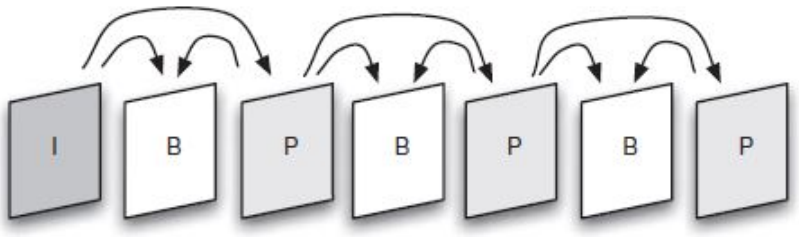


Figura 5.2: Estructura del GOP

La política B es una versión con diferente ordenación que la política A, se propone para evaluar un orden distinto. La principal diferencia entre las políticas es que en la A cuando se pierden *slices* del SG0 de un tipo de imagen y ciertas características (por ejemplo tipo B, prioridad 3) inmediatamente se pasa a *slices* de un tipo diferente de imagen (tipo P, prioridad 3) mientras que en la política B se continua con *slices* del mismo tipo (tipo B) pero del $SG \rightarrow 1$.

La filosofía de las políticas es dispersar las pérdidas a lo largo de todo el GOP, no concentrarlas. De forma que degradación de la calidad a lo largo de toda la secuencia tienda a ser uniforme. La concentración de pérdidas en una

sección del video provoca una peor experiencia para el usuario final ya que su atención se centrara principalmente en estas secciones corrompidas y aunque haya secciones con buena calidad muy probablemente no mejorara la percepción general del video.

Siempre es preferible empezar a descartar *slices* iniciando desde el final del GOP hasta llegar al inicio del mismo. El fundamento radica en las dependencias de codificación, estas serán más acentuadas en los *slices* que aparezcan más temprano dentro del GOP y disminuirá conforme se aproximen al final del GOP como se observa en la figura 5.2. Esta aseveración es valida para *slices* P, por lo cual se realizó la división del GOP en dos secciones y se incluyó esta división en las políticas. Para *slices* tipo IDR las dependencias son las más acentuadas y además mayores en comparación con las de tipo P, así las pérdidas de *slices* de baja prioridad que se encuentren en imágenes tipo IDR se ubicaron hasta la parte final en los pasos de la política.

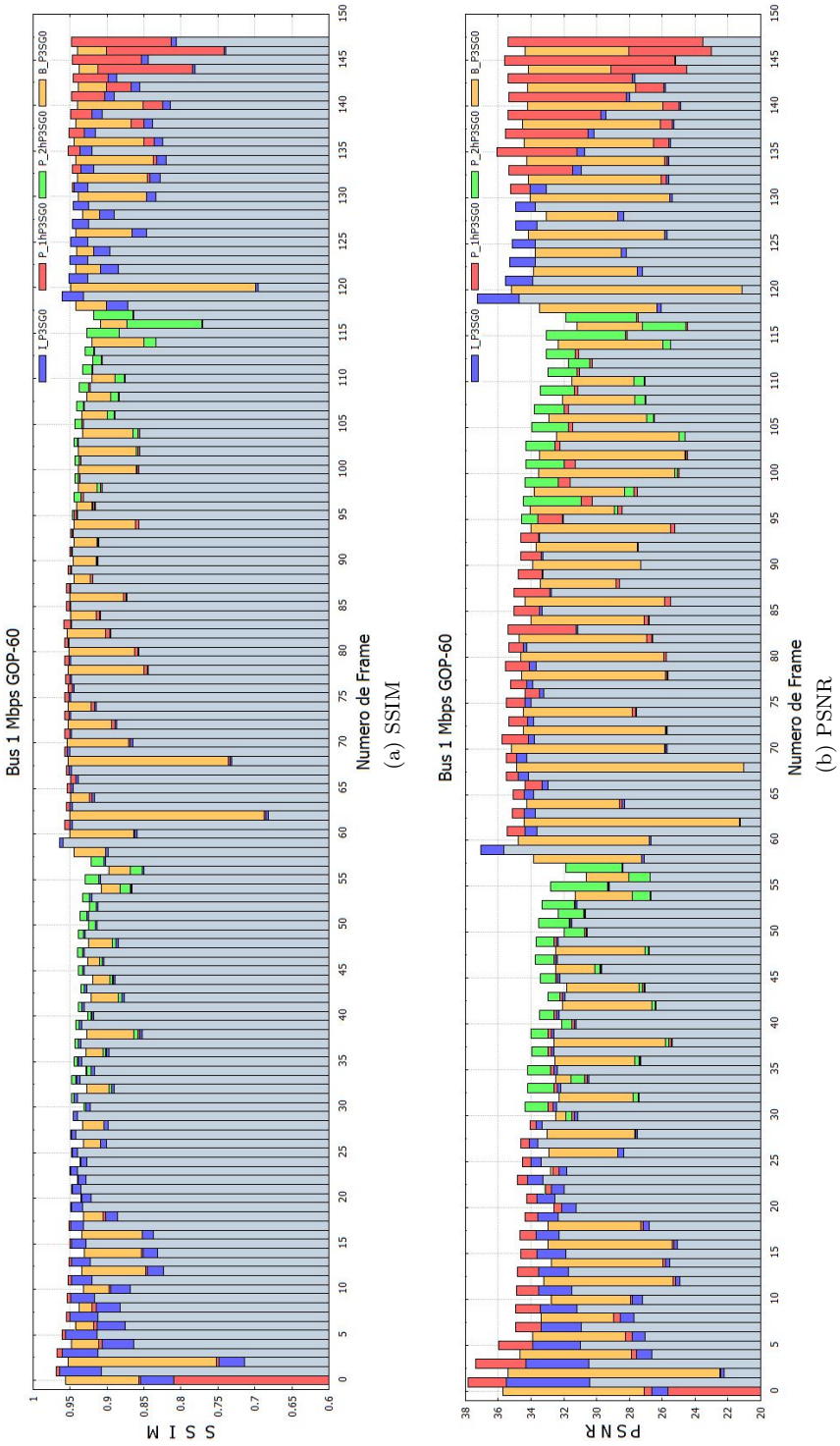
5.3. Resultados para la secuencia Bus

A continuación se presentaran algunos de los resultados de la aplicación de las políticas a las secuencias de video. Los resultados en forma de gráficas y tablas son abundantes por lo que se ha decido solo colocar algunos más representativos. En el tema anterior se expuso un par de patrones de pérdidas de *slices* para las secuencias, para la aplicación de los patrones se clasificaron los *slices* rápidamente por sus características. El orden de pérdidas de *slices* está claramente especificado en los cuadros de las políticas A y B.

Las gráficas 5.3-5.5 representan la evolución de la calidad de cada cuadro de imagen de la secuencia Bus durante la aplicación de todos los pasos de la política A, los doce pasos se encuentran divididos en tres gráficas.

En la figura 5.3 se muestran los cuatro primeros pasos, la calidad se mide en términos del PSNR y SSIM, ya que son las métricas que permiten obtener un valor de calidad por cuadro de imagen. Como se mencionó en tema de métricas de calidad el VQM solo permite obtener un solo valor de medición por cada secuencia de video completa, esta métrica se usará como parámetro de referencia en la comparación del desempeño de las políticas. La forma en que están diseñadas las gráficas hace posible identificar con facilidad la magnitud del decremento el valor resultante de la calidad en cada paso así como saber si un cuadro sufrió o no afectación en cada eliminación.

Cada secuencia codificada cuenta con un nivel de calidad de salida que queda determinado por los parámetros de configuración que se eligieron. Para los resultados que aparecen en esta sección es importante conocer la calidad del video antes de realizar cualquier prueba porque brinda una referencia como punto de partida de donde se medirá el decaimiento de la calidad.



(a) SSIM

(b) PSNR

Figura 5.3: Medición de la calidad, Política A pasos 1-4, Secuencia Bus a 1 Mbps GOP60.

En la figura 5.3 el eje horizontal simboliza los cuadros de imagen que componen la totalidad de la secuencia y el eje vertical es la escala de la métrica de calidad: SSIM o PSNR. Para el SSIM la escala completa abarca un rango de -1 a 1 pero la mayoría de las mediciones hechas oscilan en el rango de 0.5 a 1 por lo cual el eje vertical se restringe a un intervalo de proporciones similares o menores. Para el PSNR no existe una rango de valores definidos que pueda tomar, sin embargo, los resultados oscilan mayoritariamente entre 18 y 42 decibelios.

Paso de la Política	Bus 1Mbps GOP-10	Bus 1Mbps GOP-60	Bus 512Kbps GOP-10	Bus 512Kbps GOP-60
1.- B_P3SG0	115	163	38	77
2.- P_2hP3SG0	49	18	36	21
3.- P_1hP3SG0	21	39	13	31
4.- I_P3SG0	43	15	42	2
5.- B_P3SG1	12	14	8	11
6.- P_2hP3SG1	18	15	7	7
7.- P_1hP3SG1	21	19	11	13
8.- I_P3SG0	20	8	13	1
9.- B_P2SG0	44	61	30	46
10.- B_P2SG1	18	17	11	9
11.- P_P2SG0	114	161	36	68
12.- I_P2SG0	84	22	54	19

Tabla 5.2: Cantidad de *slices* perdidos en cada paso de la Política A para Bus

Cada cuadro de video es representado por medio de una barra vertical con altura variable. La altura representa la calidad medida, el valor máximo pico es la calidad que posee la imagen decodificada inicialmente. Entre mayor sea la altura mejor sera la calidad, análogamente entre mayor sea la pérdida de calidad sera mayor el descenso de su altura. Además cada barra puede contener o no secciones de distinto color. Cada color esta asociado a la aplicación de un paso de la política de pérdidas, la parte superior de la barra es la calidad del cuadro antes de su aplicación y la parte inferior es la calidad final después del un paso de la política. En muchas ocasiones varios cuadros no son afectados por un paso de la política por lo que no aparece el color asociado con ese paso dentro de la barra que lo representa.

