Depto. de Teoría de la Señal y Comunicaciones
Universidad Carlos iii de Madrid

TESIS DOCTORAL

CONTRIBUTIONS TO RECONFIGURABLE VIDEO CODING AND LOW BIT RATE VIDEO CODING

Autor: Manuel de Frutos López
Director: Dr. Fernando Díaz de María

Leganés, 2012

TESIS DOCTORAL:
CONTRIBUTIONS TO RECONFIGURABLE VIDEO CODING AND LOW BIT RATE VIDEO CODING

Autor:
MANUEL DE FRUTOS LÓPEZ

Director:
DR. FERNANDO DÍAZ DE MARÍA

El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores:

Presidente:

Secretario:

Vocal:

acuerda otorgarle la calificación de:

Leganés, a

ABSTRACT

In this PhD Thesis, two different issues on video coding are stated and their corresponding proposed solutions discussed. In the first place, some problems of the use of video coding standards are identified and the potential of new reconfigurable platforms is put to the test. Specifically, the proposal from MPEG for a Reconfigurable Video Coding (RVC) standard is compared with a more ambitious proposal for Fully Configurable Video Coding (FCVC). In both cases, the objective is to find a way for the definition of new video codecs without the concurrence of a classical standardization process, in order to reduce the time-to-market of new ideas while maintaining the proper interoperability between codecs. The main difference between these approaches is the ability of FCVC to reconfigure each program line in the encoder and decoder definition, while RVC only enables to conform the codec description from a database of standardized functional units. The proof of concept carried out in the FCVC prototype enabled to propose the incorporation of some of the FCVC capabilities in future versions of the RVC standard.

The second part of the Thesis deals with the design and implementation of a filtering algorithm in a hybrid video encoder in order to simplify the high frequencies present in the prediction residue, which are the most expensive for the encoder in terms of output bit rate. By means of this filtering, the quantization scale employed by the video encoder in low bit rate is kept in reasonable values and the risk of appearance of encoding artifacts is reduced. The proposed algorithm includes a block for filter control that determines the proper amount of filtering from the encoder operating point and the characteristics of the sequence to be processed. This filter control is tuned according to perceptual considerations related with overall subjective quality assessment. Finally, the complete algorithm was tested by means of a standard subjective video quality assessment test, and the results showed a noticeable improvement in the quality score with respect to the non-filtered version, confirming that the proposed method reduces the presence of harmful low bit rate artifacts.

RESUMEN AMPLIADO

La codificación de vídeo ha sido un importante campo de investigación desde los albores de la televisión digital. La llegada de la televisión digital supuso un importante cambio de paradigma, no solo por la significativa mejora que suponía la transmisión digital respecto de la analógica, sino también por las posibilidades que ofrecen las potentes herramientas de procesado digital y, entre ellas, las relacionadas con la codificación de vídeo. Desde los primeros codificadores de vídeo que se desarrollaron, se ha realizado un esfuerzo notable para tratar de unificar los formatos de vídeo digital, sin duda inspirados por la experiencia adquirida durante la coexistencia de los distintos formatos analógicos, que supuso una considerable y constante fuente de problemas de compatibilidad. A tal fin surgieron las iniciativas de estandarización por parte de los organismos internacionales como ISO e ITU-T, formados por un conglomerado de insituciones gubernamentales nacionales e internacionales, grandes compañías y representantes del mundo académico, que tenían como objetivo fundamental garantizar la interoperabilidad de algoritmos y equipos en el marco de la codificación de vídeo.

Del esfuerzo de dichos organismos surgieron a lo largo de las últimas décadas toda una serie de estándares de codificación como H.261, H.263, MPEG-1, MPEG-2, MPEG-4, etc., que permitieron, en distintos ámbitos de aplicación, comprimir la información de vídeo de tal modo que fueran necesarios cada vez menos bits para transmitir o almacenar vídeo manteniendo la calidad. El modelo de codificador de vídeo que durante años ha inspirado estos estándares, debido principalmente a las elevadas tasas de compresión alcanzadas, ha sido el modelo híbrido DCT/DPCM por bloques. En este esquema se combina una predicción mediante compensación de movimiento, que permite explotar la redundancia temporal entre imágenes sucesivas, con una transformada en coseno, que permite representar de manera compacta el residuo de predicción de cada bloque de modo que un codificador entrópico reduzca de manera muy eficiente la tasa de salida. Dos de los codificadores más exitosos en las últimas décadas son codificadores de este tipo. En primer lugar, el descrito en el estándar MPEG-2, que se empleó en el omnipresente DVD, y que sirvió también como

núcleo del estándar de transmisión digital de televisión DVB. En segundo lugar, el descrito en el estándar H.264/AVC, que se ha convertido en uno de los más utilizados en el intercambio de archivos de vídeo por Internet y se ha incluido recientemente en el estándar DVB para la transmisión de televisión digital en alta definición (HD). En los últimos años, desde que saliera a la luz la última versión del estándar H.264, se ha venido trabajando en el desarrollo de un nuevo estándar de codificación de vídeo con el objetivo de reducir la tasa a la mitad respecto de H.264 para la misma calidad. Dicho estándar, conocido como HEVC, está orientado a la codificación de vídeo a muy alta resolución para atender la creciente demanda de los servicios de alta definición, y verá la luz en los primeros meses de 2013.

Primera parte

Aunque los beneficios del uso de los estándares de codificación de vídeo son muy importantes en un mundo digital tan heterogéneo como el que actualmente disfrutamos, son numerosas las voces que advierten de los perjuicios que puede acarrear el uso de estos estándares. Por un lado, el desarrollo y puesta en el mercado de nuevas ideas al amparo de los estándares de codificación es inherentemente lento, puesto que el proceso de estandarización requiere en sí mismo de una serie de pasos ineludibles antes de que un determinado algoritmo que mejore el estado del arte consiga ser introducido en un estándar existente. Tanto mayor será este retraso si se trata de generar un estándar nuevo. Por otro lado existe una cierta inercia respecto del uso de estructuras o algoritmos en los estándares, debido a que en el momento de proponerse la creación de un nuevo estándar, se comparan distintas opciones en conjunto y aquellas herramientas que se han optimizado para trabajar conjuntamente (por ejemplo porque formaban parte ya de un estándar anterior) suelen obtener mejores resultados que técnicas nuevas que, por más que sean prometedoras, no han tenido la oportunidad de rodearse de otras herramientas optimizadas para obtener un mejor resultado conjunto. Quizás sea esta una de las razones por las cuales las técnicas de descomposición en sub-bandas, que han dado un buen resultado en codificación de imagen, no terminan de resultar ventajosas en codificación de vídeo respecto del procesado por bloques y la DCT, que han estado presentes desde los primeros estándares.

Cabe señalar, que si bien los estándares permiten la interoperabilidad de equipos y garantizan la compatibilidad, aún no solucionan uno de los problemas más graves de los fabricantes: la coexistencia de distintos codecs en un mismo equipo. Este es el problema que deben afrontar, por ejemplo, los fabricantes de reproductores de discos Blu-ray, ya que deben ser compatibles con los formatos MPEG-2, H.264/AVC y VC-1. Teniendo en cuenta las notables diferencias entre estos formatos, lo normal es que estos equipos (*Set Top Boxes*) presenten un hardware dedicado distinto para cada uno de los formatos.

De manera similar, la implementación de un *codec* (codificador-decodificador) según un estándar determinado debe hacerse de manera monolítica, dando soporte a todas sus funcionalidades de modo que resulte compatible conforme a ese estándar. Este *todo o nada* reduce seriamente la flexibilidad y a menudo implica que habrán de implementarse procesos y algoritmos que no serán necesarios, o que incluso serán contraproducentes desde el punto de vista de la eficiencia de compresión, para el tipo de información de vídeo que se maneja en una determinada aplicación. Incluso la definición en los estándares de una serie de *perfiles*, que precisamente van encaminados a definir dentro del estándar subconjuntos de herramientas de codificación más apropiadas para cada tipo de aplicación, constituyen solo una mejora relativa de la flexibilidad.

Por último, y no menos importante, se ha demostrado en numerosos trabajos en la literatura que la adaptación de las herramientas de codificación al contenido permite un mejor rendimiento del codificador desde el punto de vista tasa-distorsión. De hecho, cada nuevo estándar presenta una mayor flexibilidad de sus algoritmos, permitiendo controlar su funcionamiento a través de numerosos parámetros de entrada. Sin embargo, el uso de un mayor número de parámetros de configuración supone sobrecargar la sintaxis del flujo de bits, de modo que quizás fuera más acertado poder adaptar la estructura del codificador sobre la marcha (y no solamente elegir los parámetros de codificación de algunos de sus bloques), como veremos más adelante.

A la vista de esta problemática, la primera parte de esta Tesis Doctoral estará dedicada a explorar las posibilidades de la *codificación de vídeo reconfigurable*, un nuevo paradigma que revoluciona el modo en que se definen los estándares de codificación. El estándar desarrollado en este sentido por parte de los expertos del MPEG se denomina símplemente codificación reconfigurable de vídeo (RVC). Desde los albores de este nuevo *estándar de estándares*, el autor de esta tesis ha formado parte de un grupo de trabajo orientado a definir una plataforma que evite la rigidez impuesta por los estándares tradicionales de codificación de vídeo, la *codificación de vídeo totalmente configurable* (FCVC de sus siglas en inglés), alternativa (o complementaria) a la propuesta oficial desarrollada en el estándar RVC. Según FCVC, el codificador podría variar su estructura sobre la marcha, por ejemplo sustituyendo unas herramientas de codificación por otras o modificando ligeramente alguno de los algoritmos utilizados, y enviar al decodificador una información de reconfiguración conjuntamente con el flujo de bits correspondiente al vídeo codificado, que permita alterar la estructura del decodificador de modo que sea capaz de reconstruir la información de vídeo.

En primer lugar se llevó a cabo una prueba de concepto para comprobar hasta qué punto es posible reconfigurar un bloque del codificador, notificar los cambios al decodificador, y obtener un beneficio neto desde el punto de vista de la tasa-distorsión con respecto a la opción no reconfigurable. Para ello, se desarrolló un mecanismo de codificación mediante transformada adaptable, sustituyendo la transformada DCT, que se emplea habitualmente en codificación híbrida, por otras transformadas como la de Hadamard o Haar. En este ámbito, el autor de la presente Tesis desarrolló el algoritmo de codificación de transformadas para que estas sean enviadas al decodificador. Este mecanismo está basado en la interpretación, codificación y reconstrucción de una serie de grafos que representan estas transformadas, tratando de reducir al mínimo el número de bits necesarios para hacerlas llegar al decodificador de modo que no se empeore el rendimiento neto del codificador. Esta prueba piloto resultó un éxito y sirvió para determinar una hoja de ruta para definir el sistema completo de FCVC para cuando, en lugar de reconfigurar únicamente un bloque del codificador, toda la estructura del mismo pueda variarse.

De las tareas a realizar, probablemente la más importante fue la definición de una sintaxis que permitiera transmitir al decodificador la información necesaria para actualizarlo de acuerdo con los correspondientes cambios que se decidan en el lado del codificador. Se definieron para ello las características que habría de tener dicha sintaxis y se propuso un primer conjunto de instrucciones y el protocolo adecuado de comunicación entre codificador y decodificador. Aprovechando los conocimientos extraídos de la prueba piloto, el autor colaboró con un grupo de trabajo para desarrollar la denominada sintaxis de descripción del decodificador (DDS). Paralelamente, se generaban descripciones de las herramientas de codificación con la nueva sintaxis y se desarrollaba un decodificador universal capaz de interpretar la información de reconfiguración codificada mediante la DDS, de modo que se actualice el software de decodificación. Durante este proceso se llevaron a cabo actualizaciones de dicha sintaxis que dieron lugar a un conjunto de instrucciones definitivo que cumplía con los objetivos marcados.

Una vez integrada la DDS en la nueva plataforma de FCVC, pudieron realizarse nuevas pruebas de rendimiento tasa-distorsión para un escenario más realista. Los resultados obtenidos para un codec H.264 mostraron que, aparte de solucionar desde un punto de vista teórico algunos de los problemas de los estándares mencionados anteriormente, la adaptación del codificador sobre la marcha que propone FCVC es beneficiosa desde el punto de vista de la tasa-distorsión. Incluso aunque se incorpore al flujo de bits de salida del codificador la información necesaria para la reconfiguración, el beneficio neto de adaptar el codificador al contenido (secuencia de vídeo de entrada) permite una reducción significativa de la tasa necesaria para una calidad dada (o, equivalentemente, aumenta la calidad percibida para una tasa fija) respecto de la opción no reconfigurable. Algunos de los puntos flacos de FCVC tienen que ver con la posibilidad de reconfigurar literalmente cada línea de código de cada función dentro del codificador. Esto plantea algunos problemas de seguridad por poder concurrir código malicioso o erróneo que impida el correcto funcionamiento de equipos FCVC. Además, el mecanismo de toma de decisiones para la reconfiguración del codec adquiere una complejidad impensable si se le da toda la libertad para variar la estructura a alto y bajo nivel del esquema de codificación-decodificación.

Paralelamente, el desarrollo del estándar RVC continuaba su andadura. Su motivación podría decirse muy similar a la de la plataforma FCVC pero con algunas diferencias notables. Inicialmente, el codificador se configuraba en RVC al principio de la sesión mediante la interconexión de una serie de bloques estándar (unidades funcionales) que se corresponderían con las herramientas más típicas empleadas en los estándares desarrollados por el ISO e ITU-T durante los últimos años. A estas herramientas, se les podría añadir nuevas funcionalidades no estándar definidas por el usuario, pero en cualquier caso la configuración elegida al principio del proceso de codificación sería la que se mantendría para toda la sesión. Aunque, en efecto, este enfoque también permite solucionar algunos de los problemas más importantes de los estándares, no contempla la posibilidad de reconfigurar el codec sobre la marcha, atendiendo a criterios de adaptación al contenido. Estas diferencias constituyeron un interesante tema de discusión con los miembros del grupo desarrollador del RVC, que concluyó con la presentación de dos propuestas ante el MPEG encaminadas a incluir algunas funcionalidades que sobre el papel ofrecía FCVC en las nuevas versiones de RVC.

Aunque no se llegó a encontrar solución práctica para algunos de los retos que suponía la reconfiguración total del codec sobre la marcha que planteaba FCVC, los resultados fueron prometedores y, quizás como consecuencia de este esfuerzo, el nuevo estándar RVC ya incorpora la posibilidad de cambiar la configuración sobre la marcha, aunque esté restringido a emplear unidades funcionales de una base de datos estándar. En cualquier caso, creemos que con este trabajo se ha puesto de manifiesto que la idea de FCVC presenta un gran potencial y que podría merecer la pena evolucionar la idea original de la codificación reconfigurable en esa dirección.

Segunda parte

Durante los últimos años, el continuo despliegue y mejora de las redes móviles ha logrado incrementar notablemente su ancho de banda disponible, permitiendo a las aplicaciones multimedia, que tradicionalmente se habían limitado a las redes fijas con mayor ancho de banda, expandirse a los escenarios móviles. Esto no solo ha permitido que las aplicaciones tradicionales como la transferencia de vídeo, la video-telefonía o el *streaming*, puedan llegar cada vez a un mayor número de usuarios y dispositivos, sino que ha propiciado incluso la aparición de nuevas aplicaciones relacionadas con la movilidad, como servicios que tienen en cuenta la localización, aplicaciones de realidad aumentada, códigos QR, etc. Sin embargo, en lo concerniente a la transmisión de vídeo, los dispositivos móviles presentan aún una serie de restricciones que suponen importantes retos para el desarrollador de herramientas de codificación.

Por un lado, la transmisión en canales radio supone tener que enfrentarse a numerosas fuentes de distorsión, desvanecimientos y, en definitiva, a un entorno muy hostil para la transmisión de señales multimedia, sobre todo cuando se precisa una comunicación en tiempo real, que generalmente exige un bajo retardo. La tasa disponible en este tipo de canales puede caer notablemente sin previo aviso, haciendo necesario codificar de manera muy agresiva la información de vídeo para que no cese la comunicación. Por otro lado, la mayor parte de las herramientas de codificación de canal que se emplean para proteger los datos frente a este entorno hostil, suponen una reducción adicional de la tasa binaria disponible para la información de vídeo, agravando aún más el problema de la escasez de recursos.

A las dificultades propias del uso de este canal es necesario añadir las peculiaridades de los dispositivos involucrados. Si bien la capacidad de cómputo de los dispositivos móviles está creciendo significativamente en los últimos tiempos, también es cierto que el uso del procesador precisa de un consumo energético que afecta al tiempo de vida de las baterías, un recurso que aún sigue siendo bastante escaso. Del mismo modo, el *hardware* de transmisión de los dispositivos móviles supone un consumo importante de energía y, por tanto, a mayor cantidad de información transmitida, menor duración de la batería.

En este entorno, los codificadores híbridos de vídeo han de mantener una complejidad relativamente baja, aun a costa de no obtener el mejor rendimiento posible desde el punto de vista de la tasa-distorsión. A la hora de reducir la tasa de salida, el codificador emplea normalmente escalones de cuantificación altos, por ser esta la forma más sencilla de controlar el flujo de datos codificados. El empleo de escalones de cuantificación altos supone, además de un aumento de la distorsión media de la secuencia reconstruida respecto a la original, la posible aparición de artefactos de codificación como el efecto de bloques, el desenfocado, el efecto mosquito, el efecto de funciones base, etc.

En la presente Tesis Doctoral se pretende analizar algunos de los mecanismos empleados para simplificar las secuencias de vídeo con que se alimenta al codificador antes de que sean codificadas, para evitar el empleo de escalones de cuantificación muy elevados. En la literatura pueden encontrarse numerosas propuestas de filtros que permiten reducir las altas frecuencias presentes en las secuencias de vídeo a codificar, y que son las más costosas para el codificador en cuanto a tasa, de modo que el codificador emplee escalones de cuantificación menores para la misma tasa objetivo, reduciéndose el riesgo de aparición de artefactos típicos de bajas tasas como los mencionados anteriormente.

En primer lugar, aunque la idea ya aparecía en alguno de los trabajos consultados, se propone emplear un algoritmo de simplificación, en el que el filtrado se lleva a cabo sobre las muestras del residuo de predicción en lugar de filtrar las los píxeles originales de la imagen, de modo que se tiene un mayor control sobre la relación entre el potencial ahorro de bits a la salida del codificador y la cantidad de filtrado impuesta (los valores del residuo de un bloque están más directamente relacionados con su coste en bits que los píxeles originales). Dicho algoritmo fue implementado sobre un codificador H.264/AVC con el fin de encontrar la mejor configuración del filtro y diseñar un mecanismo de adaptación del mismo.

A continuación, se realizó un estudio de las distintas opciones para el filtrado del residuo, primando aquellos filtros que presentan baja complejidad, dado el escenario de aplicación elegido. La evaluación de los distintos filtros se llevó a cabo observando como afecta cada uno de ellos al residuo y al posterior proceso de codificación del mismo. A esta configuración básica se le incorporó un mecanismo para controlar la cantidad de filtrado de acuerdo con el punto de trabajo del codificador, filtrando más cuando se vayan a emplear valores más altos de QP, susceptibles de originar artefactos de codificación visibles, y filtrando menos cuando la tasa disponible sea alta y la QP de trabajo sea inferior, evitando difuminar innecesariamente el vídeo reconstruido.

Sin embargo, en nuestra opinión las herramientas propuestas hasta ahora no cumplen adecuadamente con el objetivo perseguido de un control prefiltrado. Por un lado existen numerosas propuestas que únicamente presentan una configuración de prefiltrado fija con la que pre-procesar todas las secuencias antes de ser codificadas. Esta solución es sub-óptima por cuanto que la misma cantidad de filtrado puede ser adecuada para unas secuencias y, sin embargo, ser insuficiente o excesiva para otras (en el caso de un filtrado excesivo, el fenómeno más habitual es la aparición de un efecto de difuminado visible). Por otra parte, aquellas propuestas que incluyen un control de la cantidad de filtrado, suelen hacerlo dependiente de determinadas características de bajo nivel medidas en cada plano y promediadas para toda la secuencia, como por ejemplo medidas del efecto de bloques o medidas de complejidad relativa de las distintas regiones a codificar en el plano.

Como alternativa a estos planteamientos, trabajamos con la hipótesis de que la opinión subjetiva está más relacionada con una cierta impresión global acerca de la presencia de artefactos. Dado que además es muy complicado determinar la presencia o no de los mismos (no hay medidas del todo fiables para detectar efecto de bloques, difuminado o efecto de mosaico), se propone como alternativa una metodología de entrenamiento mediante valoración subjetiva de la calidad del vídeo respecto de la cantidad de filtrado. Incluso mediante un entrenamiento en que intervenga un número reducido de sujetos, es posible obtener unos valores indicativos de la cantidad de filtrado más adecuada para un determinado punto de trabajo del codificador (tasa disponible o QP empleada) y para cada secuencia, que sean aceptables por un espectador medio.