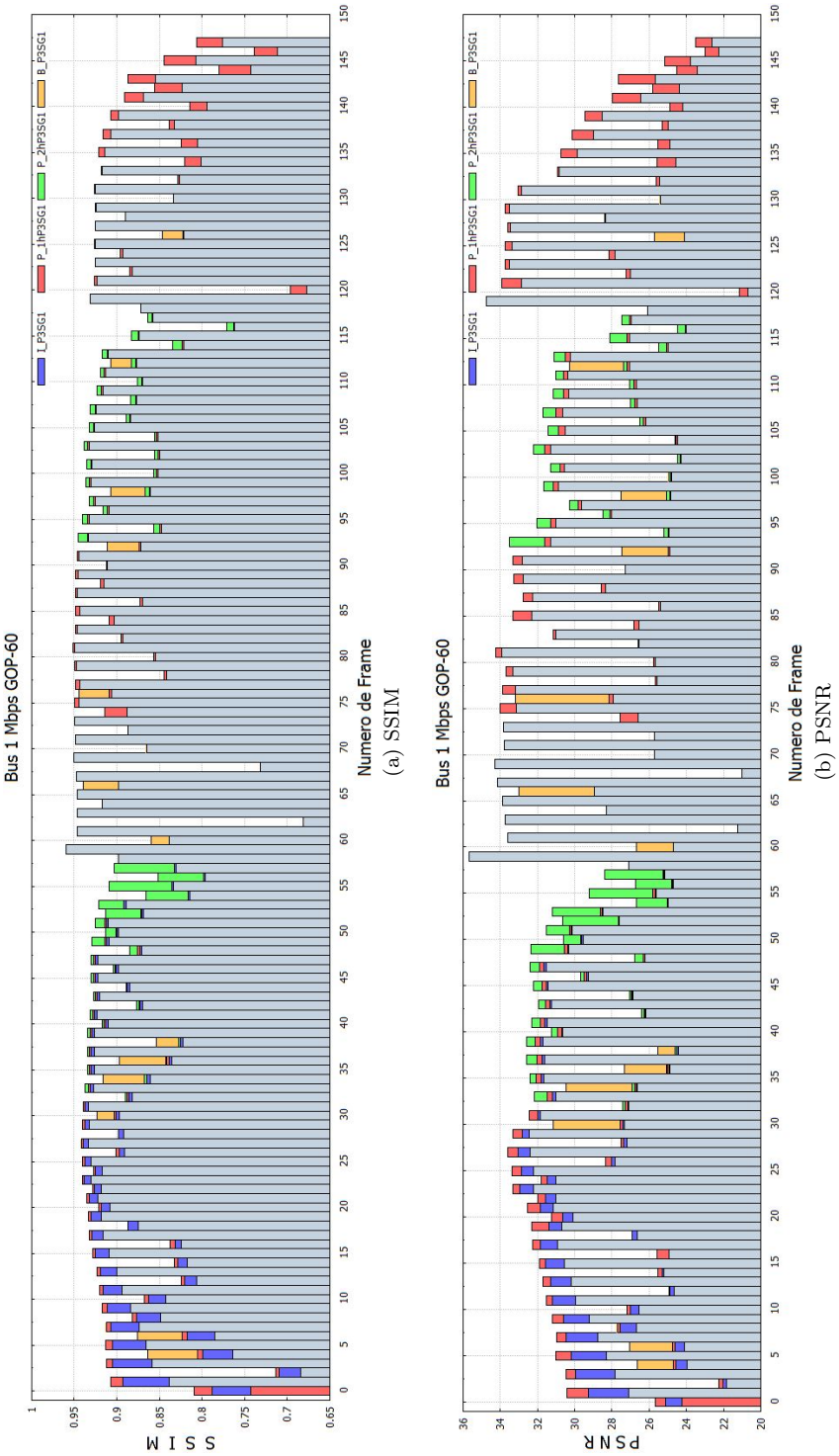
La identificación del color se realiza por medio de la tabla Política A. En cada paso de esa tabla, al final del enunciado viene un código que indica le tipo de *slices* que se debe tirar. Por ejemplo, **B_P3SG0** hace referencia a *slices* de imágenes B, prioridad tres y que pertenezcan al SG0, **P_1hP3SG1** hace

referencia a *slices* de imágenes P, que estén en la primera mitad del GOP, prioridad tres y que pertenezcan al SG1.

Una primera observación que se puede realizar de la figura 5.3 es que los valores picos de las barras dibujan una curva que surge con el inicio del GOP y decae conforme llega el final del GOP; cuando esto sucede se empieza a dibujar una nueva curva que abarca el siguiente GOP. Se distinguen dos curvas y una tercera con la mitad de longitud de las anteriores, justamente la secuencia es de 150 cuadros y el GOP de 60 cuadros por lo que cada curva 60 cuadros. Este fenómeno se debe a que la codificación en el estándar H.264 se realiza con pérdida de información, así las relativamente pequeñas pérdidas de información al inicio de GOP se van propagando e incrementando hasta el fin del GOP por las dependencias de codificación.

El primer paso de la política se aplica sobre imágenes B, como se ve existe una alternancia en la aparición de las barras color amarillo, el orden de aparición de los tipos de imagen se definió en la codificación y lo muestra la estructura del GOP de la figura 5.2. Las imágenes tipo B ocupan números de secuencia pares, los números de secuencia impares del GOP son ocupados imágenes tipo P a excepción de la primera posición de cada GOP que es usada por imágenes tipo IDR. En la secuencia Bus se eliminan para el primer paso de la política 163 *slices*, es notorio el descenso de la altura de muchas barras pero bajo esta situación las herramientas de corrección de error funcionan bien ya que disponen de toda la información de los cuadros de referencia tipo P y tipo IDR.

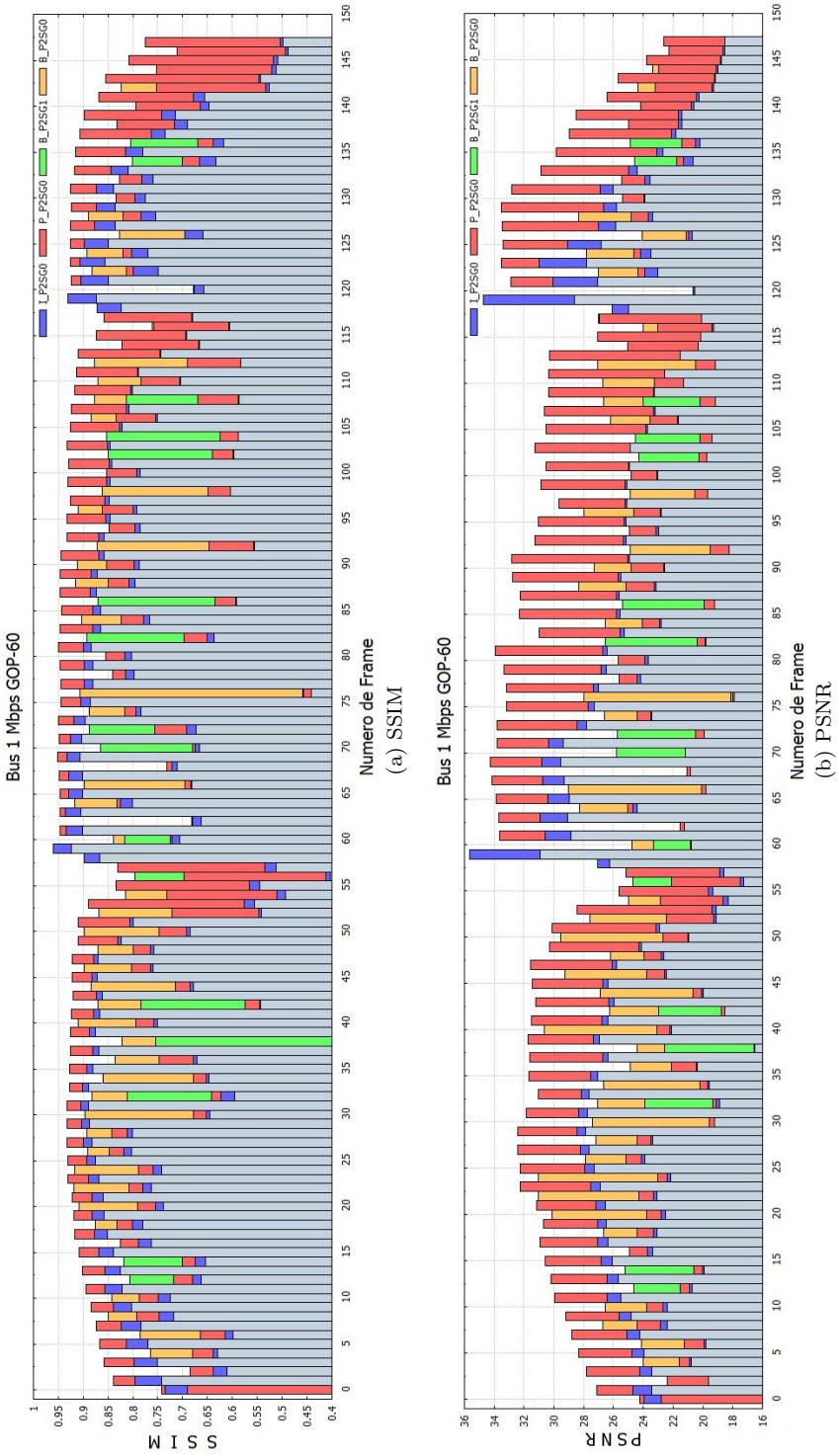
En el segundo paso, donde se pierden *slices* tipo P, prioridad 3 y de la segunda mitad del GOP el decaimiento de la calidad es muy leve, el color verde no aparece con frecuencia en las barras. Esto se debe a que son muy pocos los *slices* que cumplen con estas características, específicamente fueron eliminados en este paso solo 18 *slices*, 7 *slices* en el primer GOP y en el segundo 11. En el caso del tercer paso de la política se eliminan 39 *slices*. La gran mayoría de estos *slices* se encuentran al final del tercer GOP porque este GOP es más corto con tan solo 30 imágenes y por ende de acuerdo a la manera en que se mide la prioridad de los *slices* , esta sección presento menores valores del CMSE ya que la propagación del error se decremento sustancialmente.



(a) SSIM

(b) PSNR

Figura 5.4: Medición de la calidad, Política A pasos 5-8, Secuencia Bus a 1 Mbps GOP60.



(a) SSIM

(b) PSNR

Figura 5.5: Medición de la calidad, Política A pasos 9-12, Secuencia Bus a 1 Mbps GOP60.

Las gráficas de la aplicación de los pasos de las políticas se muestran a pares, en ellas la única diferencia es la métrica con la que se mide la calidad. Para el SSIM cuando se eliminan *slices* las variaciones de los valores de la métrica para cada cuadro de video son mínimas, en muchos casos a penas es perceptible el cambio en el valor de la calidad. Por esta razón en el capítulo anterior durante el establecimiento del esquema de pérdidas de *slices* no se eligió al SSIM como método para medir la calidad y así asignar las prioridades de cada slice. En cambio se presentan variaciones de más grandes proporciones para el PSNR. En general las tendencias de como disminuye la calidad es muy similar para ambas gráficas de métricas. Se distinguen las mismas curvas de variaciones pero para el PSNR es más acentuado obviamente.

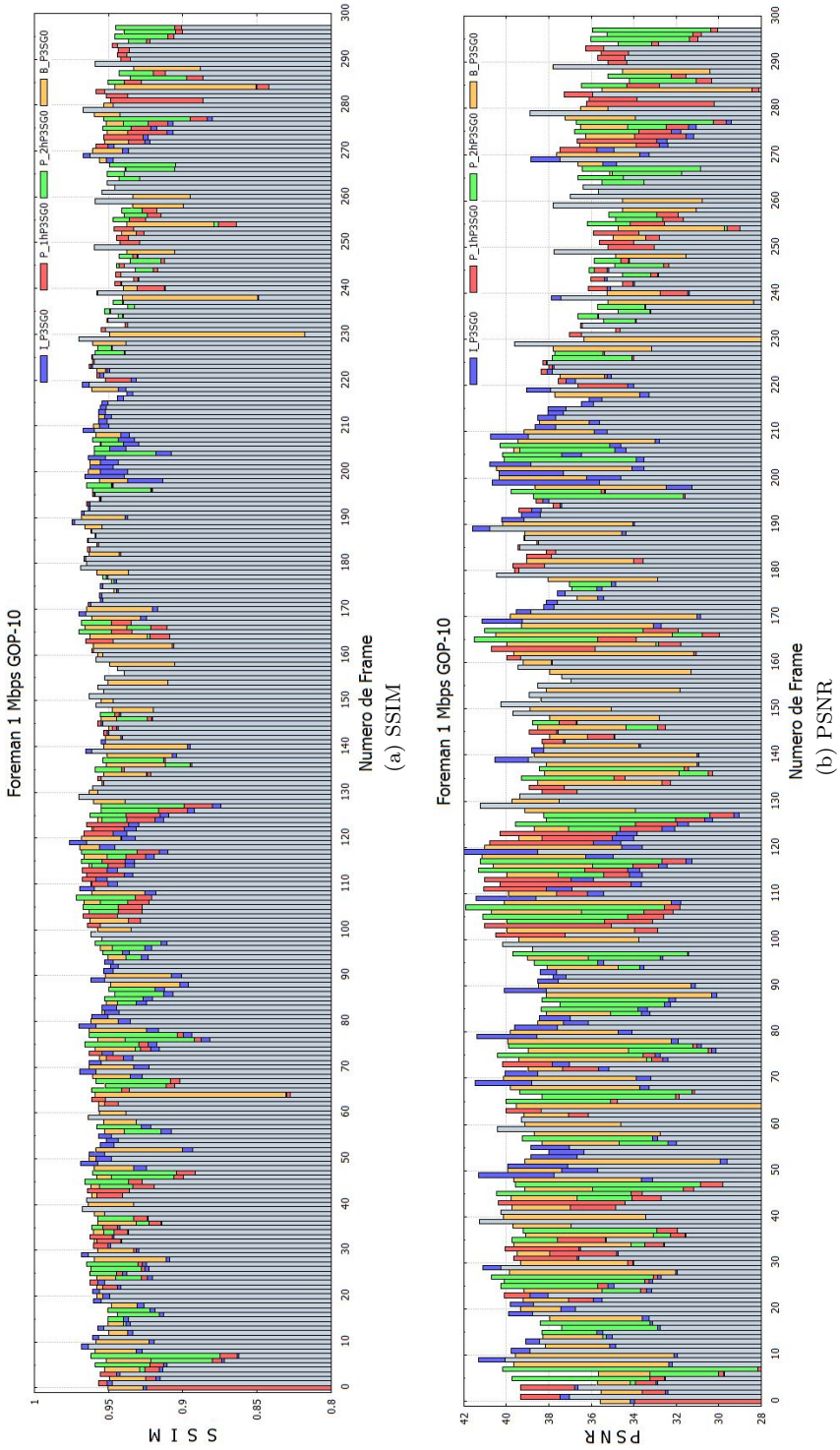
5.4. Resultados para la secuencia Foreman

Las gráficas para la secuencia Foreman toman las mismas características generales que las expuestas para la secuencia Bus: los pasos de las políticas siguen la misma terminología, cada imagen es representada por una barra vertical, etc. La secuencia Foreman cuenta con 300 cuadros de video, por ende la barras tienen un ancho más reducido pero de igual forma se distinguen claramente los cambios introducidos por la política. Una distinción es el tamaño del GOP, que ahora es de longitud 10, la propagación del error es mucho más corta.

Este tamaño de GOP implica que cada nueve imágenes P y B aparezca una imagen IDR que sirve como referencia para todo el GOP. Por la forma de codificación intra de las imágenes IDR ayudan a que al generarse alguna pérdida, esta no tendrá una efecto más allá del termino del GOP de donde surgió. La desventaja de este tipo de imágenes es que son mucho más amplias en bytes que las demás, por lo que para decodificar una imagen de este tipo en un receptor, se introduce mayor retardo ya que está compuesta de mayor número de paquetes que el resto.

El descenso de la calidad por el primer paso de la política (B_P3SG0) se ve bastante marcado en varios cuadros porque la cantidad de *slices* representa un porcentaje importante del total, se observa en muchos casos una caída de 4 a 8 dB para el PSNR. Esta caída es posible considerarla como relativamente grande. Pero valorando que la calidad del video codificado original es muy buena ya que muchos cuadros inicialmente tienen valores de calidad en PSNR de alrededor de 40 dB o más, una caída en la calidad a valores de 32 a 36 dB es bastante aceptable.

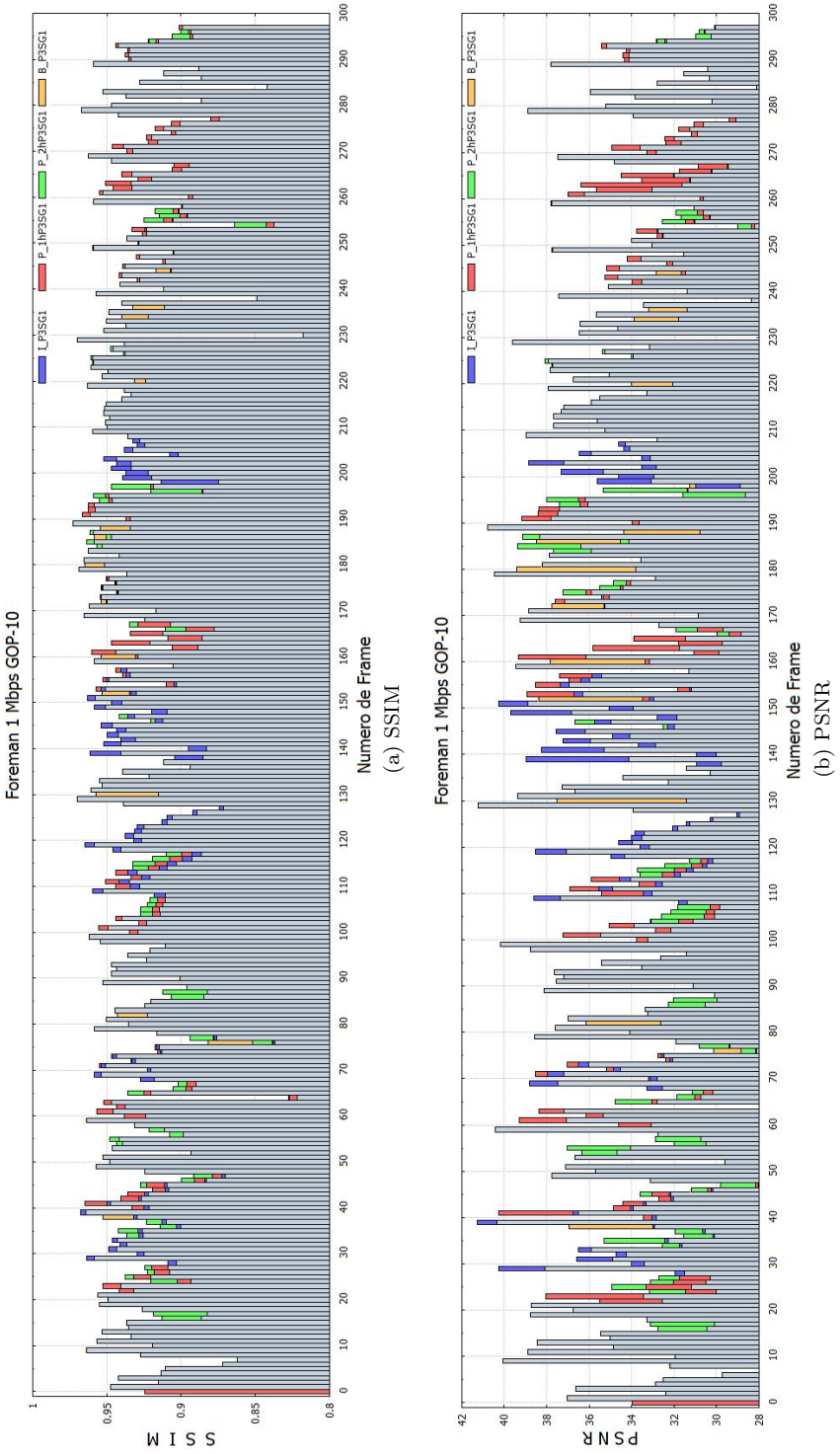
Para el segundo paso de la política existen intensas afectaciones, 168 es el número de *slices* que conforman este paso. Se da la propagación del error a imágenes tipo B posteriores a las posiciones de imágenes P donde se eliminaron *slices*. La lógica hace pensar que la cantidad de *slices* en el tercer paso de la política sera menor que los del segundo paso ya que la propagación del error es más marcada.