De los resultados de ese entrenamiento se obtuvieron una serie de funciones de coste que ayudaron a seleccionar las características más relevantes que permitieran entrenar un mecanismo de estimación de la cantidad de filtrado con consideraciones perceptuales. Dicho mecanismo adaptativo fue incluido en la implementación del filtrado del residuo realizada sobre el codificador H.264 para su posterior evaluación.

Se recabaron datos de codificación relativos a la secuencia reconstruida, como por ejemplo la PSNR, pero ante la falta de correlación entre esta y la calidad subjetiva apreciada en presencia de posibles artefactos como el efecto de bloques o el difuminado, se llevó a cabo una evaluación subjetiva estándar (acorde con la norma P-910 de la ITU que aplica en estos casos). Los resultados determinaron que, en promedio y para la mayoría de los sujetos que colaboraron en el experimento, el sistema propuesto mejora significativamente la calidad subjetiva respecto de la versión de referencia, obteniendo un valor de DMOS cercano a la unidad (en una escala de 0 a 4, siendo 0 que ambas versiones resulten indistinguibles y 4 que la versión sin filtrado se vea como mucho peor). Esto demuestra que el algoritmo permite paliar la influencia de los artefactos que se producen en la codificación de vídeo a baja tasa.

Agradecimientos

Siempre que, empujados por alguna extraña fuerza, conseguimos saltar un listón más alto que el anterior o alzarnos hasta la cima de una montaña más distante y elevada, echamos la vista atrás y tratamos de ponerle cara a esa fuerza que nos ha empujado, a esa extraña energía que ha surgido de nosotros de alguna manera. En no pocas ocasiones, esa energía tiene en realidad muchos rostros, y en solo unos minutos de reflexión es imposible citar todos los nombres o rememorar todos los gestos y pequeños hitos que nos han hecho capaces de llegar hasta la meta. Cuando además ponemos sobre nuestros hombros cargas como el miedo, la falta de confianza o el desánimo, entonces se hace mucho más importante que desde fuera nos animen a seguir, nos guíen, nos muestren las cosas con perspectiva o, simplemente, nos animen a mirar temporalmente a otro lado, a sonreír o a soñar. Pero si he llegado hasta aquí, aunque con ayuda, seguro que puedo sobreponerme a la dificultad de agradecer en este corto espacio a todos los que me han empujado hasta aquí y que me han hecho como soy. O al menos voy a intentarlo.

En primer lugar tengo que decir que esta Tesis no habría sido posible en absoluto sin el Dr. Fernando Díaz de María, mi supervisor. Y no quiero que esto pase como el típico agradecimiento protocolario, sino que de verdad tengo mucho que agradecerle por todo lo que me ha enseñado, como profesor, como jefe y como compañero. Por momentos ha tenido más fe en mí que yo mismo.

Por supuesto, este agradecimiento debe hacerse extensivo a mis compañeros del Departamento de Teoría de la Señal de la Carlos III, sobre todo a los del Grupo de Procesado Multimedia. He pasado tanto tiempo aquí que me ha dado tiempo a conocer a personas extraordinarias y compartir con ellos muchos buenos (y malos) momentos: Ángel Martín y Jesús de Vicente, a los que recuerdo con especial cariño por ser una especie de mentores; Jose Carlos Pujol, Samuel Jiménez, Inmaculada Luengo, Ana Isabel García, Yago Pereiro, Darío Martín, Oscar del Ama, con los que mantuve una gran amistad el tiempo que estuvieron en el grupo; Rosa Mª Barrio, Sara Pino, Rocío Arrollo, Luis Azpicueta, Mario de Prado, que aunque sean de otros grupos son majos; e Iván González, Sergio Sanz, Rubén Solera, Eduardo Martínez, Carmen Peláez, Chelus (no pongo el apellido que es muy largo), Amaya Jiménez, con los que he tenido que batallar en no pocos frentes y que tanto me han enseñado sobre el trabajo en equipo y la amistad. Ojalá pudiera escribir tan solo una frase para cada uno de ellos, en lugar de enumerarlos como la lista de la compra. Gracias, muchas gracias.

I would also like to thank Prof. Iain Richardson and the people at the CVC Labs at Robert Gordon University for the opportunity they gave me to work with them for a while in the development of the Fully Configurable Video Coding platform. Everything was nice in Aberdeen those days (even the weather), but it was especially nice to work with Iain, Sampath, Abhi, Maja and James.

Finalmente, tengo que salirme del ámbito laboral para agradecer todo su apoyo y Amistad (así, con mayúsculas) a mis amigos de Vallecas (e incorporaciones posteriores), con los que tanto he compartido y más que espero poder compartir a partir de ahora; a mis compañeros de la carrera, que no entienden cómo es posible que siga por la Universidad; a Marga, que lleva tanto tiempo a mi lado como esta Tesis (ya se empieza a sentir celosa) y no solamente me ha animado, empujado, apoyado, consolado, escuchado y aconsejado, sino que además ha sido para mí un ejemplo de esfuerzo, tesón y valentía.

El último agradecimiento especial es para mi familia, que han sido siempre el espejo en que me he mirado, mi punto de apoyo, mi lugar en el mundo, y que espero se sientan tan orgullosos de mí como yo de ellos. Por supuesto, un recuerdo muy especial debe ir para mi abuela Joaquina y se lo envío allá donde esté con todo mi cariño.

Ahora que lo pienso, estar o haber estado con todas estas personas que acabo de recordar es para mi mucho más valioso que cualquier logro, incluida esta Tesis. Pero al menos la Tesis me da la oportunidad de escribir: **muchas gracias a todos**.

A mi familia y amigos...

Contents

List of Figures

List of Tables

Acronyms

3DTV	3D Television
ADM	Abstract Decoder Model
AVC	Advanced Video Coding
AVS	Audio and Video coding Standard
BSDL	Bitstream Syntax Definition Language
BU	Basic Unit
CABAC	Context-Adaptive Binary Arithmetic Code
CAL	Cal Actor Language
CAVLC	Context-Adaptive Variable Length Code
CBR	Constant Bit Rate
CIF	Common Intermediate Format
CPU	Central Processing Unit
CSF	Contrast Sensitivity Function
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DVB	Digital Video Broadcasting

FNL	Functional Network Description
FTV	Free-viewpoint Television
FU	Functional Unit
GOP	Group of Pictures
HDTV	High Definition Television
HEVC	High Efficiency Video Coding
HVS	Human Visual System
IEC	International Electrotechnical Commission
ISDN	Integrated Services Digital Network
ISO	International Standardization Organization
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
JVT	Joint Video Team
KLT	Karhunen-Loéve Transform
LLVM	Low Level Virtual Machine
LS	Least Squares
MAD	Mean of Absolute Differences
MOS	Mean Opinion Score
MP3	MPEG-1/MPEG-2, Layer 3
MPEG	Moving Pictures Experts Group
MSE	Mean Squared Error

MVC	Multi-view Video Coding
NLM	Non-Local Means
OS	Order Statistics
PCA	Principal Component Analysis
PSNR	Peak Signal to Noise Ratio
QP	Quantization Parameter
RD	Rate-Distortion
RDO	Rate-Distortion Optimized
ROI	Region Of Interest
RT	Real Time
RVC	Reconfigurable Video Coding
SAD	Sum of Absolute Differences
SE	Structuring Element
SSD	Sum of Squared Differences
SVC	Scalable Video Coding
VBR	Variable Bit Rate
VCEG	Video Coding Experts Group
VLC	Variable Length Code
VQEG	Video Quality Experts Group
VTL	Video Tool Library
WLS	Weighted Least Squares

WMF	Weighted Median Filter

Chapter 1

Introduction

> *You see, wire telegraph is a kind of a very, very long cat. You pull his tail in New York and his head is meowing in Los Angeles. Do you understand this? And radio operates exactly the same way: you send signals here, they receive them there. The only difference is that there is no cat.* (Albert Einstein)

In this PhD Thesis, the author makes contributions to solve two problems in the field of video coding: 1) the use or reconfigurable platforms for avoiding some drawbacks of working under the auspices of video coding standards; and 2) a residue filtering-based technique for video simplification suitable for low bit rates. In this chapter, the motivation and objectives of both parts of the Thesis will be discussed, after a brief introduction to video coding focused on recent applications and standards. Finally, the contents of the rest of this document will be outlined.

1.1 Video Coding

In recent years, motivated and boosted by the omnipresence of Internet and the wide range of mobile devices available, multimedia applications have demanded a dramatically higher bandwidth from modern communication networks. Although the global network capacity is also continuously growing, the new universe of personal communications and device interconnections demands powerful algorithms for data compression.

Specifically, video applications like those related to 3DTV and Free-viewpoint TV (FTV), demand for a powerful set of tools for analysis, representation, and coding of multiple-view scenes. In this way, each different view to be processed and sent for a better understanding of the scene, implies additional bandwidth. Moreover, High Definition Television (HDTV), which has been broadly deployed as a consumer product in its 1080-line version, has opened a new research field for the enhancement of the user's experience in TV systems, expanding the resolution of digital video beyond imagination; the new standards for Ultra-high Definition TV (UHDTV) reach an astonishing resolution of 4K (4000-line) or even 8K (8000-line, the so-called Super Hi-vision). These resolutions allow for a better coverage of the visual field, achieving a more accurate representation of the scene and creating an immersive experience. However, the bandwidth demanded for transmission of so high resolutions is overwhelming.

Similarly, new multimedia applications related to handheld devices have expanded the concept of personal communications. For example, augmented reality is becoming a new scenario for devices with multimedia capabilities, and specifically, for mobile phone applications. In the same way, many complex image and video processing tools, including video encoders, are being implemented in devices with low processing capabilities (or low battery lifetime), even in scenarios as restricted as the sensor networks.

For the task of compressing video data, a wide variety of coding tools have been developed in the last 30 years with the main objective of achieving high quality video with fewer and fewer bits. The most widely used video encoders are based on the so-called hybrid block-based DCT/DPCM (Discrete Cosine Transform/Differential Pulse-Code Modulation) model, which entails the use of a prediction stage for taking advantage of spatiotemporal redundancies in video content, followed by a cosine transform stage for a frequency domain processing of the prediction residue, including quantization and entropy coding of transform coefficients. The coarseness of the quantization process usually sets the operation point at the encoder, establishing a compromise between output bit-rate and distortion according to the available resources.

Nevertheless, output bit rate is not the only concern in the design of a video codec (coder+decoder), but also the interoperability of such devices in an inherently heterogeneous environment. For this reason, modern video coding techniques have evolved under the sponsorship and regulation of international organizations such as the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG), whose standards guarantee the proper interoperability of video encoders and decoders, making possible that such different worlds as software development, hardware implementation, and algorithmic design work together in the same direction.

Among other interests, the ISO-MPEG developed standards aiming at digital video storage and broadcast applications, such as MPEG-1 and MPEG-2, with bit rates of 1.4Mbps and 3-5Mbps, respectively. On the other hand, the ITU-VCEG issued standards for real-time two-way video communications, usually over low bit rate networks, such as H.261 for video telephony over ISDN networks at $n \cdot 64$kbps or H.263 for Packet or Circuit Switched Networks at rates as low as 20-30kbps. Finally, the H.264/AVC coding standard, developed by both MPEG and VCEG together in the so-called Joint Video Team (JVT), is the latest of the standardization efforts in the field of video coding and it supposed a considerable improvement with respect to previous standards, not only in terms of coding gain but also in terms of flexibility, grounded in the combination of different profiles and levels, which correspond to different sub-sets of coding tools and available processing resources. By the use of these profiles and levels, H.264/AVC has become a good choice for video coding in a wide range of application scenarios, from real-time video telephony over wireless networks to digital video broadcasting (DVB) or video storage in Blu-Ray disks.

At the time this document is being written, a new video coding standard, High Efficiency Video Coding (HEVC) is under development and will probably come out as a final draft on January 2013. The core of HEVC is expected to be the same block-based DCT/DPCM hybrid scheme. In addition to several high complexity tools, it is aimed at halving the output bit rate with respect to H.264/AVC for the same quality at HD resolutions.

1.2 Motivation of this PhD Thesis

1.2.1 Pros and cons of video coding standards

Working with specific standards has helped to understand not only the pros and cons of the standards themselves but also the drawbacks inherent to the use of video coding standards in general. If we look first at the pros of video coding standards, the conclusion is evident: standards provide interoperability and we do not need additional reasons for them. Interoperability by itself justifies standards. Nevertheless, standards have also cons. The concurrence of different video coding formats in the same environment forces manufacturers to include multiple standard (and also proprietary) compression formats into their final products, i.e., the Set Top Boxes (STB).

Furthermore, as explained above, the evolution of video codecs tends to continuously increase the complexity of the video coding tools in order to further exploit the redundancies and reduce the output bit rate. Even if there exist applications (or video sequences) in which the use of these tools produce no benefit at all, video encoders and decoders are monolithically implemented. Nevertheless, video decoders have not such a flexibility given that they must decode any standard-compliant bitstream. Even with the use of profiles and levels, which enable to decide a sub-set of processing tools for the decoder implementation, many of the tools specified in a standard are never used in particular applications. For example, the ATSC Mobile/Handheld Digital TV standard [ATSC, 2008] specifies a reduced subset of the H.264/AVC baseline profile in recognition that certain options are not required for this application.

Finally, another drawback of modern video coding standards is that they somehow suppose a burden to creativity in the sense that competitiveness of new coding tools, which in addition suffer from a very long time to market under the requirements of standardization, is also compromised by the disadvantage against well-established and optimized standard-related algorithms.

1.2.2 Low bit rate video coding for mobile multimedia scenarios

The continuous development of mobile networks during the last decade has revolutionized the world of personal communications, bringing in multimedia content and Internet access to handheld devices. With an available rate of 2Mbps for 3G networks and expected to raise towards 100Mbps for 4G networks, technologies such as multimedia telephony, video streaming, multimedia messaging or broadcast systems, are gradually penetrating mobile scenarios. There are two main issues to have in mind when approaching to mobile communications: 1) the hostile channel, and 2) the power shortage.

In the first place, radio links are probably the most hostile channels for communications. The concurrence of high power losses and different sources of noise and fading (especially the multi-path induced fading), require the use of powerful channel coding tools for error correction. The basic idea is to introduce redundancy codes in order to detect whether a certain information is reliable or, in case it is corrupted, to recover the original message. Therefore, channel coding enhances the reliability while reducing the available bandwidth for video payload in wireless communications.

Apart from the channel encoding efforts, the high packet loss rates of wireless environments are typically mitigated by the source video encoder by means of some widely-used tools that can be generally grouped into three categories: 1) error propagation avoidance, in order to prevent the use of corrupted areas as the reference for predictions at the decoder; 2) redundancy introduction at entropy coding level, in order to recover a copy of lost data from the same bitstream; and 3) error concealing and recovering, which is aimed at estimating lost data from the available information at the decoder side. Although the use of these tools improves quality in the event of severe packet loss conditions, with the exception of the third, it also implies the reduction of the available resources for encoding the actual video sequence, requiring the video encoder to notably reduce the bit rate devoted to video payload.

Current Rate Control (RC) algorithms for video encoders try to adapt the output bit rate to the available resources while keeping quality as high as possible, by means of modifying encoding parameters such as prediction mode, block size and, above all, the quantization step. This last parameter is responsible for the losses generated by the video encoder: a higher quantization step, which is necessary for low bit rate encoding, reduces the amount of bits necessary to encode transform coefficients and, consequently, the quality of the reconstructed sequence drops.

Not only the particular QP value is important, but also the QP variations. Typical rate control algorithms for video coding are based on statistical models that are usually not reliable for very low or very high bit rate conditions, and the appropriate quantization parameter for each basic unit (namely a frame or a slice) is poorly estimated in these situations, causing a mismatch between the target bit rate and the bit rate actually consumed. Since most of the rate control algorithms have some mechanisms to monitor the short-term output bit rate, the result of this mismatch is usually a sudden fluctuation of the quantization parameter along consecutive frames, a behavior that not only penalizes the output quality of those frames with coarser quantization but also the overall quality, since that sudden quality changes in a sequence can be even more harmful for the observer that a sequence encoded with lower but constant quality. For an in-depth analysis of this problem, the reader is referred to [Yim and Bovik, 2011].

As mentioned, running a hybrid video encoder at very low bit rates implies the use of higher quantization steps. Apart from increasing the differences with respect to the original, these high QP values are sometimes responsible for the appearance of artifacts in the reconstructed sequence that severely compromise the final user's experience. The interested reader is referred to [Wu and Rao, 2005] for a comprehensive treatment of video coding artifacts. Here, however, we provide a quite complete list of the coding artifacts related to coarse quantization: basis image effect, blurring, color bleeding, staircase effect, ringing, mosaic patterns, false contouring, false edges, mosquito noise, and the blocking. The origin, appearance, and possible mitigation of these artifacts will be described in next chapters.

The second issue to be faced up in wireless communications is the lack of available power in handheld devices. While the capacity of modern batteries is continuously increasing, power consumption is still a major concern for the application designer. Specifically, processing and transmission hardware are the most power-demanding elements and, therefore, their use should be restricted as far as possible. Concerning processing resources, the last decades have witnessed a great development both in processing capabilities and simplified implementations of codec tools, partially overcoming this issue. On the contrary, the restrictions in transmission hardware power consumption imply a reduction of available bit rate: the less the bitstream size, the less the power consumed for video transmission.

In summary, the wireless multimedia environment requires the codecs to operate at low bit rates, where these video codecs are more prone to artifacts.

1.3 Objectives of this PhD Thesis

1.3.1 Fully Configurable Video Coding

If the monolithic structure of the standards can be seen as a burden to an efficient implementation of video codecs, how could the structure of the video coding standards be changed in order to allow for a higher flexibility? Recently, some efforts were made in adopting more flexible architectures for video encoder and decoder design and a new initiative for Reconfigurable Video Coding (RVC) is under development nowadays (a detailed description can be found in [Mattavelli et al., 2010]). The aim of RVC is to define a platform as well as a common syntax to allow for the decoding of multiple formats with the same universal reconfigurable decoder. This new paradigm makes possible to implement different decoders with only one dedicated hardware to support the reconfigurable engine and decoding tools, while enabling designers to select the most appropriate subset of encoding tools for a particular set of video data, saving processor, memory, and other computational resources.

Although the RVC standard, as described in [Mattavelli et al., 2010], aims at solving some of the drawbacks of the use of traditional standards, it is the first main objective of this PhD Thesis the analysis of the theoretical advantages and disadvantages of developing a new Fully Configurable Video Coding (FCVC) platform, as well as their discussion in the light of some experiments carried out with an FCVC prototype.

1.3.2 Video simplification for low bit rate multimedia applications

The second objective is to reduce the artifacts derived from low bit rate coding. Some contributions have been made by the author of this Thesis to the field of rate control for video coding for constant and variable bit rate channels (see [Sanz-Rodríguez et al., 2010] and [de Frutos-Lopez et al., 2010]). One of the conclusions extracted from these and other works indicate that the use of hybrid video codecs in very low bit rate environments entails the use of very large quantization steps and, therefore, the appearance of encoding artifacts diminishing the overall subjective quality is unavoidable. This led us to the questions motivating the second part of this PhD Thesis: What are the limits of the RC algorithms when working at very low bit rate? What is still to be done for enhancing the subjective quality of reconstructed video sequences subject to these tough rate constraints?

In this PhD Thesis, alternative techniques for bit rate reduction in hybrid video codecs are proposed to keep the average quantization step low and to avoid the encoding artifacts related to coarse quantization before they arise. In several previous works the use of a pre-filtering stage for the simplification of video sequences prior to the encoding process has been proved as a powerful tool for bit rate reduction (see for example [Lin and Ortega, 1997] or [Segall et al., 2001]). Unfortunately, all pre-filter techniques suffer from the same drawback: removing non-relevant components of video information can also introduce distortion in other components, generating new artifacts. As a way of an example, Gaussian bi-dimensional filters or bilateral filters, which have been widely employed for both noise reduction and simplification in image coding, produce blurring effects when applied too coarsely. Additionally, applying a pre-filtering stage to video content prior to the encoding process does not take into account the fact that certain areas of video sequence may be properly estimated by the spatial and temporal prediction techniques and hardly any resources are spent to encode the corresponding prediction residue. The benefits of the use of filters in this areas are therefore negligible in terms of bit rate reduction while an additional distortion is introduced in the encoding chain.

Summarizing, the integration of these techniques in modern hybrid video codecs, although promising in terms of artifacts avoidance for low bit rate coding, still shows open issues like the adaptation of the amount of filtering to the operating point or the combination of filtering and prediction. In the second part of this PhD Thesis, a method for reducing artifacts at very low bit rates is proposed, based on the application of pre-filtering techniques to video coding. This proposal aims at solving both pre-filtering drawbacks, by means of a proper selection of the filter strength based on the encoder operating point, and the filtering of prediction residue instead of the original frame pixels, in order to take advantage of motion compensation.

1.4 Document Outline

The rest of the document is organized as follows. First, Chapter 2 summarizes some concepts about video coding, for grounding the rest of this PhD Thesis. Then, a first part of the Thesis, covering Chapter 3, is devoted to the concept of Reconfigurable Video encoding and decoding, aimed at solving some of the cons of video coding standards. The second part of the Thesis, covering Chapters 4 and 5, deals with the problem of video coding at low bit rates and proposes a pre-filtering of the prediction residue for video simplification. Finally, some conclusions are drawn and discussed in Chapter 6.

Chapter 2

Introduction to Video Coding

The man who removes a mountain begins by carrying away small stones.
(Chinese Proverb)

2.1 Introduction

Among other scientific achievements from the latest century, the beginning of the digital era has been probably one of the most important technical improvements. The digital technologies have penetrated directly in different aspects of our lives as well as become of paramount importance as a tool for speeding up the achievements of many sciences. In the field of the audiovisual media, it has been indeed a revolution due to the possibility of working with digitalized versions of audiovisual records. The digitalization is itself a lossy process (the nature of original audiovisual materials is purely analog) but it is carried out normally without any noticeable loss of information by the human perception and entails some obvious advantages. For example, digital information can be easily protected against noise and can be easily replicated. It can be efficiently transmitted, encrypted and, above all, a wide range of powerful digital processing tools are available.

The multimedia applications came to light hand to hand with the development of digital communications and with the awakening of the Internet. In a few years, these applications demanded more and more transmission and processing resources and, although the available bandwidth of networks and the computational power of processors increased monotonically (Nielsen's and Moore's laws respectively), the encoding of multimedia information has been a matter of interest until present as it will be in the future. As the data processing capabilities grow, the complexity of the end-user applications increases and, therefore, there is a need for a larger amount of data to be transmitted. The Moore's law states that the speed of CPU's is doubled in a period of 18 months, while the Nielsen's Law estimates that network connection speeds for end-users double each 24 months. It seems that this difference is compensated by the use of data compression algorithms that take advantage of the extra processing capabilities while reducing the demand of bit rate for sending the data.

Together with the research in multimedia processing and transmission, there was a need for understanding to what extent multimedia applications distort the data from the human observer point of view. This interest led not only to methodologies for quality assessment and comparison, but also to the development of perceptual coding techniques such as the MP3 audio codec.

Probably the most important multimedia applications in these days involve the use of video data and, therefore, the encoding of video sequences is of paramount importance for modern communications. As a research topic, video coding has been recently boosted by the development of video coding standards for guaranteeing the interoperability between different devices. In this Chapter, some details about the history and structure of video coding standards will be summarized and the latest video coding standard, the H.264/AVC standard, will be described in some detail. Finally, some of the methods for measuring or estimating video quality will be discussed in order to constitute a ground for the next chapters.

2.2 Video Coding Standards

In own words of the members of the International Telecommunication Union (ITU), the aim of this Agency of the United Nations is to *allocate global radio spectrum and satellite orbits, develop the technical standards that ensure networks and technologies seamlessly interconnect, and strive to improve access to information and communication technologies to underserved communities worldwide*[1].

Likewise, International Organization for Standardization (ISO), the world's largest developer and publisher of international standards, is a network of the national standards institutes of 163 countries, forming a bridge between public and private sectors. Its interests cover a far broader range of disciplines and topics than those covered by the ITU, but both organizations share the aim to guarantee the interoperability of algorithms and devices, software and hardware, in the field of information technologies and, specifically, in digital video coding.

2.2.1 A Brief History

History of video coding standards begins for the ITU with the development of the first practical digital video coding standard, the H.261 recommendation ratified in November 1988. Its in force version was concluded in March 1993 [ITU-T, 1993]. This standard describes the video coding and decoding methods for the moving picture component of audiovisual services at the rates of $p \cdot 64$kbps, where p is in the range 1 to 30. Although it was preceded by the H.120 codec, the hybrid DCT/DPCM structure of H.261 enabled the use of bitrates as low as 40kbps for videotelephony over ISDN networks.

Only a few years after this milestone, in August 1993, the ISO Moving Pictures Experts Group (MPEG) developed the MPEG-1 standard. Although inspired by the encoding scheme of ITU H.261, the target application of this standard was the transmission of Standard Definition television at bit rates from 1.5Mbps. It is in fact a collection of different standards or layers; among them the most well-known is the

[1] From ITU web page: *http://www.itu.int/*

MPEG-1 audio part and, in particular, the Layer 3, i.e. the MP3 audio format.

In the interest of finding a global solution for the encoding of digital video for different bit rates, ITU-VCEG and ISO-MPEG worked together in the so-called Joint Video Team (JVT) to develop the H.262/MPEG-2 recommendation [ISO/IEC, 1994], improving those MPEG-1 shortcomings such as the lack of interlaced video support or the restriction to encode a maximum of two audio channels (stereo), and maintaining backwards-compatibility. Together with the video and audio parts, published in 1996, the standard also includes an specification of container formats for video transport that are still valid for current Digital Video Broadcasting (DVB) systems. MPEG-2 is probably the most successful video coding standard due to the market penetration of the DVD players.

From this point, the ITU-VCEG centered their efforts in the publication of a new digital video coding standard for low bit rate applications, H.263 [ITU-T, 1995], while ISO-MPEG worked in MPEG-4, an ambitious initiative for covering multiple aspects of a new generation of video coding tools, such as visual oriented video coding by means of textures or objects, video transmission over IP networks or digital rights management. The core of the MPEG-4 part 2 standard, called the Advanced Simple Profile (ASP) video format, was conceived as an evolution of previous coding tools, in the same way that H.263, and therefore, both organizations decided to work together again and publish them as different standards with the same content.

In spite of all the efforts devoted to its development, the complex MPEG-4 standard moved away from the market due to the limited applicability or the lack of commercial interest of some of its parts, and a new standard was conceived. The ITU-H.264 or MPEG-4 part 10 final draft was published in May 2003 as Advanced Video Coding (AVC) and it *was developed in response to a growing need for higher compression of moving pictures for various applications such as digital storage media, television broadcasting, Internet streaming, and real-time audiovisual communication. It is also designed to enable the use of the coded video representation in a flexible manner for a wide variety of network environments*[2].

[2] From summary of H.264 standard [JVT, 2003]

The success of H.264/AVC was certified with its use in popular Blu-ray discs, its penetration in the main applications of video streaming over Internet such as `YouTube` or `iTunes`, or its adoption in the new standards for high definition video broadcast. Later several corrections and extensions of this standard have come to light. The Scalable Video Coding (SVC) standard for heterogeneous environments, or the Multiview Video Coding (MVC) for systems with multiple cameras like in 3D cinema, are only two examples of extensions of H.264/AVC for specific application scenarios.

In the last years, a new effort has been made for developing a new video coding standard for the next generation of applications. The new *High Efficiency Video Coding* (HEVC) standard, also known as H.265, is again a product of the collaboration of the experts groups of ITU-T and ISO in the so-called *Joint Collaborative Team on Video Coding* (JCT-VC). The call for proposals for improving the efficiency of H.264 began in 2004, and a final draft of the HEVC standard is expected to be published in January 2013.

The HEVC standard aims at achieving a reduction of 50% in the output bit rate for an output quality comparable to that of H.264/AVC for the current HD formats. Due to the notable increase in the complexity responsible of the potential bit rate reduction, the core algorithms of the standard have been deliberately oriented to take advantage of paralell processing.

2.2.2 Characteristics of Video Coding Standards

An important feature of these standards is that they specify only the coded representation of digital video, enforcing no particular design of the video encoder itself and, therefore, enabling the optimization of successive generations of video encoder implementations for a better performance. In fact, the normative description of bitstream syntax is the key factor for the interoperability of different video coding implementations and devices. The video encoder can incorporate only a few of the functionalities supported by the standard as soon as it generates a standard-compliant bitstream. On the other hand, there is also a certain freedom in the implementation of the decoder but not in the set of functionalities included, given that it must be capable of decoding any standard-compliant bitstream.

In the interest of this interoperability, video coding standards define a set of profiles and levels. Typically, each codec profile is aimed at a certain application scenario, namely a certain range of bit rate, delay, or complexity constraints imposed by the devices or networks involved. Therefore, a video encoder implementing a particular profile employs a particular subset of video coding tools from those defined in the standard, appropriate for a specific scenario. Levels work in a similar manner by limiting certain video coding parameters in order to comply with complexity constraints.

It is worth to mention that, although video coding standards do not establish a normative implementation of the decoder, they usually specify the test set for assessing the conformance with standard and even reference software implementations of both encoder and decoder, in order to facilitate the design and testing of proprietary video coding products.

2.3 Overview of the H.264/AVC video codec

Probably the most revolutionary step forward in the field of digital video coding since the appearance of MPEG-2, came with the development of the H.264/AVC video coding standard. The epithet *advanced* does justice to this standard that gathers together all the refined methods of the family of hybrid DPCM/DCT block-based video coding in an all-purpose codec that displaced MPEG-2 as the standard for the most popular applications of digital video. The fact that it reduces by about 2 times the output bit rate with respect to MPEG-2 has enabled the use of HD resolutions in storage applications, becoming one of the supported formats in modern Blu-ray discs, and digital TV transmissions, in the framework of DVB recommendations.

This section gives an overall view of the H.264/AVC standard beginning with the hybrid block-based DPCM/DCT video encoding methods, which conform the basis of the modern video codecs from H.262/MPEG-2 to H.264/AVC standards as it will

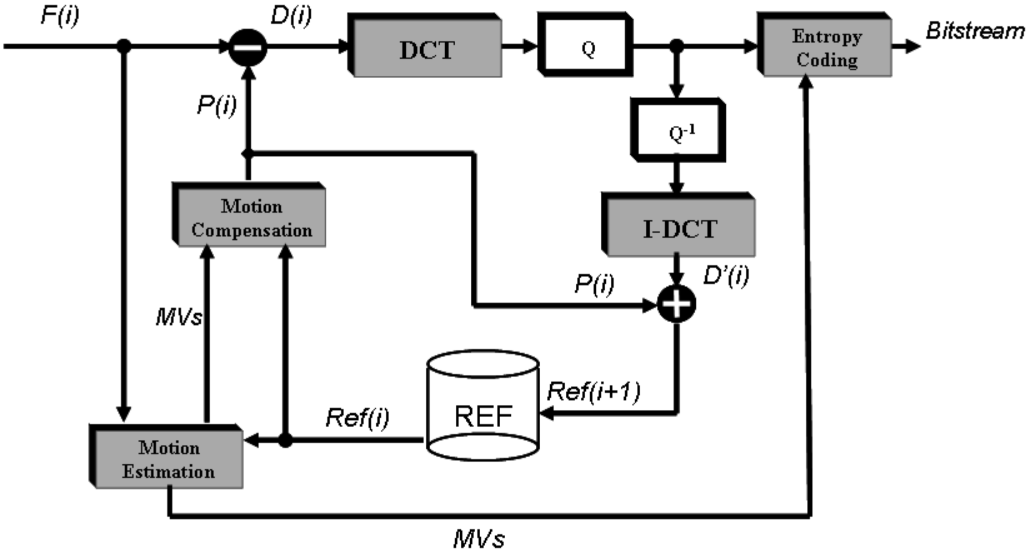


Figure 2.1: Hybrid DPCM/DCT video encoder scheme.

also do in the incoming HEVC standard. A deeper analysis of these tools will be carried out later, concerning the particular case of H.264/AVC.

2.3.1 Hybrid DPCM/DCT Video Coding

The hybrid block-based DPCM/DCT scheme has been widely employed in the latest standards as the core engine for video encoding. This model combines encoding tools of proven efficiency: the algorithm of motion compensation for taking advantage of temporal redundancies in the sequence, and the encoding of prediction residual by means of a spatial model similar to that employed in JPEG, consisting of transform, quantization, and entropy coding stages. The complete hybrid block-based DPCM/DCT encoder scheme is depicted in 2.1.

Each frame is typically partitioned in blocks of 8x8 or 16x16 luminance samples (and their correspondent crominance samples) in order to better take advantage of redundancies, obtaining more accurate predictions and more compact representations that those that could be obtained by processing the entire frame. Additionally, there is a complexity benefit of block processing: the computational cost of calculating the DCT of the entire frame would be unbearable. Encoding with a smaller block size obtains a better block prediction and, therefore, less bits are produced to encode residue but, given that it is necessary to send some side information for each block, a block size lower that 4x4 or even 8x8 is usually of no practical use.

Motion compensation searches in a certain area of a reference frame the best match of current block, i.e., a region of the same size that produces the minimum prediction error. This error is measured as the Sum of Absolute Differences (SAD) or the Sum Square Differences (SSD) between current and predicted block. The encoder reconstructs each previously encoded frame in order to rely the prediction process on the same references as those reconstructed by the decoder. The displacement between the coordinates of the current block and its best match within the frame is the Motion Vector (MV), and it is sent to the decoder together with the difference between each block and its best match (residual) in order to reconstruct the block.

The subsequent transformation accumulates most of the residual energy in a few transform coefficients. Although the optimal transform for a set of blocks in terms of energy compacting can be constructed by means of Principal Component Analysis (PCA), the so-called Karhunen-Loéve Transform (KLT) needs to be trained for a particular set of data and must be sent to the decoder. This overhead is an important drawback for the practical use of this technique for transform coding, as discussed in [Aase et al., 1999]. A good approximation to the optimum performance of the KLT can be obtained from the DCT with almost the same energy compacting results. Additionally, DCT is not data-dependent and its fixed structure enables the use of improved fast implementations. Moreover, the 2D DCT is separable and symmetrical and, therefore, it can be calculated by applying a 1D DCT to the rows and processing the result again by columns with the same 1D DCT. Certain fast implementations of the DCT like the one proposed in [Chen et al., 1977] have been proposed for optimizing the calculations in different platforms.

The next step, the quantization algorithm, reduces the number of representation levels of transform coefficients in order to further reduce the entropy of residual data. To do so, each coefficient value is divided by the quantization step and then truncated to the nearest integer. This truncation operation is responsible of the unrecoverable losses that prevent the decoder from reconstructing an exact copy of the input sequence, but also enables to achieve higher compression ratios.

Finally, the entropy encoder assigns a unique binary representation to the quantized transform coefficients of each block and puts this information together with the side information (motion vectors and other data necessary for reconstructing prediction) in the bitstream. Entropy-based Variable Length Codes (VLCs) such as Huffman or arithmetic codes take advantage of the low-entropy of the output data of the quantization step producing a highly compressed video bitstream.

2.3.2 Video Coding tools in H.264/AVC

Although the core of H.264/AVC is similar to that of previous standards, the so-called hybrid coding scheme, some key improvements in the algorithms involved and in the composition of the standard itself made a qualitative leap in flexibility and performance. For a complete summary of the algorithms involved in the encoding process in H.264/AVC and its differences with respect to MPEG-4, the interested reader is referred to [Richardson, 2003] and [Sullivan, 2004]. The profiles and the tools related to them are listed in Figure 2.2. Baseline profile is aimed at low-delay real-time applications such as video telephony or videoconferencing, while the Main profile contains the set of encoding tools more appropriate for video broadcasting or storage, and the Extended profile could fit better in streaming applications due to its tools for error resilient transmission. Since its publication, some extensions of these initial sets of tools have been added to the H.264/AVC recommendation, incorporating a High Profile defining a set of tools for handling HD content and supporting new video formats such as 4:2:2 or even 4:4:4 color sampling.

As in previous standards, the smaller coding unit is the macroblock. A macroblock is a square region of 16x16 pixels of luminance and its corresponding 8x8 pixels of each crominance (if the color sampling is defined as 4:2:0). Additionally, the H.264/AVC standard defines a intermediate unit between macroblocks and frames, the so-called Slices. Each slice is encoded independently of the other slices in a frame, guaranteeing

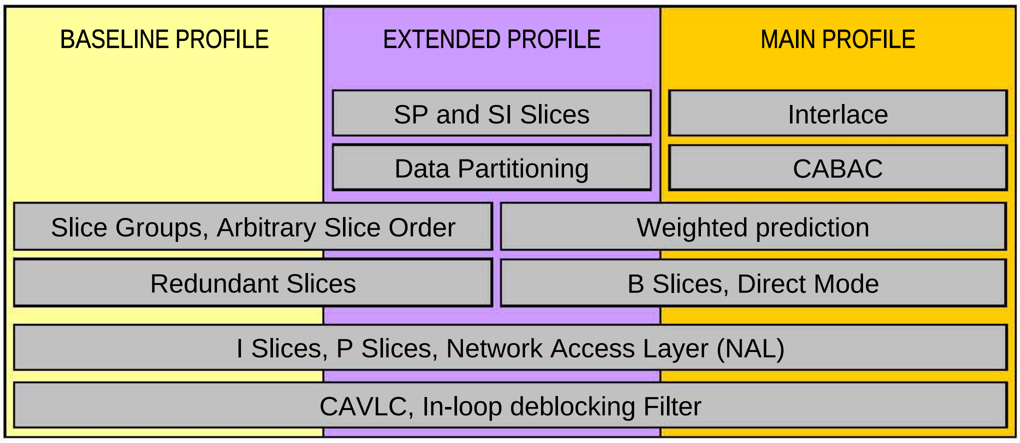


Figure 2.2: H.264/AVC profiles and corresponding tools.

that the appearance of transmission errors in one slice does not harm the rest of slices.

Some of the most interesting contributions of the standard lie in the Intra prediction stage. Spatial redundancies are exploited by the prediction of a block from previously encoded neighboring pixels. To this end, four different modes are defined for the interpolation or extrapolation of 16x16 blocks of luminance pixels and 8x8 crominance components, while nine different modes for partitions of 4x4 luminance pixels are defined. Each mode represents an interpolation direction excepting for the DC, in which the prediction is constructed by averaging all the available neighbor samples.

Concerning the Inter prediction, the mechanism of motion compensation has been also improved by adding more options such as the use of a pool of references containing up to 16 frames, a quarter pixel interpolation or different block sizes (from 16x16 to 4x4) for block-based motion estimation. B pictures are generalized in the main profile and they are no more limited to one past and one future reference, but may use any two references from the reference pool. Moreover, a weighted prediction algorithm can be employed to combine both contributions to conform the prediction.

Concerning the transmission of H.264 encoded video, a Network Access Layer (NAL) is placed as an interface between the video encoder and the network. Additionally, a set of tools are defined for error resilience and recovery for lossy environments.

In Baseline and Extended profiles, multiple redundant slices can be sent for reducing the probability of loosing an entire frame and the Flexible Macroblock Order (FMO) of slices helps to recover from errors. In the Extended profile, a hierarchy of data can also be established for putting more channel coding resources in the most important data, such as side motion information of block modes, of paramount importance for reconstructing a video sequence. Additionally, this profile also enables the use of switching P and I frames for sending multiple descriptions of the same content and dynamically switching between both streams.

Concerning the entropy coding, the Context Adaptive Variable Length Coding (CAVLC) and, above all, the Context Adaptive Binary Arithmetic Coding (CABAC) in Main profile, have contributed to increase compression ratio. They reduce the entropy of the data to be transmitted by adding context information to the encoding process of each symbol (or bit in the case of the CABAC algorithm) and adapting the codebooks to the statistics of the received data.

Finally, some of the most interesting improvements of the H.264/AVC standard will be discussed more in-depth in the following sections; namely: the in-loop deblocking filter, the rate-distortion optimization process, and the rate control.

2.3.3 In-Loop Filtering in H.264/AVC

One of the main improvements of the H.264/AVC standard is the incorporation of a deblocking filter that enhances the quality of the reconstructed image by means of reducing the harmful blocking effects appearing in block-based hybrid video coding. The main novelty of this approach is the integration of the filter inside the decoding loop at the encoder, in order to remove the artifacts from the reconstructed frames to be used as reference. A complete description can be found in [List et al., 2003] and only some notions are included here.

The deblocking filter works directly over the block boundaries in the reconstructed frame in the understanding that reducing the differences in these positions the blocking effect will be less noticeable. The key factor for the good performance of this filter is the adaptivity. Each block boundary is analyzed and a Boundary Strength

Condition	**BS**
One block is Intra and we are in a MB edge	4
One block is Intra	3
One block has coded residuals	2
MVs of blocks differ in at least one pixel	1
Blocks have different references	1
Else	0

Table 2.1: Conditions over block modes and edges for selecting the most appropriate Boundary Strength.

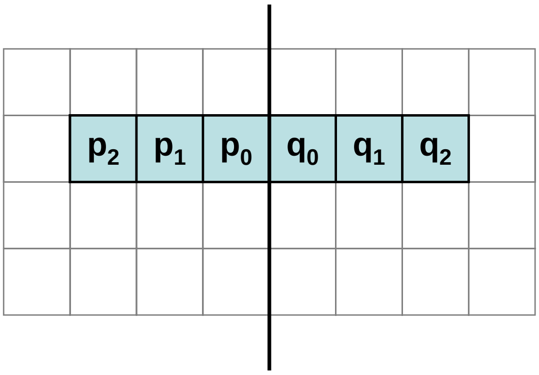


Figure 2.3: Set of samples at both sides of a block boundary for deblocking filter.