(a) SSIM

(b) PSNR

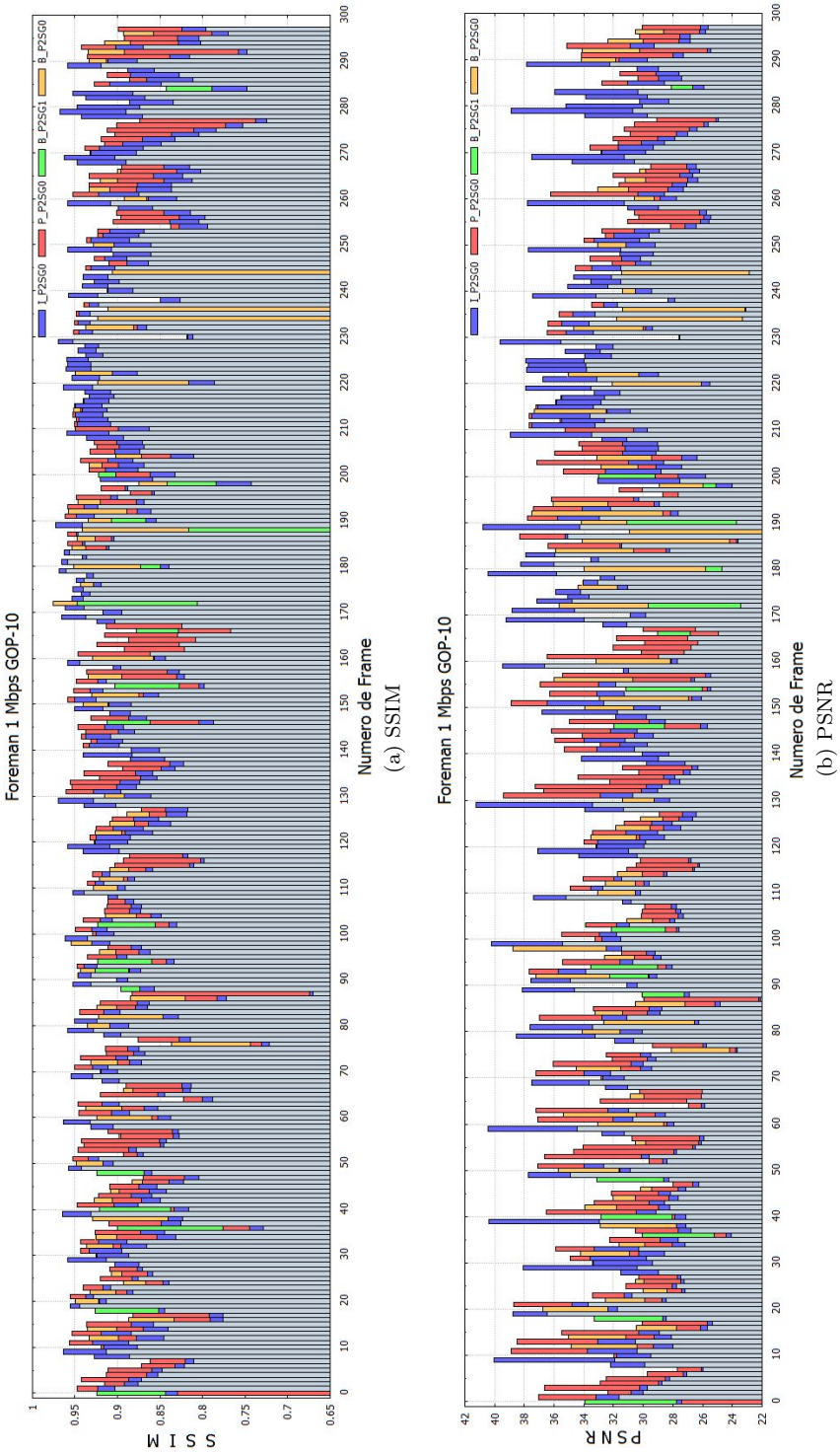
Figura 5.6: Medición de la calidad, Política A pasos 1-4, Secuencia Foreman a 1 Mbps GOP10.



(a) SSIM

(b) PSNR

Figura 5.7: Medición de la calidad, Política A pasos 5-8, Secuencia Foreman a 1 Mbps GOP10.



(a) SSIM

(b) PSNR

Figura 5.8: Medición de la calidad, Política A pasos 9-12, Secuencia Foreman a 1 Mbps GOP10.

Paso de la Política	Foreman 1Mbps GOP-10	Foreman 1Mbps GOP-60	Foreman 512Kbps GOP-10	Foreman 512Kbps GOP-60
1.- B_P3SG0	194	295	97	147
2.- P_2hP3SG0	168	133	93	85
3.- P_1hP3SG0	69	29	33	7
4.- I_P3SG0	32	1	20	1
5.- B_P3SG1	19	19	9	8
6.- P_2hP3SG1	41	71	16	42
7.- P_1hP3SG1	34	17	17	13
8.- I_P3SG0	16	3	19	0
9.- B_P2SG0	74	69	33	28
10.- B_P2SG1	20	21	9	9
11.- P_P2SG0	190	369	81	225
12.- I_P2SG0	173	17	115	1

Tabla 5.3: Cantidad de *slices* perdidos en cada paso de la Política A para Foreman

Luego en el cuarto paso se eliminan 32 *slices* tipo IDR, las consecuencias de esto no son de mucha consideración, son decrecimientos leves pero que están presentes en las imágenes que aparecen más temprano durante la longitud del GOP.

Los siguientes cuatro pasos de la política los podemos aprecia en la figura 5.7, del cinco al ocho, son muy parecidos a los primeros cuatro con la diferencia que ahora se pierden el mismo tipo de *slices* pero pertenecientes al SG1 y que cumplan con la condición de no ser adyacentes. Esta condición manifiesta que no se permite perder *slices* que posean macrobloques vecinos en el dominio espacial a los *slices* previamente ya considerados en los primeros cuatro pasos, con el propósito de que las herramientas de cancelamiento de error se desempeñen mejor.

La cantidad de *slices*, sin considerar la condición de ser no adyacentes, es abundante pero cuando se considera esta condición desciende fuertemente el número, generalmente son pocos los *slices* que cumple con todas las condiciones establecidas en estos pasos. Para la secuencia de la gráfica de la que se habla en este momento tiene solo 19 *slices* en el paso 5, 41 en el paso 6, 34 en el paso 7 y solo 16 en el paso 8. Así mismo este bajo número de *slices* se ve reflejado en la grafica, ya que se aprecia muy baja presencia de los colores que representan la afectación.

5.5. Resultados para la secuencia Akiyo

Paso de la Política	Akiyo 256Kbps GOP-10	Akiyo 256Kbps GOP-60	Akiyo 128Kbps GOP-10	Akiyo 128Kbps GOP-60
1.- B_P3SG0	126	295	115	77
2.- P_2hP3SG0	20	133	4	12
3.- P_1hP3SG0	6	29	0	2
4.- I_P3SG0	1	1	0	0
5.- B_P3SG1	1	19	0	7
6.- P_2hP3SG1	4	71	5	15
7.- P_1hP3SG1	1	17	2	10
8.- I_P3SG0	3	3	0	0
9.- B_P2SG0	22	69	34	70
10.- B_P2SG1	2	21	3	3
11.- P_P2SG0	59	369	86	32
12.- I_P2SG0	71	17	1	0

Tabla 5.4: Cantidad de *slices* perdidos en cada paso de la Política A para Akiyo

Se puede observar en la tabla 5.4 la cantidad de *slices* que contienen las características de cada paso de la política A. En la primera fila de la tabla aparecen los *slices* del paso B_P3SG0 que representan un gran porcentaje del total de los mismos. Para el resto de los pasos la cantidad de *slices* es demasiado baja, inclusive en varias posiciones no existen *slices*. Cuando se observa la evolución de la política en los video de prueba de la secuencia Akiyo los resultados son planos y no existen cambios significativos.

Esta situación hace poco benéfico el mostrar gráficas de la evolución de la calidad de los cuadros de video. Se presenta la misma tendencia para las cuatro secuencias codificadas de Akiyo. Esto hace establecer la conclusión que la causa de este fenómeno no se atribuyen a tasa de codificación ni la longitud del GOP sino se debe a las características de bajo nivel de movimiento y fondo estático del video. La única sección donde existe movimiento suficientemente perceptible en la secuencia es la cara de la presentadora de noticias, el fondo de la imagen es estático durante todo el intervalo de duración.

5.6. Evaluación de Políticas

En esta sección se presenta la evaluación del desempeño de las políticas A y B sobre los videos de prueba seleccionados. Las políticas básicamente se refieren a la clasificación de los *slices* que conforman un video y a la ordenación para predefinir un orden especifico de selección que se utilice cuando se necesite reducir el *bit rate*. Uno de los métodos más utilizados de administración de un *buffer* en redes de computadoras es el *Drop Tail*. En este método no se establece diferencia alguna entre los paquetes que se desechan, que en este caso se considera contienen vídeo codificado. Se toma este método como referencia con el cual se comparan las políticas y así poder corroborar si existe y cual es la magnitud de la mejora en términos de calidad cuando se utilizan las políticas de pérdidas.

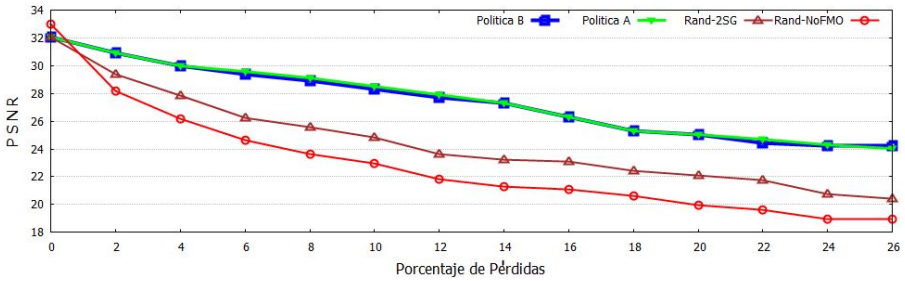
Debido a que en el método de *Drop Tail* no se diferencian paquetes, se puede asumir que los paquetes perdidos bajo este esquema son seleccionados totalmente de forma aleatoria. Para simular este esquema de pérdidas se hace uso de las secuencias de prueba consideradas previamente. El esquema de pérdidas que simula el *Drop Tail* se nombrara **Rand-2SG** con fines de identificación en gráficas posteriores.

La correcta configuración de las secuencias de video tiene un impacto significativo sobre la calidad en las secuencias ante pérdidas. Para corroborar esta aseveración se realizó la codificación de las secuencias de vídeo de prueba con la distinción de no incluir el uso de la herramienta *Flexible Macroblock Order*, por lo que no existirá más que un único Slice Group (SG). Se consideraran estos videos codificados para evaluarlos en la evolución de la calidad ante pérdidas. Con fines de identificación se nombraran **Rand-NoFMO**.

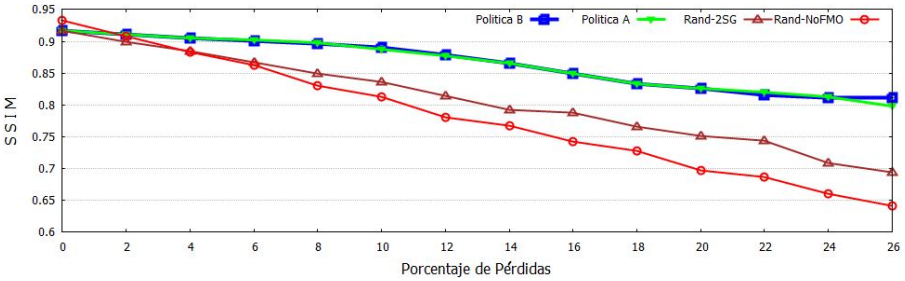
Las secuencias que se codifican sin FMO ahorran bits que se utilizaban para señalizar al decodificador el uso de la ordenación de macrobloques además de que aumenta la eficiencia de compresión al explotar de mejor forma la correlación espacial, por lo que el tamaño en bits de las secuencias disminuye en comparación con las secuencias consideradas anteriormente. En promedio los videos codificados tienen un tamaño menor en 15 % que su tamaño anterior.