(BS) parameter is selected among 5 possible levels, as described in Table 2.1. When BS equals 0, that means that no filter at all is applied, while values 1-3 of BS entail the use of a standard filter, and $BS = 4$ means that a specially strong filter is applied to the current block boundary.

Moreover, a pixel adaptation procedure is carried out in order to distinguish a real image edge from an artificial edge caused by the encoding process. The pixels involved in the analysis and filtering processes for a vertical block boundary are depicted in Figure 2.3. The differences between the pixels at both sides of the block boundary $(p_i - q_i)$ are compared to certain thresholds in order to decide whether the set of pixels is filtered out or not. These thresholds depend on the QP value selected for encoding the blocks and can also be modified by an offset parameter sent within the slice header. The basic deblocking filter for BS values from 1 to 3 corresponds to a weighted average of the boundary pixels $p1$, $p0$, $q0$ and $q1$, while the filtering process when BS equals 4 involves all the set of samples shown in Figure 2.3. The final results exhibit not only an increase in the PSNR of the reconstructed sequence at a certain bit rate, but also a considerable improvement in the subjective quality.

2.3.4 Rate Distortion Optimization in H.264/AVC

The use of a wide range of configurations in H.264/AVC makes possible to adapt the encoding process to many different situations and video contents. Although this adaptivity is the main reason for the notable reduction in the output bit rate, it also implies that it is more difficult to explore the whole range of encoding alternatives.

Even if only two different alternatives were involved, finding the best of them implies a compromise between encoding rate and quality of the reconstructed video sequence: a bit rate reduction comes usually at a cost of higher distortion in the reconstructed frames. Moreover, although the output bit rate is plainly measured in bits, the real bit count can only be measured after the whole process of encoding with a certain configuration. Additionally, quality assessment is neither an easy task: a proper estimate capable of substituting the actual subjective quality assessment is still to be found and it is necessary to employ sub-optimal quality measures derived from pixel-wise differences like the MSE or the PSNR. However, even with a proper definition of quality, a decision criterion is needed which finds the best trade-off between rate and distortion.

In the more general case, this decision criterion can be found by the application of the Rate-Distortion Optimization (RDO) theoretical framework. Given a set of M encoder parameters and a set of N encoding units, the optimum configuration of the encoder $\mathbf{X}$ is found by minimizing a certain distortion measure d under a bit rate constraint R_{max} as follows:

$$\begin{aligned} \mathbf{X} &= \underset{\mathbf{X}}{\operatorname{argmin}} \sum_{i=1}^{N} d\left(i, \mathbf{x}_i\right) && (2.1) \\ s.t \quad & \sum_{i=1}^{N} r\left(i, \mathbf{x}_i\right) < R_{max}, && (2.2) \end{aligned}$$

where $\mathbf{x}_i$ is the M-dimensional vector of encoding parameters for the ith encoding unit, the rows of the configuration matrix $\mathbf{X}$, and affects both the rate rate $r\left(i, \mathbf{x}\right)$ and distortion $d\left(i, \mathbf{x}\right)$.

Given that hybrid video coding is sequential and predictive, the encoding of a unit (for example a macroblock or a complete frame) depends on the encoding configuration of the previous encoding units. The only practical way of solving Equation 2.2 is therefore to consider all the encoding units as independent and to decide the best configuration parameters for each coding unit. However, the rate constraint is still a condition on the global output bit rate. In order to deal with this issue, the optimization problem is reformulated by applying Lagrangian optimization as described in [Ortega and Ramchandran, 1998]:

$$\begin{aligned} \mathbf{X} &= \underset{\mathbf{X}}{\operatorname{argmin}} \sum_{i=1}^{N} J(i), && (2.3) \\ where \quad J(i) &= d(i, \mathbf{x}_i) + \lambda \cdot r(i, \mathbf{x}_i), && (2.4) \end{aligned}$$

where λ is a nonnegative real parameter. A sequence of encoding parameters **X'** that minimizes the Lagrangian cost as defined in 2.4 also minimizes the total distortion under a rate constraint $R'_{max} = \sum_i r\left(i, x'_i\right)$. A different question is how we can find the λ parameter corresponding to the desired constraint R_{max} in a particular practical application. Anyhow, this reformulation helps to solve the problem of configuration decision under the assumption of independent coding units, and the global decision in Equation 2.4 can be divided into individual decisions for each coding unit:

$$\min \left\{ \sum_{i=1}^{N} d(i,\mathbf{x}_i) + \lambda \cdot r(i,\mathbf{x}_i) \right\} = \sum_{i=1}^{N} \min \left\{ d(i,\mathbf{x}_i) + \lambda \cdot r(i,\mathbf{x}_i) \right\}, \tag{2.5}$$

Even considering the huge reduction in the complexity of the decision in Equation 2.5, it is not possible for the great majority of the video encoders to test all the possible combinations of encoding parameters for each coding unit and reconstruct the decoded unit in order to reliably measure rate and distortion. The typical approach lies in performing estimations of these rate and distortion functions that help to delimit the set of configuration parameters.

For example, the decision about the best motion vector is simplified by introducing a non-optimized RD decision based in the Sum of Absolute Differences (SAD) between the original block and the prediction block according to a certain motion vector. The calculation of SAD values does not involve the complete process of encoding and decoding of the block and a huge amount of complexity is saved.

With respect to the particular values of λ for different target bit rates, they can be made dependent on the operation point of the encoder through the QP value, binding the mode decision and the rate control methods. In [Sullivan and Wiegand, 1998], the methodology for obtaining the relationship between λ and QP (or quantizer scale Q) is established and tested for the particular case of H.263 video coding. A representative set of video sequences are encoded with different values of λ and the decision in Equation 2.5 is carried out for obtaining the best Q on average for each λ. This approximation yields to the following relationship in H.263:

$$\lambda_{H263} = 0.85 \cdot (Q_{H263})^2 \tag{2.6}$$

In [Wiegand et al., 2003], the same methodology is applied to the particular case of H.264/AVC obtaining this relationship:

$$\lambda_{H264} = 0.85 \cdot 2^{(Q_{H264}-12)/3} \tag{2.7}$$

2.3.5 Rate Control in H.264/AVC

Although the rate controller is a crucial component of a video encoder, it is a non-normative block in the video coding standards. This block is responsible for finding the best set of encoding parameters guaranteeing that the output bit rate complies with a certain bit rate constraint. The output bit rate of a video encoder depends not only on the encoding parameters, but also on the particular characteristics of the sequence to be encoded. The rate control algorithm adapts to the sequence and decides on the proper set of encoding parameters for the successive coding units.

The output bit rate of a video codec depends on both the characteristics of the sequence to be encoded and the encoding parameters. Some examples of the first category are the video resolution, the frame rate, the sub-sampling of color components and, above all, the temporal and spatial complexity of the video content. On the other hand, some of the parameters affecting the output bit rate are the quantizer, the block size, the encoding modes (Inter, Intra, SKIP...), the GOP size, etc. A control mechanism aimed at modulating the output bit rate should adapt to the characteristics of the sequence by the proper selection of the encoding parameters. Among those, the QP variation is the most common way to control the output bit rate.

Constant Bit Rate scenarios

The most common scenario for a Rate Control (RC) algorithm is the Constant Bit Rate (CBR) scenario, in which a hard constraint is imposed over the encoding output bit rate, usually in Real-Time (RT) applications. The time-variant nature of video information implies that an encoder working at a constant QP produces a variable output bit rate whose oscillations may be unfit for a CBR scenario. In order to solve this, a CBR controller must monitor the produced bit rate and adapt the sequence of QPs assigned to each encoding unit. In the context of the RC algorithms, the term *basic unit* (BU) is employed to describe the minimum encoding unit (frame, slice, macroblock,...) for which the QP may variate according to the directives of the RC.

A decoding buffer is considered to be placed at the decoder side. The decoder receives the video bitstream at a constant bit rate (the one provided by the network) and this buffer enables the decoder to consume as many bits as needed for decoding an entire frame each frame period. The size of this buffer is related to the period of time the decoder waits from the instant of arrival of the first bits until the beginning of the decoding process for the first frame. The larger this period, the bigger the buffer and, correspondingly, the more tolerant the entire system to short-term variations of the encoder output bit rate. For RT applications, it is of paramount importance to prevent the decoder buffer to underflow or overflow. A decoder buffer overflow means that certain data belonging to not yet decoded frames are lost and they will be skipped during the video reproduction. On the other hand, a decoder buffer underflow means that at a certain point the decoder has no available data for decoding and it must freeze the previous frame until the decoder buffer level rises. To this end, a virtual buffer is placed at the encoder side that consist in a mirror image of that at the decoder an helps to control its level.

It is clear that a restrictive scenario such as videoconferencing, which is both real-time and low-delay, may require changing fast enough the QP in order to prevent the buffer to overflow or underflow, and therefore the appropriate BU may be as small as a macroblock. On the other hand, other applications such as storage or streaming may use a frame as the BU without risk for the decoding buffer.

The typical steps in a RC algorithm for CBR scenarios are the following:

1. The risk of overflow and underflow at the decoder is estimated by monitoring a virtual buffer at the encoder side, and the average output bit rate is also measured and compared with the target bit rate.

2. The complexity of the current BU is measured or estimated from previous BUs. The amount of bits produced at a certain QP by a basic unit increases with the spatio-temporal complexity of the unit.

3. According to the information acquired in steps 1 and 2, a bit allocation process is carried out that assigns a bit budget for the current BU or, in some cases, for a set of BUs. It usually assigns a different amount of target bits depending

on the frame type, following the GOP structure. In a lower level (i.e. when a small basic unit size is employed), the bit allocation among BUs is conducted according to the complexity estimated in step 2.

4. For each BU, a Rate-Quantization (R-Q) model determines the proper QP to meet the bit budget and the BU is actually encoded with this QP value.

5. Both the virtual buffer and the coefficients of the R-Q model are updated according to the actual encoded BU size.

The different RC implementations differ basically in the bit allocation procedure and the R-Q model. The most widely used R-Q model is the quadratic model. For example, the H.264/AVC reference software includes an RC algorithm described in [Ma et al., 2003] based on a quadratic relationship between the QP value and the rate, the same model employed in the first versions of the reference software for the HEVC standard. The algorithm also includes different layers (a GOP layer, a frame layer and a basic unit layer) for bit allocation. This hierarchical processing enables to allocate the resources wisely (according with the expected complexity of the corresponding basic units) when dealing with different frame types and regions with different characteristics within the frames. As a complexity measure for Inter frames, this RC employs the Mean of Absolute Differences (MAD) between the original unit and its motion compensated prediction. However, this MAD is only available after the mode decision stage in the encoder, which depends on the λ parameter value as discussed in Section 2.3.4. As described in [Sullivan and Wiegand, 1998], in order to achieve the best trade-off between rate and distortion, the value of λ must depend on the QP value, not decided yet by the rate control algorithm. In order to break this *chicken-and-egg* dilemma, the MAD is estimated from the MAD values of previous frames or, if a small basic unit is employed, from the MAD of the co-located BU in the previous frame.

Nevertheless, the quadratic R-Q model has raised some critics as derived from a Laplacian distribution of the transform coefficients. As discussed in [Kamaci and Altunbasak, 2005], the assumption of a Laplacian or a Gaussian distribution of the source statics does not perform properly for very high or very low bit rates and, therefore, other models have been proposed. For example, in [Kamaci and Altunbasak, 2005] an exponential R-Q model is proposed assuming a Cauchy density distribution of the values of the residual coefficients. The author of this PhD Thesis considers this model particularly interesting and some efforts were made (see [Sanz-Rodríguez et al., 2010]) to enhance this algorithm by providing this RC with a low level basic unit.

Variable Bit Rate scenarios

A totally different matter is the rate control for Variable Bit Rate (VBR) scenarios. In the more general case, a VBR scenario is considered when there is no short-time bit rate constraint to be satisfied by the bitstream at the output of the encoder process. In [Ortega, 2000], a more in-depth classification and definition of the different kinds of VBR applications was proposed, which depends on the particular constraints imposed over the output bit rate. From the point of view of the encoder designer, probably the most interesting of these models is the *constrained* scenario (C-VBR), in which the rate control mechanism embedded in the encoding engine takes advantage of the flexibility in the output bit rate constraints in order to achieve a constant quality in the reconstructed video sequence. It should be noticed that, although the short-term bit rate is not severely constrained in these applications, there are usually other conditions to be obeyed such as an average bit rate for the entire sequence or for a long-term period, as described in [Yokoyama and Ooi, 1999] for storage applications.

In any case, the design of a rate controller for VBR scenarios implies challenges more related to achieving a constant quality than to actually control the output bit rate. To this end, the short-term available bit rate for VBR applications is normally higher than in CBR scenarios. In order to explore the differences between both of them and identify the problems related to VBR coding, a proposal was made in [de Frutos-Lopez et al., 2010] by the author of this PhD Thesis for storage applications. Although there are some interesting open questions in that direction, it is far more complicated to address the problem of a severely constrained scenario, in which the second part of this PhD Thesis in centered.

2.4 Quality Measurement in Video Coding

The assessment of quality for processed digital images and video sequences has been a matter of interest since the adoption of digital video made some previous conceptions about visual quality obsolete. With the development of lossy encoding techniques for visual information, the need for a method to accurately determine the video quality turned out to be obvious. Between its many uses are the evaluation of video coding designs, the test of video communication systems, the quality monitoring in real-time applications or the enhancement of video coding and processing algorithms by means of perceptual considerations.

In the course of this PhD Thesis, different approaches to video quality assessment have been employed to design, tune and validate the algorithms developed. In the following sections a brief introduction to quality and quality measures is given, although biased to the particular quality assessment methods employed in the design and implementation stages of the video coding tools of this Thesis.

For image and video coding, the term quality is usually not identified by means of a property of the image or video sequence itself, but as the Quality of Experience (QoE) of the final user. For example, in a general scenario of image transmission, the quality is obviously related to the absence of noticeable distortions that could make difficult to appreciate image contents, but not all the distortions are the same detrimental to quality appreciation of HVS.

The first consideration to mention is that the human eye and visual system exhibit different sensitivities to distortions depending on the average brightness, the texture, or the contrast level: two distorted images with equal average pixel-wise distortion measurements could generate very different subjective quality opinions depending of the visual context. Second, the visual acuity of each individual has an influence on its perception of distortions and, moreover, even if an average sensitive function is modeled, different users have also a different noise and distortion tolerance related to previous experience (that influences its quality scale). For example, an observer used to watch streaming video sequences over a low bit rate connection has a totally shifted quality scale with respect to that of a professional of video editing.

Finally, the QoE is also related to the interest (or motivation) in the application of each individual. For example, a shop-keeper using a Close Circuit Television (CCTV) system for surveillance could refer a good quality of video signal as soon as it enables to prevent threats or identify intruders, while a movie buff would not tolerate perceiving any artifact or distortion in its favorite film. Even if the final users are not human, there are always some considerations to be made related to the QoE for automated users. For example, in a certain object detection or recognition application, a quality measure of a set of input images should be related to the achievable error rate of the detection/recognition application for that set of images.

There are two main approaches to image and video quality determination. The first one is to actually test the visual content over a set of human observers in order to determine an average quality score. The second method is the use of objective quality indexes that measure different properties of the reconstructed image or video sequence in order to establish a unique estimator of its quality. Video Quality Experts Group (VQEG) *was born from a need to bring together experts in subjective video quality assessment and objective quality measurement* and, apart from some contributions to the design of new subjective testing methods, *it has focused its efforts on the validation of objective measures as estimates of subjective quality, establishing a common framework for testing and comparing new quality measures*[3].

In order to account for the advances in both fields, subjective assessment and objective estimate of quality, next subsections will deal with the pros and cons of both approximations.

2.4.1 Testing Subjective Quality

Although subjective experiments have been conducted for centuries in the field of behavioral sciences, the development of television systems has encouraged the re-

[3] Extracted from the VCEG homepage at: http://www.its.bldrdoc.gov/vqeg/

searchers to study how the Human Visual System (HVS) perceives and interprets the visual information. Some examples are the experiments to validate the Weber's Law in luminance perception, which states how many different gray levels are perceived by an average human, the subjective tests for determining the minimum number of frames per second necessary to experiment no flicker in motion pictures display, and so on.

In video sequences, the need for an automated procedure to obtain representative quality scores from human subjects motivated the publication on 6 April 2008 by ITU-T Study Group 9 of the recommendation *Subjective video quality assessment methods for multimedia applications* [ITU-T, 2008] as their latest exponent. In this document, the set of conditions for subjective experiments such as video display characteristics, distance of the viewer, room illumination, instructions to the viewers, or times of exposure to the stimulus material are clearly stated to obtain a fair quality measure. Different kinds of experiments are proposed in order to help in the *selection of algorithms, the ranking of audiovisual performance and the evaluation of the quality level during an audiovisual connection,* among other purposes. Characteristics of video sequences, recommendations about number of subjects and the methodology for statistical treatment of the data are also included.

In the Absolute Category Rating (ACR), or single stimulus experiment, the test sequences are presented one by one and independently rated on a five- or nine-level category scale. The scale levels are then transformed to an average absolute rating, the Mean Opinion Score (MOS). The recommended qualitative scale is:

5	-	Excellent
4	-	Good
3	-	Fair
2	-	Poor
1	-	Bad

There exist also an option for hiding the reference sequences among the test, the so-called ACR-HR, in order to obtain the relationship between category rating of a sequence and that of its corresponding reference, computing a differential quality score, the Differential Mean Opinion Score (DMOS).

The Degradation Category Rating (DCR) method tries to determine how the test subjects perceive a certain degradation in a processed sequence with respect to a reference. The reference sequence is shown first during around 10 seconds, then comes a small pause and then another 10 seconds of the test sequence. The subject is then asked for a qualitative rating of the impairment between them with this recommended five-level scale:

5	-	Imperceptible
4	-	Perceptible but not annoying
3	-	Slightly annoying
2	-	Annoying
1	-	Very annoying

Finally, the Pair Comparison (PC) method is suggested for applications in which various processed sequences must be compared and none of them is considered as the reference. The processed sequences under test are compared with each other by pairs in all the possible combinations of versions in both possible display orders. For example, if three sequences A, B and C are to be compared, presented pairs would be: AB, AC, BC, BA, CA and CB. For a total of n versions of test sequences, the number of pairs to evaluate is therefore $n \times (n-1)$. As in DCR, the evaluation of each pair involves showing successively one sequence after the other but, for small resolutions such as QCIF or CIF it is possible to show both sequences simultaneously, consequently halving the total duration of the experiment.

According to [ITU-T, 2008], the appropriate method for quality assessment depends on the application and in some cases different methods can be combined. For example, the PC method shows the higher discriminative power when processed sequences are very similar, but it takes a lot of time due to the large amount of sequence pairs to be compared, therefore, when a large number of items need to be evaluated, a first round of tests can be carried out by using the ACR method with a few subjects and a subsequent second round experiment following the PC method can be used for the refinement of those sequences with less differences in MOS in the first round.

The application of any of the mentioned methods, although leading to a fair quality determination for most cases, consumes lots of time and human resources.

Moreover, the use of this recommendation is not suitable in a wide range of situations (i.e. it is not possible to conduct a subjective experiment with 15 or 20 people each time anybody wants to assess the quality of a video sequence) and a simpler and cheaper method for measuring quality is often a must.

2.4.2 Estimating Quality with Objective Measurements

As mentioned in previous section, the subjective quality assessment comes at a high price in terms of time and human resources, and an automated procedure for estimate video quality (or distortion) is needed. A lot of methods for quality assessment have been proposed that following [Wu and Rao, 2005] could be classified into pixel-wise difference measures, methods modeling the HVS or engineering-based quality estimators which combine measurements of different features over video sequences. Nevertheless, the basic categorization of these methods comes from the availability or not of the reference video sequence:

- Full-reference (FR) methods: employ an undegraded reference video sequence in order to compare it with the distorted version and determine the amount of degradation. All the pixel-wise and almost every HVS-based quality measures fall in this category.
- Non-reference (NR) methods: try to determine a quality estimation without any information about the original video sequence.
- Reduced-reference (RR) methods: although the reference is not available, the use of certain features extracted from the original video sequence is allowed, usually obtaining a better estimate than the NR methods.

Probably the most widely employed quality measures due to their simplicity are those derived from averaging the pixel-wise difference between the ith processed frame $I_p(i)$ and the ith original frame $I_{org}(i)$ for the whole sequence. Some of them are listed here:

- Sum of Absolute Differences: $SAD(i) = \sum_{h=1}^{H} \sum_{w=1}^{W} |I_p(i,h,w) - I_{org}(i,h,w)|$

- Mean of Absolute Differences: $MAD(i) = \frac{1}{WH}\sum_{h=1}^{H}\sum_{w=1}^{W}|I_p(i,h,w) - I_{org}(i,h,w)|$
- Sum of Square Differences: $SSD(i) = \sum_{h=1}^{H}\sum_{w=1}^{W}(I_p(i,h,w) - I_{org}(i,h,w))^2$
- Mean Squared Error: $MSE(i) = \frac{1}{WH}\sum_{h=1}^{H}\sum_{w=1}^{W}(I_p(i,h,w) - I_{org}(i,h,w))^2$
- Peak Signal to Noise Ratio: $PSNR(i) = 10log_{10}\frac{255^2}{MSE(i)}$

In these measures, $W \times H$ represents the resolution of the video sequence and 255 is taken as the maximum luminance value allowed for a pixel in the calculus of the PSNR (corresponding to a bit depth of 8 bits per pixel). It is to be noticed that all these pixel-wise measures fall into the FR category given that they need the original video sequence.

Unfortunately, these measures not always coincide with the evaluation of a human expert nor even with that of a naive observer. For example, if a frame is displaced one pixel in any direction, it could produce a poor pixel-wise quality score, while a human observer probably would not notice the distortion. As mentioned before, a possible solution is to incorporate HVS considerations to the quality measurement in order to achieve a better quality estimate. Methods based in HVS modeling actually incorporate a sequence of different models representing well-studied psychophysical phenomena such as color perception, contrast sensitivity or masking effects, even taking into account high complexity models derived from neurobiology studies. The objective is to apply a pixel-wise quality measure such as those described above but weighting the contribution of each pixel by a factor which depends on the distortion visibility in the region according to the HVS model.

For a detailed survey on vision modeling for perceptual video quality assessment the reader is referred to [Winkler, 1999] but, for the purposes of this PhD Thesis, it is enough to list some of their drawbacks:

- Almost every measure based on HVS models is a FR measure due to the need of a reference to estimate the perceived distortion.

- The complexity of these methods is usually too high (above all in those derived from neurobiology) to be practical for some of the uses of video quality assessment.

- Given that many of these methods rely on psychophysics experiments that imply the measure of threshold values (the levels at which certain stimuli begin to be perceived), they do not necessarily perform well in the suprathreshold range, i.e., when the levels are far higher than those of detection and the objective is not to detect but to rate the differences between stimuli levels.

Trying to solve some of these drawbacks, a third family of methods arose from a point of view near the engineering that combine different features such as complexity parameters or particular visual artifacts in the test video sequences to compose a quality estimate. For a practical approach to video quality assessment, it may be assumed a previous knowledge about the kind of distortions to be expected from a certain process. For example, the Digital Video Quality (DVQ) method described in [Watson et al., 2001] is applied to measure quality for DCT-based video coding by means of detecting certain distortions related to processing video sequences in the DCT domain.

Probably the most well-known are those methods that monitor the changes in the structural information of the image, defined as *those attributes that represent the structure of the objects in the image* and measured by detecting local patterns of pixel intensities. Only to mention two examples of this family of algorithms, the extraction of a structural similarity is applied in [Wang et al., 2004a] to calculate a FR quality measure for digital images and in [Wang et al., 2004b] it is employed for FR video quality assessment.

Given that these methods rely on some knowledge of the changes undergone by the video signal during processing, they need a certain calibration process in order to adjust the contributions of the different components to the final quality measure. This calibration process could be viewed as a drawback for a reliable use of these methods for quality assessment in a generic application such as video coding.

Part I

Contributions to Reconfigurable Video Coding

Chapter 3

Reconfigurable Platforms for Video Coding

3.1 Introduction and Motivation

> *Be careful the environment you choose for it will shape you; be careful the friends you choose for you will become like them.* (W. Clement Stone)

As described in Section 2.2, video coding standards play an important role for the interoperability of video coding tools and equipment, establishing a common framework for encoders and decoders, for software developers, hardware designers, and final users. However, they have also some disadvantages.

1. The time-to-market of new ideas and video coding tools may be quite long because of the complicated standardization process. Moreover, high coding rates are achieved in modern video coding standards by means of the combination of specialized tools. Thus, and a new tool designed to obtain a significant improvement in coding rate should choose between achieving a good performance combined with the existing framework or, on the other hand, proposing a new set of related tools for replacing existing framework.

2. The design of a general purpose video codec involves a trade-off between offering

a range of tools wide enough for achieving a good performance in every potential scenario and keeping the codec complexity low. As a consequence of this wide range of tools, in modern standards, some of them are implemented but never used in particular applications.

3. Furthermore, the flexibility achievable with an standard solution is somehow limited and certain contents and application scenarios could benefit from tailored coding functions.

4. It is well known that a better performance in RD terms may be obtained by adapting the video compression algorithm to the particular characteristics of video content. Although recent standards have improved their adaptivity by increasing the number of encoding options, these options and tools are anyway predefined, and consequently, the flexibility is limited.

5. Device manufacturers must increasingly support multiple codecs in their products, rising the costs and the final price of these products. For example, a Bluray player must give support for multiple incompatible codecs like H.264/AVC, MPEG-2, MPEG-4 or VC-1 in the same hardware device.

The ideal solution for the mentioned drawbacks can be obtained by a change in the whole paradigm of video coding, expressed in [Kannangara et al., 2010] as *enabling flexible reconfiguration for efficient decoding of multiple video formats, rapid deployment of new coding algorithms, and dynamic adaptation to suit the video content.* It is necessary to break the assumption of a fixed decoder structure and bitstream syntax and to consider that the only constraint should come from the lack of constraints. This almost philosophical statement will be put on the test in this chapter.

This degree of adaptivity can be accomplished by making a decision at the encoder side about the codec configuration and sending a new decoder definition together with the encoded video information. A new method for defining video decoders is therefore needed according to the following characteristics:

1. Capable of describing the complete functionality and structure of the video decoder.

2. Platform independent.

3. Dynamically executable on different processing platforms.

4. Capable of describing both existing and new formats.

5. Modifiable during a video communication session.

The Decoder Description (DD) obeying these conditions could be sent to the decoder prior to the actual decoding process in order to reconfigure an adaptive decoding engine. In this way, the need for implementing tools that won't be used as part of a standard codec could be conveniently overcome. Moreover, there would be no need for implementing different standards in the same decoding device. In the absence of a fixed structure, the design and test of new self-designed video processing could be done without the concurrence of a standardization process and certain specific applications could benefit from tailored encoding solutions. Finally, the on-the-fly modification of the decoder description could mean a rate-distortion benefit due to the fine-grain adaptation of the encoding tools to video content.

As will be described subsequently, the Reconfigurable Video Coding (RVC), an initiative of the MPEG, faced the challenge of solving some of the problems of video coding standards as described in the previous list. It was originated as a normative method for describing new standard video coding tools and configurations, due to the increasing speed in the research and development of new tools, which was being burdened by the "slow" standardization procedures. At the same time, this initiative had an eye on the issues in hardware development of multi-codec devices. For these reasons, the description of a video decoder in RVC covers mainly the second and third issues of the previous list, while partially addresses the rest of the items, as will be discussed later in this chapter. On the other hand, some experiments carried out by the video coding group at the Robert Gordon University (RGU) demonstrated the importance of the on-the-fly reconfiguration of a video codec during the encoding session, generating a new proposal regarding an alternative fully configurable codec that intended to go beyond RVC. The author of this PhD Thesis spent a research stay at the RGU coinciding with the origin and the development of this innovative proposal.

In this chapter, these latest improvements in reconfigurable coding of video are analyzed. First, the RGU works aimed at demonstrating the potential of reconfigurable codecs are described in Section 3.2 in order to ground the Fully Configurable Video Coding (FCVC) paradigm, that will be described from a theoretical point of view and supported byan experimental proof of concept in Section 3.3. Then, Section 3.4 will be devoted to describe the fundamentals of RVC in order to conclude the chapter with a summary of the pros and cons of both paradigms as well as some possible future lines for the improvement of reconfigurable solutions.

3.2 An exploratory experiment on Dynamic Transform Replacement

> *The pessimist complains about the wind; the optimist expects it to change; the realist adjusts the sails.* (William Arthur Ward)

3.2.1 Rationale of the algorithm

In the light of the successive modern video coding standards were published, it is well known that adaptive solutions in video coding may offer significant benefits. The latest example is H.264/AVC in which, as shown in Section 2.3, complex processing tools are being used and a large amount of parameters can be explored for improving the rate-distortion performance. The next HEVC standard seems to be designed under the same principles of achieving a notable RD gain by means of an improved set of alternative coding tools. The challenge we addressed in [Kannangara et al., 2008] and [Richardson et al., 2008] was to develop a framework for exploiting the benefits of adaptive reconfiguration in video coding without the limitations imposed by the video coding standards.

The main components of the proposed framework are the reconfigurable encoder

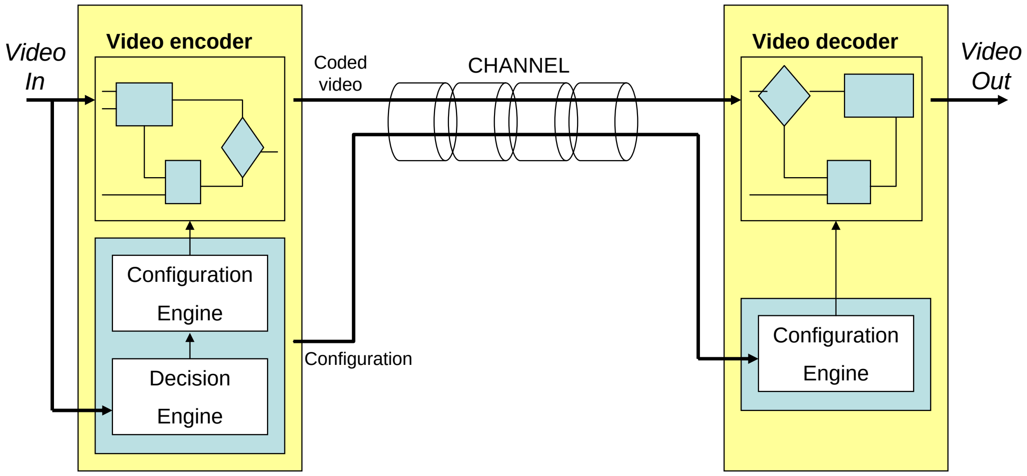


Figure 3.1: Dynamically configurable video coding scenario.

and decoder, capable of being altered at the beginning or during the encoding session. Additionally, a decision engine is responsible for the changes in the structure of the encoder, ideally basing this decision on a rate-distortion criterion, that may require the corresponding update of the decoder. The new reconfiguration must be then communicated to the decoder through the same channel employed by the video content information or a side channel and the decoder must be accordingly adapted for a proper decoding of the video bitstream. The complete system scheme is depicted in Figure 3.1.

In the same way that many previous works in the literature had pointed to the adaptation of the transform block to increase encoding performance (see [Dony and Haykin, 1995], [Effros and Chou, 1995] and [Kamisli and Lim, 2009] for some examples), our first test for the dynamically configurable video coding scenario was the adaptation of the transform within an intra video codec. Although this test was initially run on a non-standard basic Intra codec similar to motion JPEG, later experiments were carried out using an H.264/AVC intra encoder with similar results, as described in [Richardson et al., 2008]. Both works were revisited and extended in [Bystrom et al., 2010].

3.2.2 System Description

The system works as follows. 1)The codec employs initially a DCT for encoding video frames and the decoder is implemented with an inverse DCT for decoding. 2)After the beginning of the encoding process, a decision engine in the encoder is able to dynamically exchange the DCT by a Haar transform; and 3)the corresponding signaling is sent to the decoder. This signaling consists of a reconfiguration command and a description of the new transform to be used, which is not supposedly to be available at the decoder implementation. 4)The decoder implements the new transform and, from this moment, the encoder may employ any of them for processing each block, signaling with a new syntax element in the bitstream whether the DCT or the Haar transform was used.

The experiments carried out with both the JPEG-like and the H.264/AVC codecs are based in the same syntax for describing and transmitting a transform, which will be summarized in the rest of this subsection following [Kannangara et al., 2008].

Syntax for transform reconfiguration

For certain direct and inverse transforms, there exist efficient implementations in the literature described by means of flowgraphs. As a way of an example, an 8-point 1D inverse Haar transform is depicted in Figure 3.2 after being rearranged from an original flowgraph extracted from [Roeser and Jernigan, 1982]. In our proposal, these compact representations are taken as a basis to describe transforms due to its simple structure. These flowgraphs will be encoded and described by means of a predefined syntax as follows:

- The algorithm is limited to the description of separable symmetrical transforms defined in terms of a flowgraph.

- The global structure of each transform graph is defined as a certain number of successive stages with the same number of nodes per stage, as many as the length of the 1D transform.

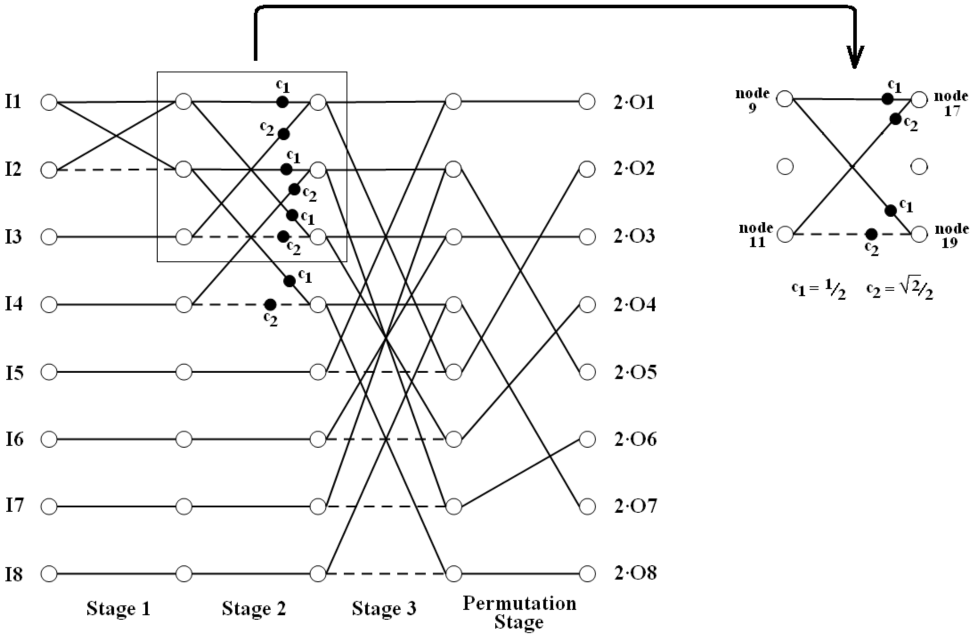


Figure 3.2: Flowgraph for inverse 1D-Haar transform and the detail of a multi-mode (*butterfly*) structure.

- Each node represents an operation between the input data obtained from one or more nodes in the previous stage (it can be limited to two inputs without loss of generality) and produces an output to be sent to one or more nodes in the following stage.

- A considerable reduction in the size of the transform description can be achieved by rearranging the flowgraph in order to connect all the nodes at a given stage with their respective co-located nodes at the following stage. In this way, only one index is to be coded for describing the inputs to a certain node, given that the other input is always the co-located node.

- Nevertheless, reordering the nodes of each stage for the *straight path* assumption, as described in previous paragraph, implies that the outputs of the last stage were probably disordered. In order to solve this, an additional permutation stage is defined which rearranges the flowgraph outputs, as depicted in Figure 3.2.

node_type	**Description**	**Codeword**
1	Identity node	10
2	Addition	1110
3	Weighted addition	1111
4	Butterfly	0
5	Permutation	110

Table 3.1: Node types and Huffman codewords.

- Concerning nodes, each one is described by means of an operation code depending of the kind of operation it represents. The basic operation considered is the addition or subtraction of two weighted inputs. Given that the same weighting factors are repeatedly used in different places in the most commonly employed 1D transforms, a list of *unique numbers* is transmitted in floating point format and referred by means of *unique number indexes* at the moment a weight needs to be communicated.

- In the same way, a global scale factor can be sent for weighting all the outputs, like the factor 2 in Figure 3.2.

- Additionally, for a more compact representation, a butterfly structure is also defined that consists in a special type of multi-node relation, very common in efficient transform implementations such as FFT or Haar, as can be seen in Figure 3.2, where it has been highlighted.

- Finally, the last elements to be described are the identity nodes, i.e. nodes with a direct connection between its input and output, in which all the necessary information can be inferred by sending only the corresponding operation code. Although the use of these nodes could be viewed as wasted resources, they are necessary to preserve the structure of equally-sized node stages.

- A summary of the different node types and their corresponding codewords is shown in Table 3.1.

In this way, all the functionality of the inverse transform is reduced to 1) a general description of the structure of its flowgraph, which consists of the number of stages,

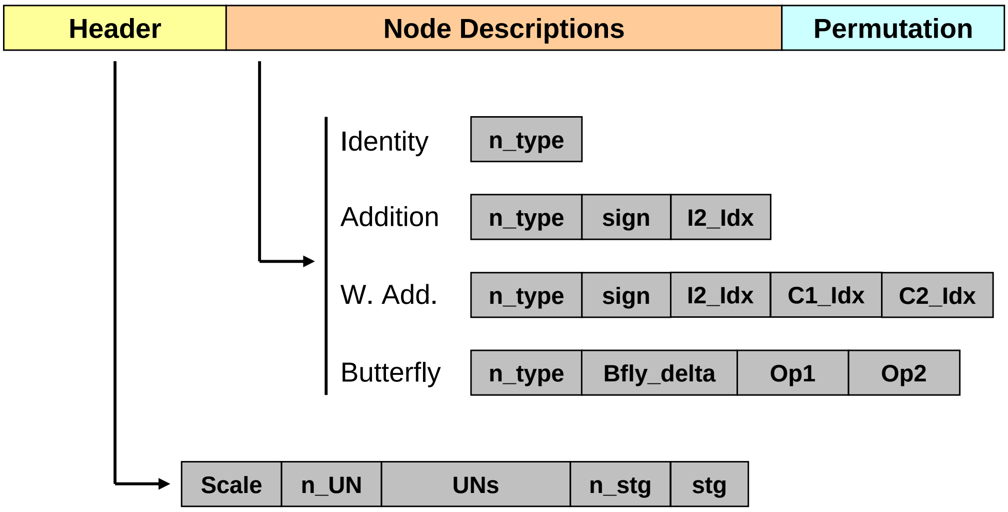


Figure 3.3: Bitstream syntax for coding a transform flowgraph.

Inverse transform	Type bits	Sign bits	Node bits	Factor bits	Header bits	Total bits
DCT	54	52	50	60	80	296
Haar	44	28	33	8	39	152
Hadamard	39	48	40	0	23	150

Table 3.2: 8x8 inverse transform flowgraph bit counts.

the number of nodes per stage, and several constants (the so-called unique numbers) to be used as weights in the transform flowgraph; 2) the sequence of operation codes defining the functionality of nodes; and 3) a permutation stage for rearranging outputs. The complete syntax description is illustrated in Figure 3.3.

Finally, in order to make the transform description syntax as general and compact as possible, different transforms were analyzed and some elements were differentially-encoded or entropy-encoded based on the resulting statistics. For example, the number of unique numbers in the transform description is encoded by using a Rice code, and a Huffman coding is employed for the coding of the distance between the input 1 and the input 2 nodes in a butterfly node (or between the output 1 and the output 2), namely the *Bfly_delta* parameter. The resulting bit counts for the different transforms employed in these experiments are listed in Table 3.2.

Reconfigurable Decoder

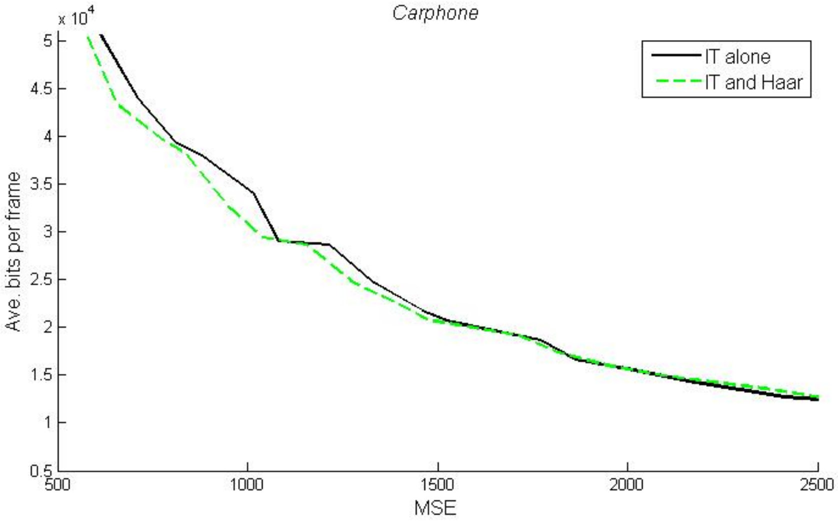
In order to implement a generic inverse transform defined through the syntax described above, a generic inverse transform object is described into the decoder object and when the corresponding inverse transform is received, this object is created and initialized with the received parameters. This inverse transform implementation is far from being optimal in a computational performance sense, as will be shown in the following section.