A continuación se mostraran una serie de gráficas de la calidad de secuencias de video sujetas a pérdidas bajo los esquemas de la Política A y B, el esquema Rand-NoFMO y Rand-2SG. Las gráficas mostraran en el eje vertical escalas de métricas de calidad y en el horizontal las pérdidas del vídeo. Para representar las pérdidas existen dos posibles formas de proceder, una de ellas es colocar en el eje horizontal el número de *slices* perdidos en cada punto de la gráfica, la segunda es colocar el porcentaje que representa del tamaño total de la secuencia, los *slices* que se eliminan en cada punto de la gráfica.

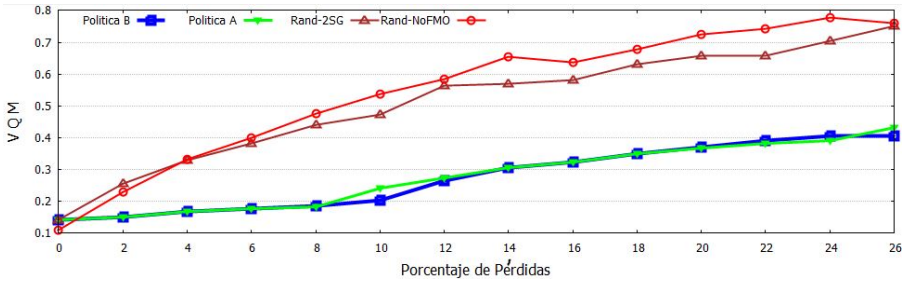
Los *slices* de todos los videos tienen un tamaño variable limitado a 400 bytes, si en el eje horizontal se colocan perdidas de acuerdo al número de *slices*, los puntos de evaluación se distribuirían cada cierto número de *slices*, así también cada esquema de perdidas implicara que se eliminen diferentes cantidades de bytes aunque correspondan a la misma cantidad de *slices* para cada punto de



(a) PSNR



(b) SSIM

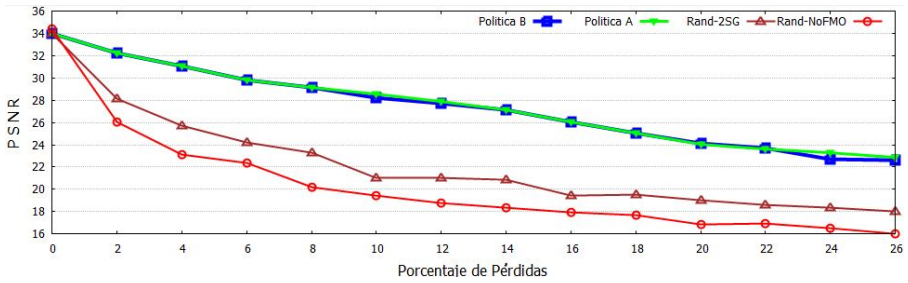


(c) VQM

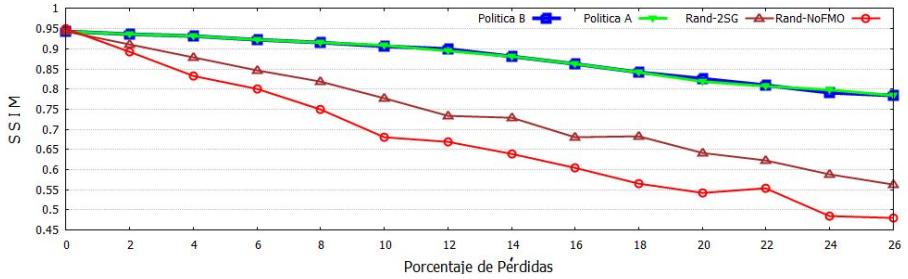
Figura 5.9: Comparación de Políticas Secuencia Bus 1 Mbps GOP-10

evaluación. Distinta cantidad de bytes corresponde diferente magnitud de perdida de información lo que representaría condiciones desiguales de evaluación que se verían reflejadas en la calidad del video. Por esta razón se opto por establecer en el eje horizontal la cantidad de perdidas en función del porcentaje de bits que representan los *slices* perdidos. Para tres de los cuatro esquemas de evaluación se usa la misma secuencia de video codificada y la ultima es la secuencia codificada sin FMO.

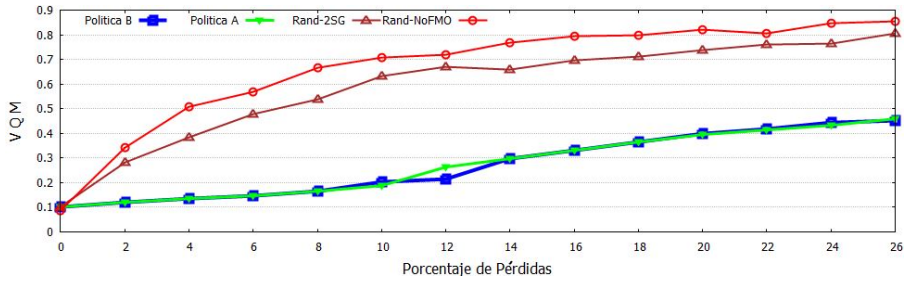
En estas gráficas de resultados el primer punto de evaluación de calidad siempre esta posicionado en un valor de cero en el eje horizontal. Este punto indica la calidad del video codificado sin ninguna pérdida y sera el punto de referencia a partir del cual la calidad sera la mejor y posteriormente ira deca-



(a) PSNR



(b) SSIM



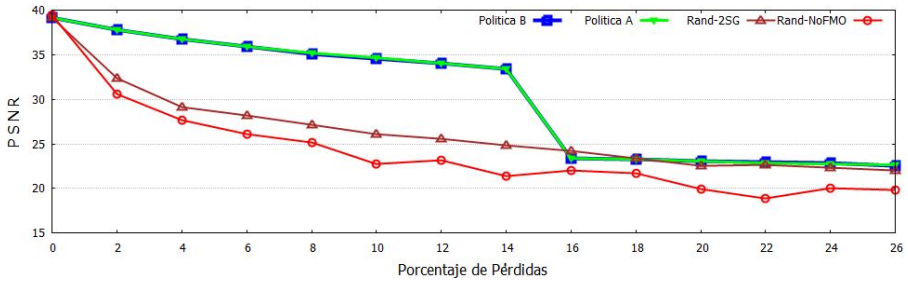
(c) VQM

Figura 5.10: Comparación de Políticas Secuencia Bus 1 Mbps GOP-60

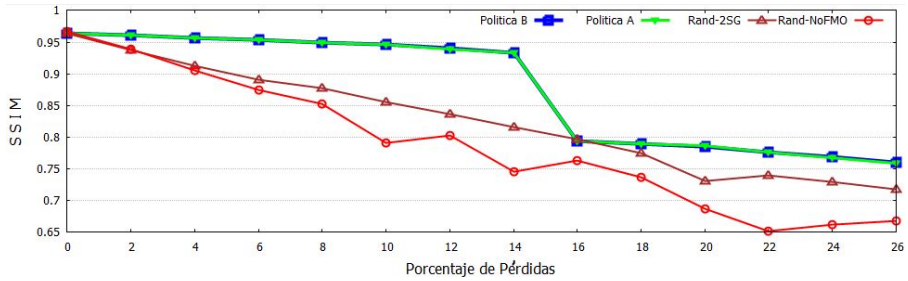
yendo conforme avance el porcentaje de pérdidas. El esquema Rand-NoFMO generalmente tiene una calidad ligeramente mejor que los demás esquemas, en PSNR este valor esta en el rango de 0,5 dB.

Las gráficas se muestran en grupos de tres, cada una contiene una métrica de calidad para tener más de una referencia de evaluación de los esquemas, ya que como se expuso, las métricas objetivas no poseen una confiabilidad total. Cada grupo de gráficas pertenece a una secuencia de video. Todas las gráficas tienen un rango de pérdidas desde cero hasta veintiséis por ciento. Los videos normalmente soportan este rango de perdidas salvo algunas excepciones.

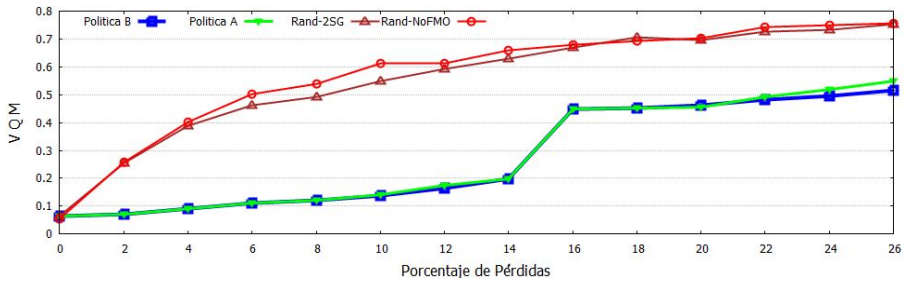
Las gráficas de la figura 5.9 corresponden a la secuencia Bus 1 Mbps GOP=10, cada esquema de pérdidas tiene una tendencia decreciente para las métricas



(a) PSNR



(b) SSIM

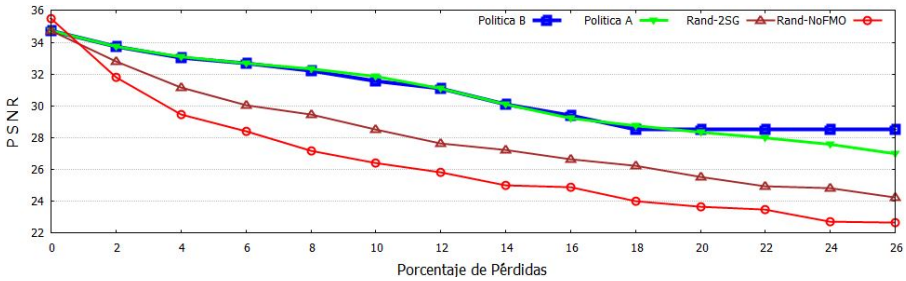


(c) VQM

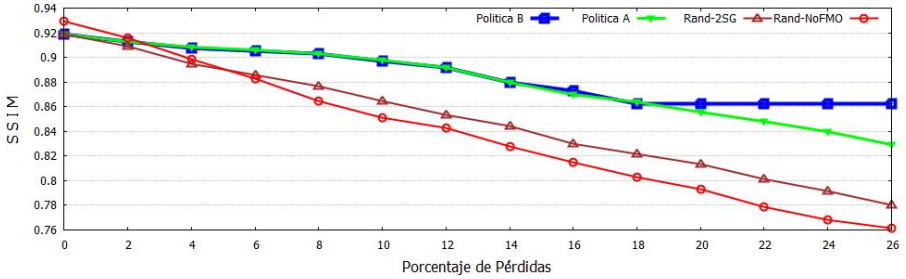
Figura 5.11: Comparación de Políticas Secuencia Foreman 1 Mbps GOP-60

PSNR y SSIM y creciente para el VQM. En el primer punto de evaluación, con una pérdida del 2 % los esquemas de la política A y B coinciden en resultados porque eliminan los mismo *slices*, esta similitud de valores ocurre en varios puntos de evaluación de esta figura pero también en las demás gráficas subsecuentes.