3.2.3 Results

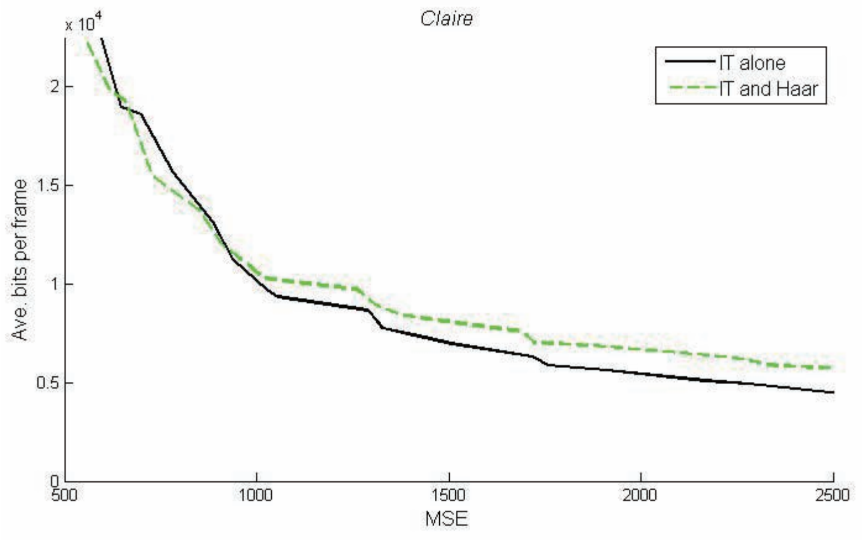
In the experiments described in [Bystrom et al., 2010], a non-reconfigurable implementation of an H.264/AVC codec (referred to as codec A) is tested against multiple-transform implementations. A second non-reconfigurable test codec has been implemented, codec B, that starts the encoding session with both the 4x4 Haar and DCT transforms implemented and enables the use of both of them for encoding each block. Finally, the codec C is a reconfigurable version of codec A capable of updating the decoder by sending a 4x4 inverse Haar transform and use it together with the standard DCT for each subsequent block.

The rate-distortion performance of every codec is tested by encoding 100 frames of different QCIF sequences at 30 fps. As can be seen in the graphs of Figures 3.4 and 3.5, extracted from [Richardson et al., 2008], where results of codec B have been omitted for the sake of clarity, a rate-distortion improvement is achieved from the adaptive transform configuration (codec C) with respect to the single-transform (codec A), especially at higher bit rates, in which the overhead for sending a new transform becomes negligible. A 10-15% of bit rate savings is reported for some sequences at high bit rates. On the other hand, for lower quality, the reconfigurable option seems to be less effective.

Results of codec B (not shown in Figures 3.4 and 3.5) are very similar to that of codec C. This fact confirms that the lower coding performance at low bit rates observed in the graphs of Figures 3.4 and 3.5 is mainly due to the overhead of sending

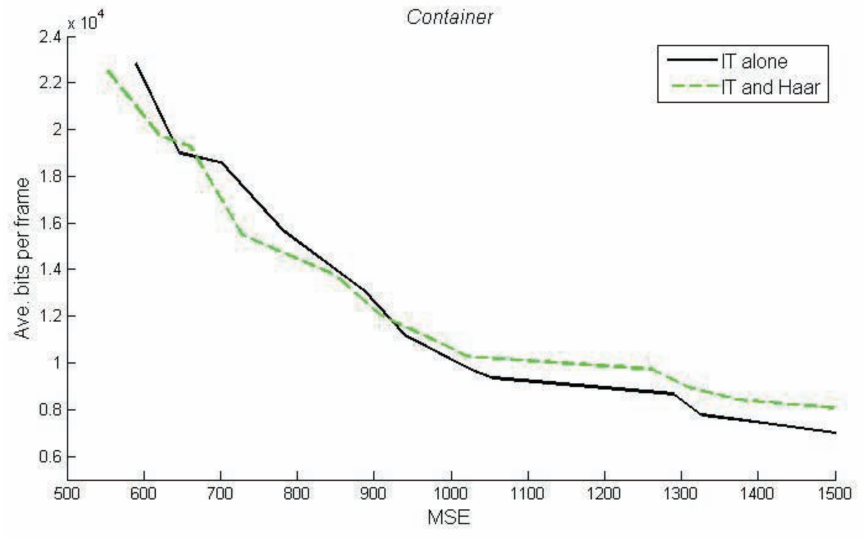


(a)

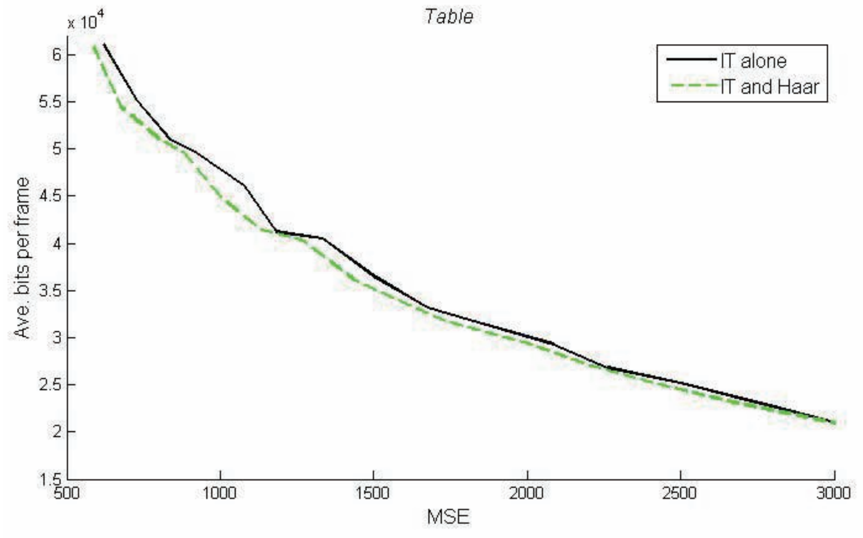


(b)

Figure 3.4: Bit rate (in bits per frame) generated by the encoder vs. the observed MSE for fixed (IT alone) and adaptive (IT and Haar) transforms, for sequences (a) *Carphone* and (b) *Claire*. (From [Richardson et al., 2008])



(a)



(b)

Figure 3.5: Bit rate (in bits per frame) generated by the encoder vs. the observed MSE for fixed (IT alone) and adaptive (IT and Haar) transforms, for sequences (a) *Container* and (b) *Tabletennis*. (From [Richardson et al., 2008])

a flag, common to codec B and codec C, that signals the transform used for each block. On the contrary, transmitting a new transform with the proposed syntax does not imply a significant overhead, as can be seen in Table 3.2 for different 8x8 transforms. Paying only 152 bits for sending the 1D 8x8 inverse Haar transform, 150 for the 8x8 Hadamard, and 296 for the 8x8 DCT would be a low price for the flexibility of codec C, which has no need of knowing the transform or transforms to be used in advance. In fact, the 4x4 Haar transform employed for drawing the graphs in Figures 3.4 and 3.5 entails an overhead of only 80 bits.

On the other hand, the complexity of the reconfigurable decoder in terms of processing resources is increased by a factor of more than 2 when compared to non-reconfigurable decoders. This poor result in terms of complexity is explained by the design of the reconfiguration process at the decoder, where the method for executing new transforms is adapted to the particular reconfiguration syntax and not optimized for a fast execution.

3.2.4 Conclusions

This work clearly demonstrates that there is a considerable potential benefit in the flexible reconfiguration of a video codec with a modest increase of the computational complexity at the decoder. Nevertheless, it states the basis for the definition of the FCVC paradigm to be described in next section, just after emphasizing some considerations that will give rise to some of the main components of the FCVC framework:

- The syntax for transmitting a new tool, namely the flow graph-coding in this approach, is encapsulated as a part of a more general syntax that actually signals the changes in the decoder structure.

- Working with image transforms for reconfigurable coding has lead to a interesting procedure for communicating functionality to the decoder that may be generalized:

1. A flowgraph is defined for describing the functionality of a 1D-transform as a high-level representation.
2. The flowgraph describing a transform is redefined taking its general structure and the functional description of its parts separately. Or generally speaking, a high-level description of the decoding tool is parsed into a decoding description syntax, which includes a low level description of the components of the tool and the way they are interconnected.
3. The transform description is optimized and its elements are entropy-encoded for transmitting a compact representation.

- Together with the description of the new tools, the encoded video bitstream syntax may also be changed during the encoding session.

3.3 Fully Configurable Video Coding

Hell, there are no rules here - we're trying to accomplish something.
(Thomas A. Edison)

The framework for Fully Configurable Video Coding (FCVC) was proposed as a generalization of the structure introduced in the previous section. An in-depth description can be found in subsequent works (see [Kannangara et al., 2010]). In Figure 3.6, a complete representation of the proposed framework is shown together with the different stages involved in the encoding-decoding reconfiguration process.

3.3.1 Overview of the FCVC platform

As can be seen in Figure 3.6, a decision device at the encoder side may adapt the encoder structure, e.g. obeying rate-distortion criteria, and communicate the updated decoding procedure to the decoder side. In order to accomplish this, the high-level description of the new decoding structure is parsed to a Decoder Description Syntax (DDS), optimized and entropy encoded for overhead reduction prior to its transmission together with the video bitstream. This DDS is capable of being instantiated

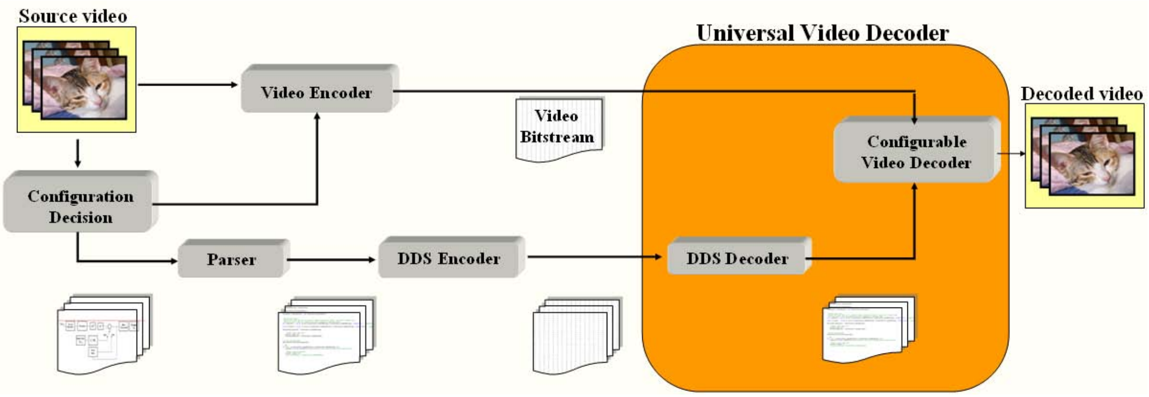


Figure 3.6: The FCVC framework.

by a Universal Video Decoder (UVD). In the next sub-sections, each component is described for unraveling the complete system operation.

Video Encoder Configuration

Similarly to the definition of the traditional video coding standards, the particular functionalities of the encoder engine are out of the scope of the FCVC paradigm, leaving its design and implementation to the codec designer. In any case, it is enough to say that the underlying DDS supports any degree of adaptivity for the configuration of the video decoder (to follow the encoder) between the following extremes:

- Long-term reconfiguration: it entails the complete definition of the decoder engine structure and the functionality of its components. Typically, the decoder description is sent once and employed during the entire encoding session.

- Short-term reconfiguration: small changes in the decoder configuration. Typically it entails fine-grain updates in the decoder structure or in some of its functions in order to achieve a better adaptation to video content.

Again, although it is also out of the scope of the present PhD Thesis to describe the codec reconfiguration decision, it would be necessary to establish a delicate compromise between different figures of merit:

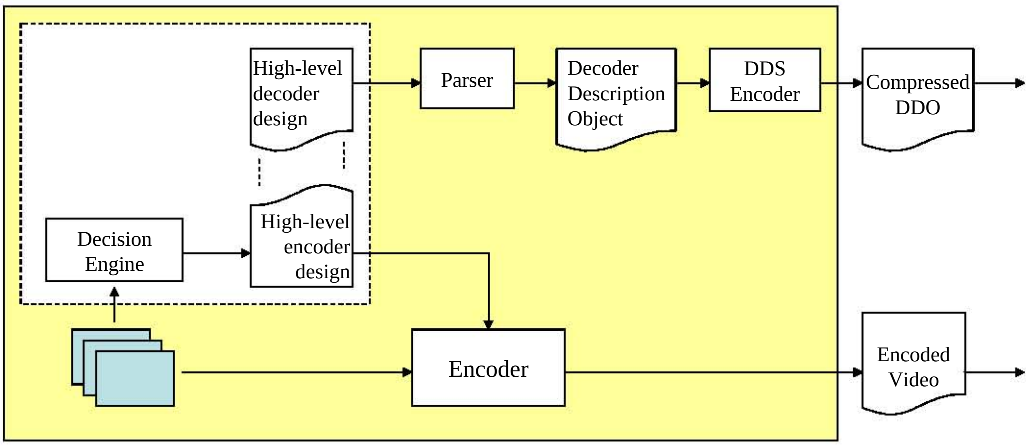

High-level decoder design
Parser
Decoder Description Object
DDS Encoder
Compressed DDO
Decision Engine
High-level encoder design
Encoder
Encoded Video

Figure 3.7: Process for encoding and transmitting decoder descriptions.

- Distortion of the reconstructed video. Ideally it should be assessed by using a subjective test.

- Total bit rate: considering both the encoded video data and the reconfiguration information overhead.

- Complexity in both encoder and decoder sides.

- Delay to instantiate a new decoder description at the decoder side.

As described in Section 2.3.4, making proper decisions to take advantage of the flexibility of a modern video codec such as H.264/AVC is already a major problem. Moreover, if a potentially non-restricted structure was to be considered in the codec, the amount of options to be evaluated could rise to an unreachable number. Does this mean that a reconfigurable video encoder has no practical use? The answer is provided by the experiments carried out in the previous section: a simple implementation of a reconfigurable encoder can take advantage of the flexibility allowed by FCVC. In general, a previous knowledge of the nature of the video sequences to be encoded in a certain application should help to determine which encoding tools are more likely to influence the performance, and in that case only a certain number of possible encoding options should be explored.

Decoder Description Syntax

As mentioned before, FCVC aims at describing the decoder functionality by means of a certain platform-independent DDS, capable of being executed dynamically in any platform. The proposal for an adaptive transform encoding described in Section 3.2 highlighted the relevance of a proper design of the DDS as well as the convenience of partitioning the decoder description process in three successive stages: 1) implementation of new functionality in a high-level language; 2) parsing the high-level definition into a Decoder Description Object (DDO), described with DDS; and 3) efficiently encoding the DDO for reducing the bitstream overhead. The complete description of the proposed syntax will be discussed in-depth later in Section 3.3.2.

Universal Video Decoder

The idea behind the UVD was initially to be run like a virtual machine, interpreting the DDS primitives as an intermediate language and producing the corresponding execution of operations optimized for the particular platform under UVD. As will be described in Section 3.3.3, the first implementation of the UVD consisted in a software-based implementation for the execution of pre-compiled C functions according to the received DDO, but in order to avoid the developing of a particular UVD for each platform, a more general scenario was suggested in which the UVD actually implements a Low-Level Virtual Machine (LLVM) whose output is an intermediate format capable of being linked in platform-dependent assembly code. The UVD is then designed as a combination of object-oriented programming and just-in-time (JIT) compilation. Although the prototype to be introduced in Section 3.3.3 was implemented by means of a set of pre-compiled functions, in the general case a JIT compiler would read the received instructions and create machine code to the particular platform under the UVD.

3.3.2 DDS in detail

Process for Decoder Description

As shown in the summary of the DDO creation and transmission sketched out in Figure 3.7, the first step is the high-level definition of decoder functionality. The complexity of this high-level code depends on the parsing tool used to decompose the instructions into DDS primitives, which is the second step. As the decoding functionality is converted to the common DDS, an additional stage is defined for compressing the decoder description prior to its transmission together with the video bitstream.

As highlighted also by the experiments in Section 3.2, a DDO contains not only the definition of the functionality of each tool involved in the decoding process, but also a description of the interconnection between this tools within the decoder and the actual syntax of the encoded video bitstream. In the adaptive transform example, the structure of the decoder was not altered because only a function, the inverse image transform, is exchanged by another transform in the decoder, but the definition of the new transform was transmitted as well as the changes in the video bitstream syntax for allowing the transform selection by a transform flag, a new syntax element.

It is to be noticed that the procedure for reading and decoding the video bitstream is considered in FCVC as just another function inside the decoder description, subject to the same reconfiguration procedure as the rest of the decoder functionality. As will be shown in next section, this point of view is opposite to the RVC approach, that defines a particular description of the bitstream syntax.

Functions and Functional Processing primitives

In the general approach to the DDS, described in [Philp et al., 2009], the basic unit for reconfiguration is the function. The decoder functionality is divided into a set of basic processing units or functions and their interconnections, namely the function calls. In this way, the high-level definition of the decoder is taken as the starting point for defining the decoder functionality, described by means of a set of low-level platform-independent primitives. Typically, each function consists of a data segment defining variables and constants and a program segment defining the actual processing instructions.

The DDS supports a subset of all the possible data types[1] that suppose a compromise between transmission efficiency and flexibility. In the definition of a function, its input and output parameters are declared (no global parameters or variables are allowed in DDS) as well as the internal variables and constants, conforming a pool of registers internal to the function, classified according to the data types.

Concerning the program segment, in the same way that the inverse transforms of Section 3.2 were decomposed in a reduced set of possible node operations, each generic function in the decoder description is decomposed into some basic operations, covering the most commonly used processing tools such as arithmetic, bitwise and memory management operations, as well as some video processing specific operations. This set of operations was designed also as a compromise between flexibility and performance, and it was initially composed of 63 different operations[2].

Standard Instruction Format

In order to communicate these operations, a Standard Instruction Format (SIF) is defined in Table 3.3. As can be seen, all the operations are supposed to have two inputs and one output, and these data are taken from the list of variables in the function pool of registers by means of variable indexes. Some complex operations, like conditional loops, may require more information to be carried out, occupying more than one instruction slot.

Optimizing DDS

Although there is an important issue in the reconfiguration delay for real-time video coding using FCVC, it is not of fewer importance the overhead caused by sending the DDO together with the bitstream. Given that an overall rate-distortion improvement

[1] For a complete list see http://www.tsc.uc3m.es/~mfrutos/DDS/variables_and_constants.htm

[2] For a complete list of the set of operations see http://www.tsc.uc3m.es/~mfrutos/DDS/

Field	Description
OPCODE	Instruction or operation
TYPE_O	Output type
OUT	Output index
TYPE_I1	Input1 type
IN1	Input 1 index
TYPE_I2	Input2 type
IN2	Input 2 index

Table 3.3: Standard Instruction Format (SIF) for the DDS.

is pursued, the decoder description should be as compact as possible at the time of being transmitted. The standardized format for function definition in the DDS described here was designed to be platform-independent and descriptive enough, but it can be further simplified for transmission purposes by means of inferring operands or operand types in some functions, reducing the number of bits for representing each element according to its valid values, or entropy-encoding the most commonly-used values of each syntax element. Some results of the improvement achieved by this optimization will be discussed in Section 3.3.3.

As previously mentioned, the set of operations defined in the DDS was designed as a compromise between descriptiveness, compactness, and computational efficiency. In this way, the granularity in the definition of the different functional processing primitives guarantees a good performance to be executed in the reconfigurable platform, but this could come at the price of an excessive overhead in the bitstream for certain scenarios. For example, in a videoconferencing application, the available bandwidth is devoted to achieve a real-time point-to-point communication and any overhead could seriously degrade the quality of reconstructed video, while the amount of processing resources available in the decoder is high enough. In this situation, a more compact reconfiguration syntax is preferable even at exchange for an increase in the complexity.

In order to overcome this drawback, some experiments were carried out for optimizing and reducing the DDS to its minimum. In the same way that the optimization for transform reconfiguration described in 3.2, all the operation codes described by the SIF are simplified by reducing the number of bits to transmit each field in the DDS according to these considerations:

- First of all, the number of variables of each type to be used in the function is sent prior to is program segment. A variable pool is allocated for the function and shared by the different variable types. The order in this variable pool follows the same order of appearance of the variables among the program segment. This structure helps to save some bits when the operation codes are sent, as will be show next.
    - The variable pool can be addressed with a `unique_index` for the complete pool when the variable type is not known, or a `type_index`, which corresponds to a reduced index for the subset of the pool corresponding to each variable type.
    - If a variable type can be inferred from the operation, only the `type_index` is sent. For example, the operand representing the condition in a `if` operation must be boolean.
    - Additionally, the first time a variable is used, instead of sending its index, only the variable type is sent and the index is automatically selected according to the first available (not yet used) variable index in the pool corresponding to its type. To this end, a register of the already-used pool slots has to be managed in both the encoder and decoder.
    - Finally, when using a variable that was previously used, the `unique_index` is sent and, given that the decoder already has the information about it, there is no need of sending the variable type.
- Vector data are treated in the same way as normal variables except for the initialization procedure, in which a run-level code is used for transmitting constant vectors.
- Matrix variables are considered vectors and an additional syntax element is transmitted for defining the row size to reconstruct the bi-dimensional structure

when needed.