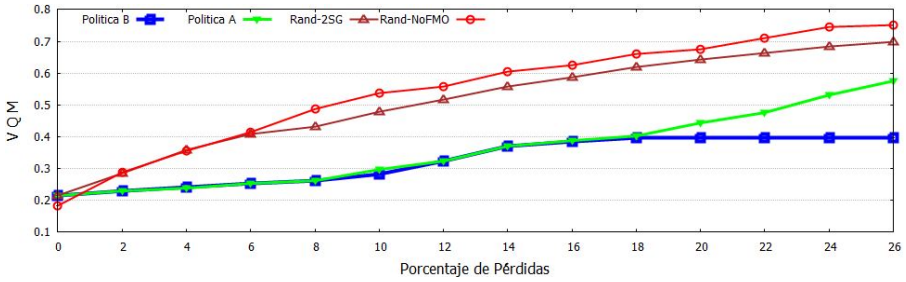
La figura 5.11 de la secuencia Foreman tiene una discontinuidad abrupta para un porcentaje de perdidas del 16 % en los esquemas de las políticas A y B. Esto surge porque en una sección de la secuencia Foreman con bajo movimiento la mayoría de los *slices* obtienen el menor nivel de prioridad posible, lo que los hace estar incluidos en las políticas y susceptibles de ser perdidos. Cuando se realiza el desarrollo de las políticas en esta secuencia se concentran las pérdidas



(a) PSNR



(b) SSIM

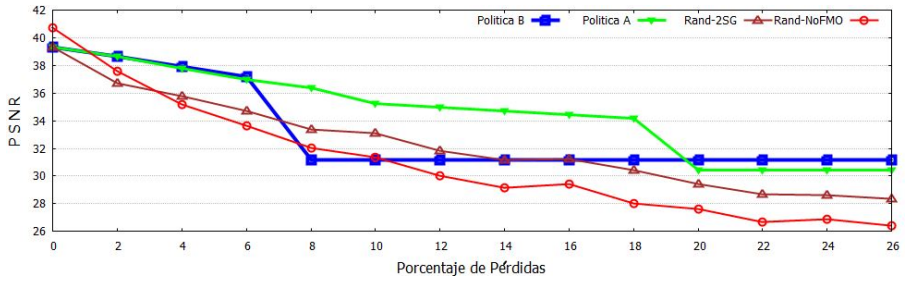


(c) VQM

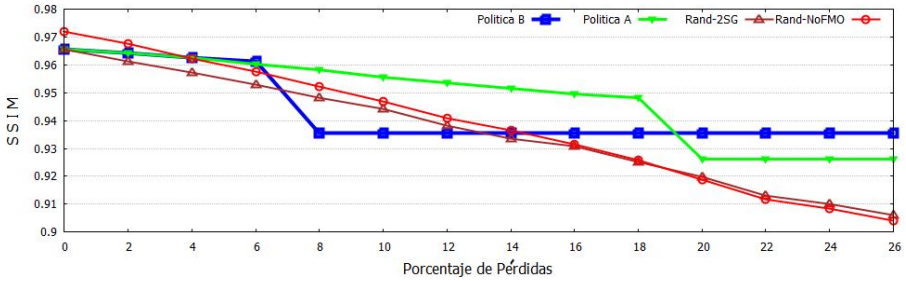
Figura 5.12: Comparación de Políticas Secuencia Foreman 512 Kbps GOP-10

a un grado no soportado por las herramientas de recuperación de error en el decodificador, que genera pérdida de sincronización del que no se puede recuperar el decodificador porque las pérdida de varias imágenes completas consecutivas rompe totalmente el proceso de predicción y dependencias de codificación.

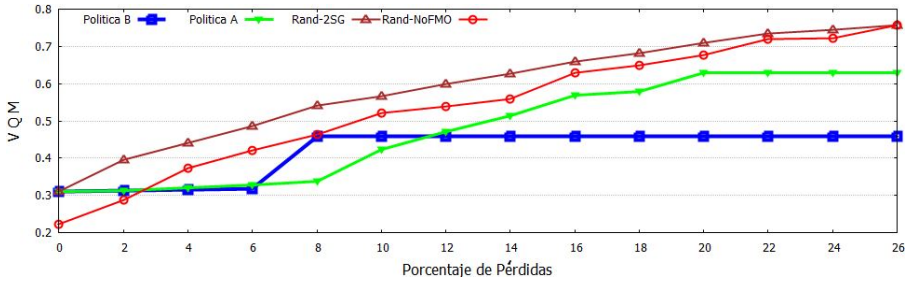
En las políticas A y B surgen diferencias cuando las pérdidas están en valores del 10 % y 24 %. Para la primer diferencia probablemente se de deba a que en la política A se eliminan *slices* de baja prioridad de imágenes tipo IDR que impactan ligeramente la calidad a través de la propagación de error a imágenes posteriores del GOP, mientras que en la política B primero se eliminan *slices* de bajas prioridades que cumplen con la propiedad de no ser adyacentes a otro previamente perdido. Para el porcentaje de pérdidas de 24 % la diferencia surge



(a) PSNR



(b) SSIM

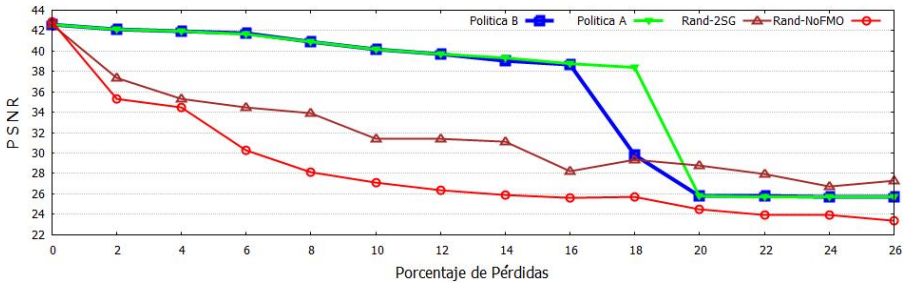


(c) VQM

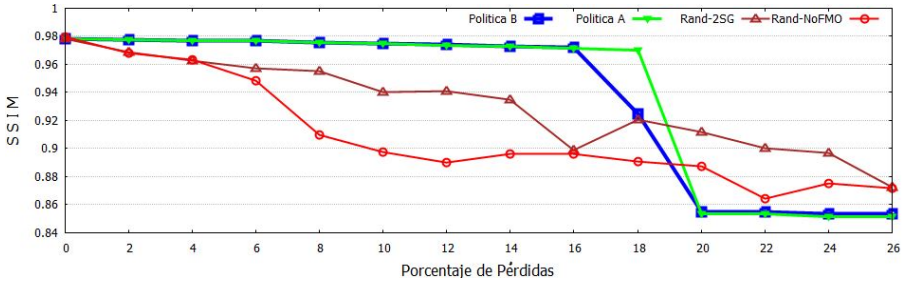
Figura 5.13: Comparación de Políticas Secuencia Akiyo 256 Kbps GOP-10

porque en el paso once de la política A se eligen *slices* tipo P sin evaluar si existen *slices* adyacentes a los perdidos previamente mientras para la política B si se evalúan los *slices* adyacentes.

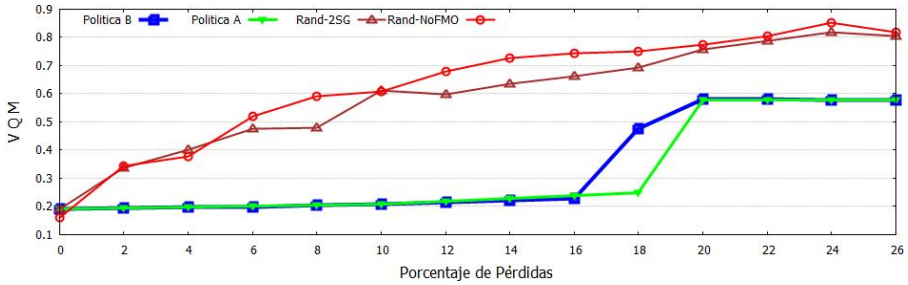
En las gráficas de las figuras 5.13 y 5.14 se muestra la evolución de las secuencias Akiyo 256 Kbps de $GOP = 10$ y $GOP = 60$ respectivamente. En la primera figura para la política B a partir de un porcentaje del 10 % la calidad no cambia mas, se mantiene fija en una linea recta horizontal. El motivo de esto es que los *slices* considerados por la política B se han agotado, tan solo representando el 10 % del tamaño total del video. Akiyo por sus características propias es fácil de comprimir y los *slices* generalmente tienen una corta longitud en bytes cuando pertenecen a imágenes tipo B y algunas tipo P.



(a) PSNR



(b) SSIM



(c) VQM

Figura 5.14: Comparación de Políticas Secuencia Akiyo 256 Kbps GOP-60

Los dos esquemas aleatorios tienen un desempeño inferior que los de las políticas en todas las secuencias evaluadas. Al ser aleatorios los esquemas, es posible considerar como esperado que el resultado sea inferior, lo sobresaliente de esta situación es la magnitud de la diferencia existente en los niveles de calidad. Desde el primer punto de evaluación la diferencia es de al menos 2 dB para el PSNR y de 0.2 para el VQM con tan solo 2 % de pérdidas. En los puntos de evaluación con mayores pérdidas esta diferencia se incrementa aun más llegándose a presentar diferencias de hasta 8 dB en PSNR.

Capítulo 6

Conclusión

En la actualidad el uso de aplicaciones de video se ha diseminado ampliamente en redes con cobertura inalámbrica. Con el objetivo de usar más eficientemente el ancho de banda, el video que se transmite sobre esas redes se codifica usando avanzadas técnicas de compresión. El estándar H.264 es uno de los estándares más recientes de codificación de video y que además cuenta con mucho mejor desempeño sobre sus predecesores en varios aspectos. Todos los beneficios que ofrece el estándar vienen con un costo implícito, se han optimizados muchos sub-procesos de la codificación como la predicción, transformación de dominio de la información para una mejor compresión, cuantización y estimación de movimiento por mencionar algunas. Pero todos estos sub-procesos del estándar conjuntamente lo convierten en una herramienta sumamente poderosa pero a su vez muy compleja.

La complejidad que representa el funcionamiento del estándar ha hecho que sea difícil vislumbrar claramente las importantes relaciones existentes entre cada aspecto de la configuración del codificador y las fortalezas o debilidades que tiene una secuencia de video codificada por este medio. Para su transmisión en ambientes propensos al error y con otro tipo de afectaciones. Es fundamental formular concepciones sencillas que hagan posible simplificar las relaciones referidas, con el fin que se cree una comprensión de como una secuencia de video podría afrontar reducciones del *bit rate* con desempeño satisfactorio o aceptable.

Durante los primeros capítulos de esta tesis se estudia la estructura del estándar H.264, de forma general se sigue cuál es el proceso de codificación de una secuencia. Es un proceso complejo y que conlleva un gasto computacional elevado. La recomendación oficial del estándar habla de la sintaxis y de como decodificar un video en este formato pero no especifica como se debe realizar la codificación, aunque la recomendación define gran parte del estándar precisamente aquí deja oportunidad para innovación dentro de él, como ejemplo es posible elegir de forma arbitraria el método para predicción de movimiento entre muchos otros más. Dentro del capítulo 2 se expusieron los bloques o procesos

de mayor relevancia, como la predicción espacial y temporal, transformación inversa, cuantización y codificación de longitud variable. De acuerdo al objetivo planteado de esta tesis, no es necesario desarrollar un conocimiento a profundidad de cada proceso pero si comprender muy adecuadamente su función.

En el capítulo 3 se expusieron características más especificas del estándar que se aprovechan para dar robustez al video y hacerlo más fuerte antes situaciones no deseables, estas características son muy útiles para el estudio de pérdidas. Dentro de las herramientas de resistencia al error más trascendentes están la ordenación flexible de macrobloques y los métodos de ocultación del error que hacen posible mejorar la experiencia al usuario ante pérdidas de información no controladas.

En el capítulo 4 se introduce la codificación en H.264 de las secuencias de prueba seleccionadas, se asigna una configuración al codificador con la activación de las herramientas de resistencia al error. En este capítulo se brinda una de las principales contribuciones de esta tesis, un método básico de ponderación de la importancia de los *slices* que componen un video basado en medición objetiva de la calidad. Fundamentalmente se descartan los *slices* y se mide el descenso de la calidad de la imagen a la que pertenecen y la propagación del error que se produce hasta el fin del GOP en el que se encuentra. Las dependencias de codificación y la propagación del error son factores determinantes para asignar un nivel de prioridad a cada *slice*. La prioridad del *slice* constituye un mecanismo con gran potencial de uso en redes de transmisión ya que permitiría establecer un control de tráfico más eficiente y una herramienta para el alivio de la congestión.

Luego de analizar la composición de los videos en función de Slice Group, tipo de imagen y prioridad se presentan las bases para poder desarrollar un patrón de pérdidas. Este análisis proporciono una nueva perspectiva de la relevancia de que se pensaba podrían tener cada tipo de slice en función al tipo de imagen de la que forman parte. Obviamente las imágenes IDR contienen la mayoría de *slices* con mayor importancia en las secuencias pero este análisis permitió conocer que también existen *slices* de imágenes IDR que son poco relevantes en la calidad y factibles de ser descartados si se requiere introducir pérdidas.

Los patrones de pérdidas, nombrados como política se abordan en el capítulo 5. Esta sección constituye otra aportación de la tesis ya que se propone un esquema que estipula una serie de pasos que conforman un esquema de pérdidas. En cada paso se definen las características de *slices* más factibles descartar por orden de importancia. Las características de los *slices* incluyen SG, prioridad, tipo de imagen a la que pertenecen. El patrón está basado en mediciones objetivas de la calidad por medio de métricas como el CMSE, PSNR, SSIM y VQM.

Las políticas propuestas se aplicaron a las secuencias junto con un esquema que simula el método *Drop Tail*. Los resultados muestran las ganancias en calidad que se logran con su uso. Las ganancias claramente marcan un tendencia muy favorable en rango de 2 *db* a 8 *db* para el PSNR.

6.1. Trabajo Futuro

Este trabajo cuenta con gran potencial a futuro que puede ser aprovechado de la siguiente forma:

Una idea interesante sería explotar las características de la partición de datos en H.264. Esta herramienta de codificación hace que los datos de video sean más robustos y resistentes a errores generados en canales inalámbricos de comunicación. Por lo que la priorización de datos en diferentes particiones ayudaría a mejorar el desempeño.

Es posible explorar diferentes configuraciones FMO como de tipo *Region of Interest (ROI)*. Durante la codificación de secuencias con selección de ROI existe una sección de la imagen o región de interés a la que se codifica con mayor calidad que la región de fondo. Por ejemplo la cara de una persona podría considerarse como una región de interés y para así asignarle un mayor nivel de prioridad.

A partir de las mediciones del CMSE que introducen las pérdidas de *slices* sería posible detectar un cambio de escena en la secuencia. Cuando un cambio de secuencia sucede se rompen las referencia de predicción, además de que las herramientas de cancelamiento de error funcionan de forma desacertada. Entonces los *slices* podrían ser nuevamente ponderados en importancia cuando se detecte este cambio.

La asignación de prioridad a los *slices* permitiría crear un punto de partida para implementar métodos de protección contra errores en las secuencias de video. Un método con gran potencial sería la combinación con Unequal Error Protection (UEP). De acuerdo al nivel de prioridad obtenido con el estudio de pérdidas se puede proteger en mayor medida a los *slices* con alta prioridad y proteger menos a los de baja prioridad para que no se registre demasiado incremento en el tamaño de la secuencia.

Basados en la prioridad de cada *slice* surge una idea interesante, con base en este parámetro llevar a cabo la administración del *buffer* de un nodo de red para ofrecer privilegios a los *slices* con prioridades altas a costa de los de bajas prioridades, esto considerando situaciones de alta congestión en la red de servicio. En combinación con la identificación de usuarios, la prioridad de *slices* podría ofrecer diferenciación en la calidad de servicio para cada usuario de distintas jerarquías.

Apéndices

Apéndice A

Perfiles y Niveles de H.264/AVC

Aplicación	*Requerimientos*	*Perfiles H.264*
Transmisión de TV	Eficiencia en la codificación, confiabilidad en canales de distribución controlados, transmisión entrelazada de los campos de cada cuadro de video, decodificador de baja complejidad.	Principal
video por Cable o Internet	Eficiencia en la codificación, confiabilidad en canales de distribución no controlados para redes basadas en paquetes de distribución y escalabilidad.	Extendido
Almacenamiento y reproducción de video	Eficiencia en la codificación, transmisión entrelazada de los campos de cada cuadro de video, decodificador de baja complejidad.	Principal
Videoconferencia	Eficiencia en la codificación, confiabilidad, baja complejidad del codificador y decodificador.	Básico
video a través de redes inalámbricas	Eficiencia en la codificación, confiabilidad, baja complejidad del codificador y decodificador, bajo consumo de potencia.	Básico
Distribución de video	Características de compresión sin pérdidas o cercanas, transmisión entrelazada de los campos de cada cuadro de video, transcodificación efectiva	Principal Alto

Tabla A.1: Requerimientos técnicos de video para distinto tipo de aplicaciones

Característica / Perfil	*Baseline*	*Extended*	*Main*	*High*
slices I P	√	√	√	√
slices B	×	√	√	√
slices SI SP	×	√	×	×
Múltiples Imágenes de Referencia	√	√	√	√
Deblocking Filter	√	√	√	√
CAVLC	√	√	√	√
CABAC	×	×	√	√
Flexible MB Order	√	√	×	×
Arbitrary Slice Ordering	√	√	×	×
Redundant *slices*	√	√	×	×
Data Partitioning	×	√	×	×
Interlaced Coding	×	√	√	√
4:2:0 Chroma Format	√	√	√	√
4:0:0 Monochrome	×	×	×	√
4:2:2 Chroma	×	×	×	×
4:4:4 Chroma	×	×	×	×
8 bits Sample Deep	√	√	√	√
9 and 10 bit Sample Deep	×	×	×	×
11 to 14 bit Sample Deep	×	×	×	×
8x8 vs. 4x4 Transform Adaptivity	×	×	×	√

Tabla A.2: Características Habilitadas en cada Perfil de H.264

Nivel	*Máxima tasa de procesamiento de MB (MB/s)*	*Máximo tamaño de la imagen en MB*	*Máximo tamaño del Bufer de imágenes decodificadas en MB*	*Máxima Tasa de Transmisión de video*	*Ejemplos en High. Resolution @ Frame-Rate (Max Store Frames)*
1	1485	99	396	64 Kbps	128×96@30.9(8) 176×144 @15.0(4)
1b	1485	99	396	128 Kbps	128×96@30.9(8) 176×144@15.0(4)
1.1	3000	396	900	192 Kbps	176×144@30.3(9) 352×288@7.5(2)
1.2	6000	396	2376	384 Kbps	320×240@20.0(7) 352×288@15.2(6)
1.3	11880	396	2376	768 Kbps	320×240@36.0(7) 325×288@30.0(6)
2	11880	396	2376	2 Mbps	320×240@36.0(7) 325×288@30.0(6)
2.1	19800	792	4752	4 Mbps	352×480@30.0(7) 325×576@25.0(6)
2.2	20250	1620	8100	4 Mbps	352×480@30.7(10) 720×576@12.5(5)
3	40500	1620	8100	10 Mbps	352×480@61.4(12) 720×576@25.0(5)
3.1	108000	3600	18000	14 Mbps	720×480@80.0(13) 1280×720@30.0(5)
3.2	216000	5120	20480	20 Mbps	1280×720@60.0(5) 1280×1024@42.2(4)
4	245760	8192	32768	20 Mbps	1280×720@68.3(9) 2048×1024@30.0(4)
4.1	245760	8192	32768	50 Mbps	1280×720@68.3(9) 2048×1024@30.0(4)
4.2	522240	8704	34816	50 Mbps	1920×1088@64.0(4) 2048×1088@60.0(4)
5	589824	22080	110400	135 Mbps	1920×1088@72.3(13) 3680×1536@26.7(5)
5.1	983040	36864	184320	240 Mbps	1920×1088@120.5(16) 4096×2304@26.7(5)
5.2	2073600	36864	184320	240 Mbps	-

Tabla A.3: Limites de Capacidad para Niveles en H.264

Apéndice B

Sistema de Visión Humano

Aunque el procesamiento digital de video se basa en formulaciones matemáticas, la percepción humana juega un rol fundamental por lo que es necesario comprender las limitaciones físicas de la visión. El sistema de visión humano es incapaz de distinguir una sucesión rápida de imágenes. Al observar su respuesta en frecuencia, se determinó que su comportamiento correspondía al de un filtro paso-bajas, cuya frecuencia de corte se ubicaba en el intervalo de 24 a 30 imágenes por segundo. Este fenómeno se puede apreciar en los televisores antiguos, pues su frecuencia de barrido vertical es de menor rapidez que el necesario, para que el ojo pueda ver una imagen continua, presentando un efecto conocido como *flicker*. También la respuesta en frecuencia del ojo puede variar, según la intensidad de la luz, ante imágenes poco brillantes la frecuencia de corte es menor, y con imágenes altamente brillantes la frecuencia de corte aumenta.