- For many operations in which the type of an operand can not be inferred, the number of possible operand types is less than the total number of data types, enabling some bit savings in sending the variable types syntax elements. For example, for operations restricted to single variable types (excluding vectors or matrices), only 5 data types are possible and 3 bits are needed.

Summary

Summarizing, with the same process for converting high-level functions into a hierarchical series of functions defined by means of single operations, new decoder configurations or totally new decoding tools as well as the corresponding changes in the functions related to the video bitstream syntax, can be transmitted to the UVD. Moreover, in order to encapsulate and signal the decoder reconfiguration, the DDS also provides with a transmission syntax consisting on DDS commands that specifically allow us to `create_function`, `modify_function` or `delete_function`. Function replacements are carried out by means of a `create_function` command with a function ID identical to that of an existing function as an argument.

3.3.3 FCVC Prototype and Results

A proof of concept of the abovementioned proposal was finally developed and it is described in [Richardson et al., 2009b] and [Richardson et al., 2009a]. The real-time UVD software prototype was developed using C++ pre-compiled functions representing the different opcodes in the DDS syntax. After receiving the DDS, each function is represented by a C++ object with its particular parameters and program memory. The UVD acts then as a virtual processor, making use of these objects and interconnecting the different inputs and outputs between them.

This UVD was encapsulated in a real-time coding environment developed using the *Microsoft DirectShow* framework. At the beginning of the encoding process, the complete decoder description is sent to the decoder by means of the DDS syntax,

Configuration	DDS Size bytes	Compressed DDS Size (bytes)	Reduction
dct.dds	734	372	50%
decoder_dct.dds	2874	1351	53%
haar.dds	465	218	53%
decoder_haar.dds	2605	1195	54%

Table 3.4: Bit counts for different DDOs.

the decoder receives the DDO and instantiates the UVD and uses it to decode the received video. During the encoding process, a reconfiguration may take place once per frame and the corresponding commands for modifying functions or sending new ones are employed. The UVD receives the new DDO and instantiates the decoder, proceeding subsequently with the decoding process again.

The tested scenario consists in a basic intra codec like the one described in Section 3.2, with transform, quantization, and Exp-Golomb entropy coding. The decision engine has the possibility of describing and sending a new transform (Haar) for substituting the default DCT. The results in terms of rate and distortion are practically the same as those depicted in Figures 3.4 and 3.5, changing only in this new approach the particular implementation of the generic DDS, which slightly increases the amount of data to be sent to the decoder. This overhead is listed in Table 3.4.

Obviously, as predicted by the experiments carried out in Section 3.2, the reconfigurable decoding engine is slower than a non-reconfigurable one, achieving a frame rate of 13 frames per second on an Intel Core II Duo at 1.8GHz. This is approximately 5 times slower than a fixed pre-compiled software decoder. The maximum reconfiguration delay is approximately 1 ms (about 1/20 of a frame time), but the typical values are around 0.1 ms.

3.3.4 Challenges in the development of FCVC

As demonstrated in previous sections, the FCVC paradigm involves a novel concept in the field of reconfigurable video coding by extending some previous ideas about adaptivity from other fields of signal processing to the problem of video coding. The main contribution of this approach is the design of a syntax for defining, transmitting, and instantiating new decoding tools in order to provide a framework that allows us 1) to conform a decoder only with those tools needed for a particular application; 2) to take advantage of tailored solutions for particular sequences; or 3) to change on-the-fly the video coding tools for a fine-grain adaptivity of the codec. Moreover, a new field for defining new non-standard tools and complete codecs is open for a rapid development of new techniques.

Both in the literature and in our own successive experiments, it has been proved that the use of adaptive solutions produces a notable rate-distortion improvement. Although modern standard codecs are designed with a certain (high) degree of adaptivity, they are limited by a pre-defined syntax and many of their tools are selected or designed according to the average benefit for a variety of sequences, putting some limits to adaptivity. On the other hand, solutions based in FCVC are not limited by a predefined syntax, but unrestricted to evolve the codec according to the particular video content. This new approach opens up a number of research challenges, including (but not limited to) the following:

- The abovementioned tests consist in the use of manually-parsed functions for encoder configuration. Developing new approaches to automatically generating and optimizing DDS could increase RD performance of reconfigurable coding.
- In order to ensure successful DDS implementation on resource-constrained platforms and prevent the execution of "malicious" DDS code, some limits must be imposed to the DDS execution environment in terms of memory size, file accesses, and processing resources.
- It is necessary to increase the computational efficiency of the UVD/DDS on software platforms to approach that of conventional, non-reconfigurable decoder designs. Possible methods might include efficient handling of data structures such as arrays and blocks, just-in-time compilation, etc. Some discussion about this topic will be carried out later.
- In order to grant random access to a bitstream encoded with reconfiguration information in FCVC, the complete decoder description must be sent from time

to time in order to refresh the decoder structure in case previous configuration codes were not processed. Another option is to determine a reconfiguration period after which the decoder resets to its default configuration and changes need to be sent again. Of course, these options generate an additional overhead to consider.

- UVD/DDS should be efficiently implemented on a wide range of software and hardware platforms, for example through hardware/software co-design techniques for reconfigurable hardware platforms.

3.3.5 Summary of contributions of the author

The development of the FCVC platform and prototype was the result of the efforts of the Centre for Video Communications (CVC) at the Robert Gordon University, under the direction of Prof. Iain G. Richardson. The author of this PhD Thesis enjoyed a research stay in the CVC labs and, during and after this stay, continued collaborating in the many discussions and experiments for making the FCVC proposal as competitive and attractive as possible.

Firstly, in the exploratory experiment of adaptive-transform coding, the syntax for transmitting an inverse transform flowgraph definition, as described in Section 3.2.2, was an original contribution of the author to the prototype. In the same way, as the different parts of the FCVC platform were developed, the author collaborated in the DDS definition and the description of a protocol for efficient transmission of reconfiguration information.

3.4 Reconfigurable Video Coding

The MPEG-RVC initiative has recently developed two new standards, known as MPEG-B part 4 and MPEG-C part 4, published with latest amendments in 2011 (see [RVC, 2011a] and [RVC, 2011b], respectively). These standards aim at addressing some of the issues related to modern video coding and to video codec standard-

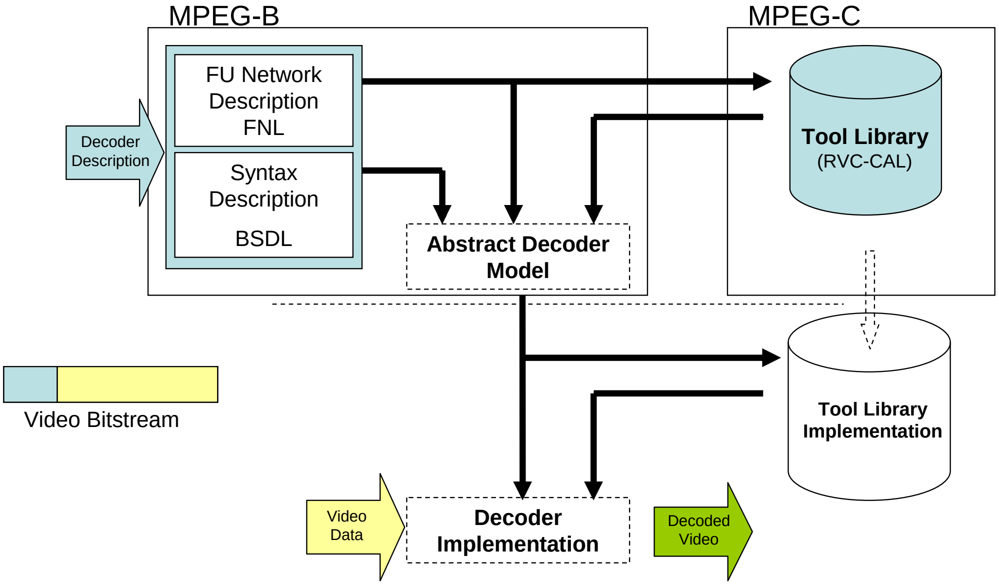


Figure 3.8: The RVC framework.

ization mentioned in Section 3.1, by providing a framework for the development and integration of future video coding tools in a faster and more agile way.

The RVC initiative was inspired by previous works such as [Avaro et al., 1997], which supposed a revolutionary approach to flexible decoding in MPEG-4. Within this framework and as part of the MPEG-4 standard, a certain interaction between user and decoder device, and between decoder and encoder, was assumed, and new video coding algorithms or profiles (combinations of existing tools) could be signaled to the decoder within the bitstream, providing a mechanism for codec evolution. The so-called flexible decoders would receive this reconfiguration information in the form of an intermediate language to be interpreted by a virtual machine together with a syntax description of the actual encoded video data.

RVC proceeds in a similar way by incorporating some new tools and procedures designed for the implementation of adaptive software for video coding. The complete system proposed by the RVC initiative is depicted in Figure 3.8 and outlined in the following sections, but for a more comprehensive description the interested reader is referred to [Mattavelli et al., 2010].

3.4.1 RVC System description

The first of the mentioned RVC standards, the MPEG-B part 4, introduces the framework of the system as depicted in Figure 3.8. The basic idea of the RVC framework is to associate a decoder description with the encoded video bitstream and send it to a compatible reconfigurable device capable of instantiating the received and interpreted Abstract Decoder Model (ADM). In order to facilitate the decoder description for the construction of the ADM, every codec functionality is expressed as a combination of single Functional Units (FU), describing simple video coding tools and algorithms that will be stored in a common Video Tool Library (VTL), a normative part of RVC described in the second of the RVC standards ([RVC, 2011b]).

The Functional Units and the Video Tool Library

According to [Mattavelli et al., 2010], the FUs are chosen in the RVC framework *not too large that they would not allow their reuse in the wide spectrum of codec configurations, and not with a granularity too fine that the resulting number of modules in the library was too large for an efficient and practical reconfiguration process at the codec implementation side.* A forward DCT transform, the zig-zag ordering of transform coefficients or the Huffman encoding, could be examples of individual FUs, as well as simple mathematical functions such as the absolute value.

The structure of each FU is described by means of the so-called RVC-CAL, specified in the MPEG-B standard as a subset of the dataflow language CAL (Cal Actor Language). CAL was designed at the University of California at Berkeley as part of the Ptolemy project and it was aimed at dataflow-oriented systems (a complete report of the original CAL can be found in [Eker and Janneck, 2003]). According to CAL, systems consist of different interconnected units, namely actors, that interact with each other by means of sending and receiving *tokens*. An actor receives sequences of tokens from its input ports and sometimes produces tokens in its output ports. Each computation step is called a *firing*, which happens in some actor *state* and implies the following results: a prefix of the tokens received in the actor's *input ports* (or maybe all of them) are consumed, the state may change, and some tokens may be produced in the *output ports*.

This method for describing functionality is particularly well suited for signal processing applications and enables the VTL to be implemented in a wide range of different platforms, including uniprocessor systems, FPGAs or multi-core CPUs. Moreover, processes in an actor network are executed concurrently, making possible the implementation of complex systems defined using CAL over multi-core platforms.

The complete VTL is specified in the MPEG-C standard and includes the definition of the processes related to some of the most commonly used MPEG codecs. At the time of writing of this PhD Thesis it actually includes: the MPEG-2 Main Profile, the MPEG-4 Part 2 Simple and Advanced Simple profiles, the H.264/AVC constrained Baseline and High (FREXT) profiles, and the MPEG-4 AVC Scalable profile.

In a similar way, new encoding tools can be described in RVC-CAL and included in a non-standard library for being used together with those in the VTL for the implementation of decoding solutions. As a way of an example, in [Ding et al., 2009] the similarities between the Chinese Audio and Video coding Standard (AVS) and the H.264/AVC standard are exploited for implementing a reconfigurable device capable of being instantiated as an H.264/AVC decoder, an AVS decoder, or as a hybrid decoder with a mixture of tools from both standards. Moreover, the standard also specifies a process for evaluating video coding tools for their potential incorporation to the VTL, with the purpose of reducing the time-to-market of new ideas.

It is to be noticed that the actual implementation of the functionalities in the VTL is not included in the standard, in order to take advantage of those resources in the particular software/hardware platform.

The Abstract Decoder Model

As shown in Figure 3.8, the ADM is obtained by combining a video Bitstream Syntax Description (BSD), specified in the RVC Bitstream Syntax Description Language (BSDL), with the description of the network of Functional Units (FU), defined in the corresponding FU Network Language (FNL). Both the BSDL and the FNL are specified in the MPEG-B part 4 standard, together with the RVC-CAL for the description of the functionality of each FU.

Reconfigurable Decoder

Although the reconfigurable device for instantiating a decoder for an RVC bitstream is non-normative, different proposals of Virtual Machines have been made for translating the standard definition of the ADM to a particular implementation without loosing the features inherent to an RVC description: portability, scalability and reconfigurability. We find of particular interest the approach described in [Gorin et al., 2011a], in which a novel adaptive decoding engine is proposed based on the LLVM infrastructure[3]. In this proposal, the RVC-CAL description of coding tools is transformed into a LLVM Intermediate Representation (IR) to be dynamically instantiated to create decoders. Some details of this approach will be discussed in Section 3.5.

3.4.2 RVC achievements

Although some parts of the RVC standards are under development and there is still a long way to go, at least from a theoretical point of view, the RVC approach supposes a great improvement in the search of adaptive solutions for the heterogeneous scenario of the video coding applications. These are the main contributions of RVC for solving the drawbacks of video coding standards listed in Section 3.1:

1. Instead of using different decoder implementations for different incompatible encoding standards in a hardware device, only an RVC decoder implementation is needed that takes advantage of the redundancies between coding tools in the different codecs for saving resources

[3] Formerly known as Low Level Virtual Machine, was released under Open Source License by the University of Illinois. It was originally developed as a research platform for dynamic programming. It can be used for optimizing and static compiling as with GCC, or for Just-in-time compilation of machine code in a way similar to Java

2. The framework permits the use of non-standard tools together with those defined in the VTL, facilitating the development of proprietary solutions.

3. If a video coding tool defined in a video coding standard is not necessary or counterproductive for encoding a certain class of sequences or for a certain application, RVC allows the inclusion of the corresponding changes for avoiding it, with the resulting savings in decoding resources and bitstream syntax elements.

4. In the same way, the reconfiguration capabilities or the RVC framework enable to create new profiles that combine the tools present at a certain codec in a more flexible manner, going beyond the constraints of predefined profiles.

5. RVC includes a new method for testing new tools and incorporating them to the VTL. It supposes a *de facto* standardization process, which entails a period of time considerably shorter than the development of a new standard in the traditional way.

Concerning the drawbacks of the RVC initiative, some of them will be discussed in the next section in the context of summarizing the differences between RVC and FCVC.

3.5 Conclusions

The video coding standards have played an important role in the development of video communications given that they guarantee the interoperability between different devices and implementations in the heterogeneous scenario of modern multimedia applications. On the other hand, as highlighted by the latest improvements in video processing and coding, the standardization procedure has revealed itself as a burden for the development of new ideas and for the fast and efficient implementation of different alternatives for decoding devices.

Two different ways for facilitating the development of new ideas that, otherwise, would entail longer time-to-market, have been described in this chapter. On the one hand, the MPEG-RVC initiative provides the codec with reconfigurability capabilities by defining a syntax for the description of new decoder configurations, expressed in an intermediate language as the interconnection of different standard or proprietary video tools extracted from a pool of functional units. On the other hand, the FCVC paradigm aims to provide with a framework for a free configuration of the codec by means of a syntax supporting the description of any standard or non-standard functionality and a universal video decoder for instantiating any configuration. Although these brief definitions highlight the main difference between both approaches, in the rest of this chapter an in-depth analysis will be carried out of the pros and cons of the reconfigurable platforms for video coding.

3.5.1 Differences between RVC and FCVC

With the purpose of putting in context the differences between both approaches, the discussion concerning these differences starts from a brief remainder of their common points. Given the contemporary origins of both paradigms, some strong influences of the RVC initiative are to be observed in some elements of FCVC framework, and certain changes in the initial objectives of MPEG-RVC could have been influenced by the experiments and proposals executed as part of FCVC. In any case, an in-depth comparison of both methods is summarized in the following items (we label the differences as pros and cons from the FCVC point of view):

- Video Tool Library *vs.* open reconfiguration: The use of a standardized VTL enables to easily deploy standard codecs, to define new profiles with the tools of a particular standard, or even to combine tools from different codecs to conform a new codec, in the same way as the definition of a new decoder is enabled through the DDS in a FCVC platform. However, there are some key differences between both approaches to the problem:
  - (*pro*) The use of a standardized VTL could still be a burden to the time-to-market of new coding solutions, given that new tools must undergo an standardization process that, although shorter than the traditional process

for generating a new standard, is still a considerable bottleneck. Moreover, since each new FU is added to the VTL, the new VTL version must be disseminated to every decoder and many different versions of the standard VTL may coexist causing incompatibility problems.

  – (*con*) The FCVC approach is based on the total freedom of the encoder engine to design the functionality of the decoder. This implies that the decoder engine has almost no control over the functions to be instantiated, and there is a risk of "malicious" behavior or malfunction. This risk is practically avoided by using a standard VTL.
  – (*pro*) Although RVC permits to define non-standard tools, there is still a kind of inertia in the design of a new codec, in the sense that new tools have a handicap when tested against those in the VTL, defined previously for working together in video coding standards.
  – (*con*) The use of a standardized set of building blocks permits to design efficient implementations of these tools in the particular platforms where the RVC decoder is to be instantiated. This is of particular interest when managing hardware implementations. A more flexible approach to functionality description makes not possible to anticipate this kind of optimizations in advance.
  – (*pro*) As demonstrated in the experiments on the FCVC platform, fine-grain reconfigurability enables to obtain a RD improvement by the short-term adaptation to video content. Such an improvement is not obtained from a combination of pre-defined tools under the RVC platform.

- (*pro*) Dynamic Configuration: Although some improvements have been made in the latest versions of the RVC standards, the first documents about RVC did not consider the possibility of sending a new decoder configuration during the encoding session. The idea was the simple communication of the decoder configuration as an initial header to be read prior to the actual decoding of the video bitstream. At the time of writing this PhD. Thesis, some mechanisms for the transport of codec descriptions are under evaluation in order to endow the RVC platform with the possibility of dynamic update of codec configuration. One of the options to be assessed could be the proposal of the RGU group for translating the RVC decoder description into the FCVC-DDS for using it as low-level transport layer, as described in [Richardson et al., 2009c].

- (*con*) BSDL: The video bitstream syntax in RVC is communicated to the decoder separately from the decoder description, while FCVC makes no difference between those functions devoted to bitstream reading and those for video processing, and the need for two different languages is avoided. Although it is not a crucial difference, with the RVC-BSDL it is possible to easily differentiate between both segments enabling Unequal Error Protection (UEP), for example, if it were needed.

- (*con*) Decoding engine: The UVD in FCVC consists in a virtual machine implementation that makes use of a DDS decoder and a JIT compiler to translate the received bytecodes into a function prototype to be compiled to machine code. Therefore, the UVD would have a different implementation depending on the underlying platform. This drawback could be overcome by developing the UVD in a low-level virtual machine, as suggested in [Gorin et al., 2011a] for RVC.

3.5.2 Future Challenges

It is not fair to show this chapter's conclusions only as a comparison between two approaches to the same challenge that have so many in common as RVC and FCVC. In fact, the proposals of the FCVC, included as a part of this PhD Thesis, were aimed at feeding and collaborating in a debate for improving the potential of the RVC initiative and, as mentioned in previous sub-section, the main improvements suggested by the FCVC, namely the dynamic configuration of the video codec and the definition and transmission of totally new functional units, could be reflected in future amendments of the RVC standards.

Moreover, there are two main bottlenecks for the development of the FCVC ideas: the huge amount of resources necessary to deal with a potentially unrestricted codec configuration, and the security issues related to the transmission of uncontrolled programs described in DDS. We understand that as the processing capabilities increases, the first mentioned issue lose some relevance and, for now, some simpler implementations of reconfigurable encoders are feasible, as demonstrated in our experiments.

Concerning the second issue, as long as the UVD is instantiated on a virtual machine paradigm, the potential risk of an undesirable execution is reduced by putting some limits to the access to memory, registers, and other machine resources in the runtime of reconfigurable codecs. The risk should not be more serious than that of decoding a corrupted video file in a fixed decoder implementation.

With this in mind, we think that major efforts should be dedicated to the convergence of FCVC ideas (if not the particular way of implementing them) to the RVC initiative, by following the route map described in [Richardson et al., 2009c]. The DDS could be employed as a transport layer for disseminating new functional units to update the VLT in the core of each RCV decoder, providing a more powerful and fast deployment of new ideas.