El ruido en las imágenes se manifiesta como un aumento de la brillantez de algunos pixeles más que en otros, es un problema común y complicado de resolver. La nitidez es la ausencia de ruido, entre mas nítida sea una imagen menos ruido tiene. El ojo humano es más tolerante con el ruido que contra la falta de nitidez. Solo cuando hay demasiado ruido es preferible en una imagen es preferible filtrarla. Como el ruido existe tanto en frecuencias altas como en frecuencias bajas y las imágenes sólo tienen valores significativos de amplitud en frecuencias bajas, una forma de suprimir el ruido de la imagen es aplicándole un filtro pasa-bajas, para eliminar componentes de alta frecuencia.

Otro fenómeno de la percepción humana es la creación de ilusiones ópticas, los ojos pueden captar información no existente o percibir propiedades geométricas erróneas de los objetos.

Percepción del Relieve El SVH posee la capacidad de percibir el relieve mediante el uso de mecanismos tanto psicológicos como fisiológicos, según la visión sea monocular o binocular, que tiene su origen en la percepción de información producto de la experiencia.

1. El tamaño de los objetos es relativo, si un objeto A que posee las mismas dimensiones que un objetivo B, se ve de mayor tamaño, entonces el objeto A está más cerca.
2. El grado de detalles o nitidez que posee un objeto es mayor cuando esta mas cercano.
3. Asociada a la perspectiva la decoloración de objetos ocurre en forma directamente proporcional a la distancia.
4. Cuando un objeto es cubierto por otro, entonces el primer objeto está más cercano.
5. Diferencias en el enfoque del ojo son un factor fisiológico complementario en la determinación de la cercanía de los objetos.

Adaptación de Luminosidad Dado que las imágenes digitales son desplegadas como un conjunto discreto de intensidades, la capacidad del ojo para discriminar entre niveles de intensidad diferentes es una importante consideración en el procesamiento digital de imágenes y video. El rango de niveles de intensidad de luz al cual el sistema de visión humano puede adaptarse es muy amplio, de varios ordenes de magnitud, desde el umbral estócopico al límite del resplandor. La evidencia experimental indica que la luminosidad subjetiva (intensidad percibida por el sistema visual humano) es una función logarítmica de la intensidad de luz incidente en el ojo. El rango total de niveles de intensidad que puede discriminar simultáneamente es muy pequeño comparado con el rango de adaptación total. Para cualquier conjunto dado de condiciones, el nivel de sensibilidad actual del sistema visual es llamado el nivel de adaptación de luminosidad.

La capacidad del ojo para discriminar entre cambios de intensidad de luz en cualquier adaptación específica es también de considerable interés. Un experimento clásico empleado para determinar la habilidad del sistema visual humano para la discriminación de luminosidad consiste en echar una mirada en un plano, uniformemente iluminado de gran área, suficiente para ocupar el campo entero de la vista. Esta área típicamente es un difusor, tal como un vidrio opaco, que es iluminado por una fuente de luz cuya intensidad I, puede ser variada. A este campo es añadido un incremento de iluminación ΔI, en forma de un flash de corta duración que aparece como un círculo en el centro del campo iluminado uniformemente.

Si ΔI no es suficiente brillante, el individuo da una respuesta negativa, indicando ningún cambio perceptible. Cuando ΔI es muy fuerte, el individuo dará una respuesta positiva todo el tiempo. La cantidad donde el incremento de iluminación discriminable es del 50 % del tiempo con iluminación de fondo, es llamada razón o promedio de Weber. Un valor pequeño significa que un cambio pequeño en porcentaje es discriminable y se cuenta con buena discriminación de la luminosidad. De manera inversa, un valor grande significa que un cambio grande de porcentaje en intensidad es requerido para ser detectado.

Bibliografía

[1] S. Kumar et al., "Error Resiliency Schemes in H.264/AVC Video Coding Standard", Elsevier J. of Visual Communication and Image Representation, vol 17, no. 2, pp. 425-450, Apr. 2006.

[2] Nemethova et al., "Robust Error Detection for H.264/AVC Using Relation ased Fragile Watermarking", Paper presented at the International Conference on Systems, Signals and Image Processing (IWSSIP), Budapest, Hungary, September 2006.

[3] `http://iphome.hhi.de/suehring/tml/`

[4] Thomas Wiengand et al., "Overview of the H.264/AVC Video Coding Standard", Transaction on Circuits and Systems dor Video Technology, Vol. 13, No. 7, July 2003.

[5] T. Connie et al., "Video Packetization Techniques for Enhancing H.264 Video Transmission over 3G Networks, in 5th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, 2008, pp. 800-804.

[6] P. J. Lee et al., "A New Error Concealment Algorithm for H.264 Video Transmission", in Proceedings of IEEE International Symposium on Intelligent Multimedia, Hong Kong, China, 2004, pp.619-622.

[7] P. Nasiopoulos et al., "An Improved Error Concealment Algorithm for Intra-Frames in H.264/AVC", Proc. IEEE Int. Symp. Circuits Syst., vol. 1, pp.320-323, 2005.

[8] S. K. Bandyopadhyay et al., "An Error Concealment Scheme for Entire Frame Losses for H.264/AVC?, in IEEE Sarnoff Symposium, Princeton, NJ, 2006, pp. 27-28

[9] G. Bjøntegard and K. Lillevold, "Context-adaptive VLC (CAVLC) coding of coefficients?, Fairfax, VA, Tech. Rep. C028, ITU-T VCEG — ISO/IEC MPEG (JVT), May 2002.

[10] J. Korhonen and P. Frossard, "Bit-error resilient packetization for streaming H.264/AVC video, in Proceedings of the International Workshop on Workshop on Mobile Video, Augsburg, Germany, 2007, pp. 25-30.

[11] S. K. Bandyopadhyay et al., "Frame los error concealment for H.264/AVC", presented at the 73rd MPEG meeting and 16th Joint Video Team meeting. Pozman, Poland July 2005.

[12] P. Nasiopoulos et al., "An Improved error concealment algorithm for intra-frames in H.264/AVC", Proc. IEEE Int. Symp. Circuits Syst., vol. 1, pp. 320-323, 2005.

[13] D. Levine, William E. Lynch., "Observations on Error Detection in H.264", 50th Midwest Symposium on Circuits and Systems, 2007. Pages: 815-818.

[14] http://iphome.hhi.de/suehring/tml

[15] Zhou Wang et al., "Image Quality Assessment: From Error Visibility to Structural Similarity", IEEE Transactions on Image Processing, Vol. 13 No 4 April 2004.

[16] S. Mys et al., "A Performance Evaluation of the Data Partitioning Tool in H.264/AVC?, Proc. of SPIE, vol. 6391, p. 639102, Oct. 2006.

[17] Y. Xu, and Y. Zhou, "H.264 Video Communication Based Refined Error Concealment Schemes", IEEE Trans. Consum. Electron., vol. 50, no.4, pp. 1135-1141, 2004.

[18] Siwei Ma, Wen Gao, "Rate-Distortion Analysis for H.264/AVC Video Coding and its Application to Rate ControlÏEEE Transactions on Circuits and Systems for Video Technology, Vol. 15 December 2005, pp. 1533-1544.

[19] Tian Hanmei, "A Review of Error Resilience Technique for Video Coding Using Concealment", IEEE International Conference on Information Science and Engineering (ICISE2009) 2009.

[20] Yen Lin Tung et al., "An Error Detection and Concealment Scheme for H.264 Video Transmission", IEEE International Conference on Multimedia and Expo (ICME) 2004.

[21] Yao Wang, "Error Control and Concealment for Video Communication: A Review", Proceedings of the IEEE, Vol. 86, No. 5, May 1998. pp. 974-997.

[22] Guan-Lin Wu, Shao-Yi Chien, "Spatial-Temporal Error Detection Scheme for Video Transmission over Noisy Channels", Ninth IEEE International Symposium on Multimedia 2007

[23] Du Li, Junquin Wu, "The Test and Analysis of FMO Model in Error Concealment Based on H.264", IEEE International Conference on Computer Science and Education (ICCSE 2011) August 3-5, 2011.

[24] Shyamprasad Chikkerrur et al., "Objective Video Quality Assessment Methods: A Classification, Review, and Performance ComparisonÏEEE Transactions on Broadcasting, Vol. 57, No. 2, June 2011.

[25] Margaret H. Pinson and Stephen Wolf, "A New Standardized Method for Objectively Measuring Video Quality", NTIA General Model and its associated calibration techniques.

[26] Vineeth S. Kolkeri et al., "Error Concealment Techniques in H.264/AVC for Wireless Video Transmission in Mobile Networks", Advances in Engineering Science Sect. C (3), July-September 2008, pp. 9-16.

[27] Zhou Xin and Zhjou, "An Effective Video Anti-error Algorithm for H.264", IEEE International Conference on Signal Processing Systems (ICSPS).

[28] H.264/14496-10 AVC Reference Software Manual. Source: Dolby Laboratories Inc., Fraunhofer-Institute HHI, Microsoft Corporation.

[29] Yuxia Wang et al., "Network-based Model for Video Packet Importance Considering Both Compression Artifacts and Packet Losses", IEEE Globecom 2010 proceedings.

[30] Yuxia Wang et al., "Packet Dropping for H.264 Videos Considering Both Coding and Packet-Loss Artifacts", Proceedings of 2010 IEEE 18th International Packet Video Workshop.

[31] Ting-Lan Lin et al. "Packet Dropping for Widely Varying Bit Reduction Rates Using a Network-Based Packet Loss Visibility Model", IEEE 2010 Data Compression Conference.

[32] Ting-Lan Lin et al. "Perceptual Quality Based Packet Dropping for Generalized Video GOP Structures", IEEE International Conference on Acoustics, Speech, and Signal Processing 2009.

[33] Imed Bouazizi, "Size-Distortion Optimization for Application-Specific Packet Dropping: The Case of Video Traffic", Proceedings of the Eighth IEEE International Symposium on Computers and Communication (ISCC'03).

[34] Cheng-Han Lin, "The Packet Loss Effect on MPEG Video Transmission in Wireless Networks", Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06).

[35] Sunil Kumar, "Robust H.264/AVC Video Coding With Priority Classification, Adaptive NALU Size and FragmentationÏEEE Milcom 2009.