Some advances in platform-independent implementations of a reconfigurable engine, such as the UVD or the JIT Adaptive Decoder Engine (Jade) described in [Gorin et al., 2011b], could lead to the standardization of the reconfigurable engine. This step could also be crucial for the future of reconfigurable coding.

Part II

Residue Filtering for video simplification

Chapter 4

Background

4.1 Coding Artifacts

The objective of a video transmission or storage system is to represent video content so that the final user perceives the best possible quality given the available resources. In this sense, almost every multimedia application dealing with video information needs of a proper algorithm for lossy video compression due to the lack (or high cost) of transmission bandwidth or storage capacity. An in-depth study of the distortions resulting from this process is mandatory for the evaluation of the overall quality of the system.

Due to the use of compression algorithms, the degradation of the visual information increases as the available bandwidth decreases, not only reducing the resemblance to the original video data, but also causing the apparition of certain encoding artifacts that severely drop the overall perceived quality of the encoded video. Leaving aside those systems in which the final user is not human (for example an automatic system for pattern recognition), some notions of the characteristics of the human visual system and the process of visual data acquisition are mandatory for the design of a competitive video system that takes into account the relationship between the distortion introduced by the encoding process and the subjective perceived quality.

In this Section, an introduction to the psychophysical analysis of the HVS is presented in Subsection 4.1.1 in order to ground an in-depth survey to the artifacts related to image and video coding and processing, discussed in Subsection 4.1.2.

4.1.1 A psychophysical introduction to the HVS

Although the study of the human visual system is out the scope of this PhD Thesis (the interested reader is referred to [Wu and Rao, 2005]), some considerations about the perception of distortion (and distortion artifacts) and the use of perceptual considerations for encoding visual information, play a key role in the design of image and video coding systems and, more specifically, in the design of the residue filtering algorithm proposed in this Part of the Thesis for simplifying the encoding operation.

During the last decades, several attempts have been made from many different disciplines to understand the processes involved in human vision. Physics and Physiology help to understand, respectively, the nature of and image and the structure of the human eye and the human visual system, and how their interaction happens. Neurology explains the transmission and processing of the nerve impulses representing the visual data along the different paths within the nervous system. Psychology also plays a key role in order to determine how the motive of a certain task influences the visual perception.

As described in [van den Branden Lambrecht and Verscheure, 1996], the psychophysical approach to the problem of human vision models the HVS by means of a transfer function. In order to obtain this model, the human perception of a scene is mainly defined by the concurrence of *sensitivity* and *masking* aspects. The sensitivity is the ability of the HVS to detect a certain isolated stimulus, while masking processes take place when a number of stimuli occur concurrently in neighboring spatial locations. Further discussion of these concepts is worth it.

Sensitivity

It has been determined with the aid of a experimental framework that the visual system is more sensitive to low spatial frequencies, with a certain cut-off frequency. In this context, the term sensitivity is defined as the inverse of the detection threshold of a certain stimulus. A Contrast Sensitivity Function (CSF) of the spatial frequency is then defined for modeling the HVS. Moreover, this sensitivity also varies with the orientation of the stimulus, showing higher values for horizontally and vertically-oriented patterns.

The part of the brain in charge of processing the visual information, the primary visual cortex, consists of different specialized areas, which can be related to different spatial frequencies and orientations and, therefore, the CSF can be understood as the result of the addition of several sensitivity functions from almost independent *channels*. This multiple-channel approach permits to analyze and process visual information in a way similar to that of the HVS, by means of a filter bank.

If the temporal dimension is introduced in this analysis, it is also true that the CSF depends on the object movement. In general, the HVS tends to exhibit a higher contrast sensitivity for moving objects than for static regions as long as the motion is not too fast to be tracked.

An additional limitation of the HVS to perceive a stimulus comes from the foveation phenomenon. The fovea corresponds to the region in the retina with the highest density of cones and ganglion cells, and it is the region in which the center of the scene (the fixation point) is projected. For this reason, the fovea exhibits the highest contrast sensitivity while, as the distance with respect to the fovea is increased, the contrast sensitivity drops. This property of the HVS is used by the so-called *foveated image processing*, with many contributions to different application fields. As a way of an example, the ROI-based perceptual encoders aim at determining the fixation point by analyzing the visual content in the scene in order to allocate more resources to those areas susceptible of attracting the attention of the observer and less resources to those regions in which, due to the foveation phenomenon, the distortion will be less noticeable.

Masking

Nevertheless, the stimuli appear in the real world in a certain context, and never alone. The context of a certain visual stimulus influences in its detection threshold.

This interaction affects the way in which the stimulus is perceived by *facilitating* or *masking* its detection. This is particularly interesting when dealing with distortions introduced when processing visual signals. As a way of an example, determining the masking potential of the different regions in a frame enables to wisely distribute the distortion (or the coding resources) in a way that the overall subjective quality is optimized.

Some of the most referred works in perceptual coding are related to masking properties of the HVS and how to take advantage of them for allocating resources in a way that the overall subjective quality is enhanced (for some examples see [Minoo and Nguyen, 2005], [Yu et al., 2005] or [Tan et al., 1996]).

The intensity contrast masking or brightness masking appears when a certain stimulus is placed in a very dark or very bright region. A *shield effect* takes place and the stimulus is harder to distinguish. This kind of masking is typical in photography due to underexposing or overexposing.

Additionally, texture masking appears in the presence of complex textures. Complex or even hardly-recognizable patterns mask other stimuli placed in the same location. A good example could be a landscape photograph in which a certain region of the picture is covered by grass or a complex structure of interlaced branches and leaves. In those areas, the HVS would be less sensitive to noise or other distortions given that it hardly perceives the real texture. It is important to differentiate this phenomenon from the lack of sensitivity at high frequencies. The presence of a masker (a complex-textured region) is the key factor here making difficult to perceive any other stimulus.

Although it is not clearly defined in the literature, there may be also a motion masking, related to the inability of the HVS to properly track very fast moving objects. In this situation, the human brain is more concerned to track the trajectory of the object than to distinguish its texture or even its color. If any noise or distorting signal accompanies the moving object, its presence will be masked and it will remain unnoticed.

4.1.2 Summary of Encoding artifacts

Some encoding artifacts have been identified as responsible for reducing the output quality of a sequence processed by a hybrid DPCM/DCT video encoder. A complete list can be found in [Wu and Rao, 2005] together with a detailed explanation of their causes and effects, but those more important for low bit-rate video coding are summarized in this section.

Block artifacts

This category includes all the artifacts derived from the block-wise processing of video sequences in blocks and are obviously related to the visibility of subtle (or coarse) differences between neighboring blocks, highlighted by their regular spacing. Although there are different kinds of artifacts related to blocks, almost all of them take place when the bit rate is highly constrained and the quantization scale is coarse.

The *blocking effect* is detected because of the differences between pixel values at the block borders with respect to those of its neighbors. It is caused by the different encoding options of the blocks involved, in terms of quantization parameter or block type. It is more visible with higher quantizer values and in homogeneous regions. For Inter frames, the blocking effect is limited to motion regions, i.e., those producing some prediction residue, and its effects are more visible in blocks covering the borders of moving objects. On the contrary, the static areas are well predicted and there is almost no risk of blocking artifacts. A secondary blocking effect can be also detected in Inter frames in the boundaries between small partitions of the macroblock.

Another form of artifacts related to the use of macroblocks for video coding is known as the *mosaic effect*, a large scale artifact that makes the image appear as composed of a set of different small square tiles (the macroblocks). Although it usually come hand to hand to the blocking artifact, they are not necessarily concurrent: there is not a unique cause of the mosaic effect and, moreover, the presence of blocking in certain areas (for example in homogeneous regions) does not necessarily produces a mosaic effect. Moreover, its possible that a deblocking post-filter process removing the differences in the block boundaries did not succeed in removing the mosaic effect.

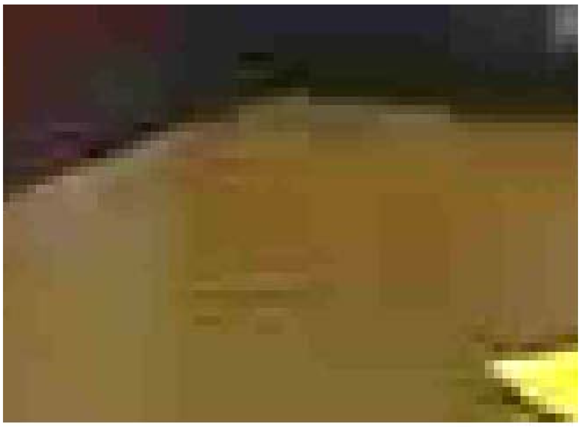


Figure 4.1: Detail of basis function effect in the sequence *paris*.

Finally, the *basis image effect* appears when certain blocks are reconstructed in a way that resembles the pattern of one of the basis functions of the DCT (or a combination of some of them), as can be seen in 4.1. It is caused by a coarse quantization that removes all but a few transform coefficients in a high-detailed residue. This effect also tends to increase the visibility of mosaic patterns and blocking effect.

Blurring

The *blurring effect* is normally caused by a reduction of high frequency components, which results from either a coarse quantization process or other reasons. This loss of high-frequency components fades boundaries and details of objects, making them appear as out of focus and plain, as in Figure . Furthermore, encoding is not the only source of blurring effect, but also the video capture, editing or interlace handling can introduce visible blurring artifacts.

The corresponding color version of this artifact is the *color bleeding*, for which object borders in crominance components are blurred as a consequence of a loss in high-frequency components. This effect does not necessarily appears together with the luminance blurring and it is sometimes separately detected.

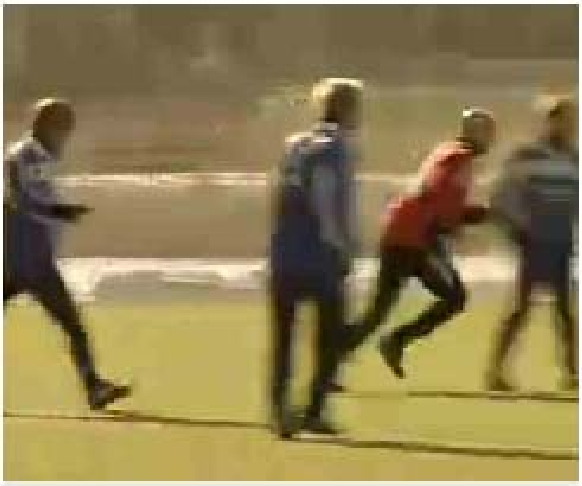


Figure 4.2: Blurring effect in the sequence *soccer*.

Edge artifacts

Numerous artifacts are related to a poor representation of edges, both in intra- and inter-predicted frames. For example, the edges of a moving object can be distorted or misplaced due to a poor representation of the encoding residue after motion compensating a block containing objects with different motion characteristics. Although these errors could increase the blocking and mosaic artifacts, they also appear themselves as high-frequency fluctuations, the so-called *mosquito effect*.

But probably the most recognizable edge artifact is the *ringing effect*. It is a particular consequence of the Gibb's Phenomenon, in which a signal showing an abrupt leap or discontinuity is represented in the frequency domain with only a limited number of coefficients, resulting in a rippling effect near the discontinuity. In a 2D space, this effect appears as a succession of displaced copies of the edge like in a wave pattern. Figure 4.3 shows an example of ringing effect around the silhouette of the skaters. This ringing effect can be more easily perceived when working with a large quantization step, which tends to reduce high-frequency components, limiting the frequency resolution of the reconstructed block, both in inter- and intra-predicted frames.

Coarse quantization of transform coefficients of the residue of intra-predicted

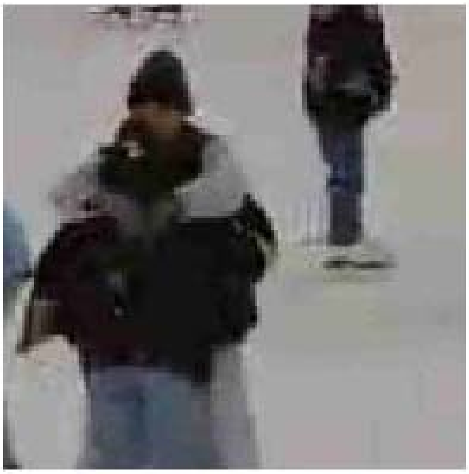


Figure 4.3: Detail of ringing effect in the sequence *ice*.

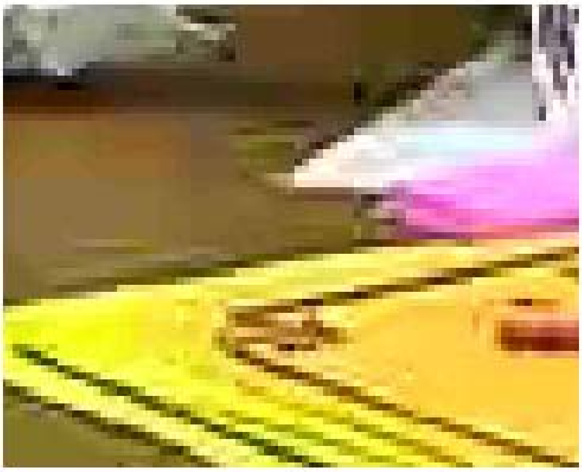


Figure 4.4: Detail of staircase effect in the sequence *paris*.

blocks also leads to a staircase effect, in which high-frequency diagonal frequencies are removed from the reconstructed pixels and diagonal object borders are represented by successive horizontal and/or vertical segments, as can be seen in Figure 4.4.

Temporal artifacts

When enough bits are available for encoding the residue, the use of inter-prediction tends to mitigate encoding artifacts in successive frames. On the other hand, the motion compensation can also be a source of new artifacts if not enough resources are allocated for the residue representation. For example, motion estimating and compensating a frame over a reference containing blocking effects may replicate them in different positions of current reconstructed frame creating *false edges*, which are more noticeable in homogeneous regions and can not be mitigated by a traditional deblocking filter because they are not to be located in block border positions. This effect can be observed in Figure 4.3, around the leg of the skater at the right side of the frame.

Another example of the motion compensation highlighting an artifact is the mosquito effect. Although it has been previously introduced, it should be noticed that it may be also related to temporal fluctuations near strong edges or moving objects, sometimes as a consequence of replicating ringing effects from one frame to the next due to motion compensation.

Additionally, there are two different kinds of flicker effects to be considered in hybrid video coding. In the first place, in homogeneous regions, a *block fluctuation* artifact may appear. It is caused by the encoding of co-located blocks in successive images with different encoding parameters, such as quantizer, prediction mode or partition. In these situations, the flickering effect will be more visible if there is not enough bit rate for encoding the prediction residue. Second, the most common kind of flicker artifacts is caused by the periodic introduction of Intra frames due to the GOP structure. These intra-predicted frame tend to refresh certain details lost during the continuous inter-prediction process carried out between successive frames in the GOP. The refreshing effect sometimes is perceived as a flicker with a period that equals that of the GOP structure. Again, this artifact is more visible when working at low bit rates due to the lost of details in the inter-prediction residue.

Other artifacts

There are plenty of artifacts other than those stated previously in this Section caused by different processes involved in video processing and coding. For example, there is a *color shifting* artifact related to the fact that motion estimation is performed only in the luminance component and the resulting motion vector is scaled and employed in the crominance motion compensation. This approximation is not always good and there could be a drift between luminance and crominance in the reconstructed frame.

Another examples could be the aliasing artifacts related to a temporal down-sampling of the frames in the sequence, the distortions produced by the de-interlacing of fields, or the scaling between different video formats (e.g.: from 4:3 to 16:9).

4.2 Image and video filtering

Filtering has been a common task in signal processing due to the ubiquitous presence of noise in every electronic device. Although digital signal processing methods have significantly improved and more powerful techniques for error resilient transmission and storage are available, some noise sources simply cannot be avoided and the denoising of image and video is still a matter of interest. The interested reader is pointed to [Buades et al., 2005] for a more complete survey in modern denoising techniques. In this section, given that we need some notions about simple mechanisms for image and video filtering (those capable of being integrated into an encoder engine as will be discussed later on), noise modeling is introduced together with some of the most broadly-employed denoising mechanisms summarized from [Bovik, 2009].

The basic model for noise in image or video frames is the additive noise model and obeys the following equation:

$$S\left(i,j\right) = I\left(i,j\right) + w\left(i,j\right), \tag{4.1}$$

where $I(i,j)$ are the original pixel values with coordinates i and j, w represents the noise signal, and S are the observed pixels. The main concern in the design of filters for denoising is to discriminate between the actual noise and the particular texture in the image, or equivalently, to estimate I from the noisy S values.

Noise is usually modeled as a random variable whose distribution depends highly on the nature of the involved noise sources. For example, thermal noise can be modeled by a Gaussian distribution in virtue of the Central Limit Theorem (CLT). According to this theorem, the addition of a high number of independent noise sources without any dominant component can be modeled as a unique Gaussian noise with a Probability Density Function (PDF), $p_w(x)$, described for the univariate case as:

$$p_w(x) = \frac{\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}{\sqrt{2\pi\sigma^2}}, \tag{4.2}$$

with mean μ and variance σ^2.

The vibration of electrons in electronic devices due to the temperature is an excellent example of a signal obeying the CLT. Therefore, the thermal noise exhibits a Gaussian distribution. Nevertheless, there is also room for other noise distributions of certain importance in the field of signal processing. Specifically, some kinds of noise found in visual registers are briefly summarized here:

- The *photon counting noise* is observed in image and video capture, for example with Charge-Coupled Devices (CCD), and follows a Poisson PDF with higher noise values in bright regions.

- The *salt and pepper* noise is result of the random appearance of a small number of black or white pixels in an image. This noise follows a distribution $p_w(x)$ with zero mean but no negligible values for large x. This statistical model is sometimes referred as a *heavy-tailed* PDF.

- The quantization carried out in the digitalization process of image and video introduces a quantization noise which is usually modeled as a uniform distribution in the range $[-\Delta/2, \Delta/2]$ with zero probability of larger values and, therefore, with no tails at all. Other processes including a quantization stage, such as JPEG-like encoding, suffer from a similar noise.

- Other PDFs like Laplacian, exponential, Cauchy, etc., can be used for modeling certain noise sources, but under different conditions all of them may be considered as Gaussian without an important loss of generality.

Unless noted otherwise, from now on we will assume a zero-mean Additive White Gaussian Noise (AWGN) model.

Given that image and video sequences exhibit mainly low and medium frequencies, the basic approach for noise removal consists in a smoothing filter that averages each sample from those in its spatial or temporal neighborhood, removing high frequencies in which the noise prevails over the signal components. Of course, this approach may fail when pixel values change abruptly. In this cases, high frequency information may be distorted as a result of the smoothing filter. For example, as will be discussed in next section, linear filters are sub-optimal for noise removal in the sense that certain image details are blurred.

Additionally, non-linear approaches based on Order Statistics (OS) are very useful for smoothing a noise-corrupted frame or sequence, reducing the influence of noise while preserving high-frequency image details, even for impulsive noise, like the *salt and pepper* noise. The most widely-used non-linear OS filter is the median operator. Morphological filters are also important for applications such as pattern recognition and classification but they are not widely used for denoising. These non-linear alternatives will be discussed in Section 4.2.2.

4.2.1 Linear Filters

A linear filter applied to a noisy signal is in essence the linear prediction of the original signal from the noisy samples, following this formula:

$$\hat{I}\left(i,j\right) = \sum_{x,y\in\Omega} h_{xy} \cdot S\left(i-x,j-y\right), \tag{4.3}$$

where Ω is the filter support, defined as an offset with respect to the original coordinates (i, j), and h_{xy} are the weighting factors for the samples within the filter support. If the filter is adapted to the statistics of the particular source, the weights minimizing the Mean Square Error (MSE) between the estimate $\hat{I}$ and the original image values I can be found towards the well-known analysis of Wiener-Hopf.

The weights in Equation 4.3 can be expressed as a weighting matrix H which characterizes the filter, whose operation can also be defined as a linear convolution $\hat{I} = S * H$, with H referred as the filter *kernel*. The particular structure and values of H determine the function of the linear filter.

A basic smoothing filter calculates the average of the pixels around the current position by multiplying by a square-mask with constant value in the understanding that image pixel values are locally stationary. Given that this assumption tend to fail for real images, basic averaging filters tend to excessively blur important image details. Instead of using these constant masks, other designs giving more importance to the central pixel achieve a better performance. Particularly, a Gaussian filter is very effective for avoiding drops on the sides of the support and preventing the appearance of some artifacts. Its weights correspond to a two-dimensional Gaussian function described as follows:

$$H_G(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \tag{4.4}$$

where σ^2 is the variance of the Gaussian function and enables to control the filter *strength*, an important issue in adaptive filtering. In order to properly work with this kind of low-pass filters, the filter support should cover at least two-times the value of σ in each direction with respect to the center in order to avoid abrupt drops of the filter response in the support boundaries. Moreover, Gaussian filters are also separable: a 2D Gaussian filter can be applied as a pair of 1D filters leading to an important advantage for implementation.

Blurring in linear filters

In general, linear filters do not depend on signal values, and this constitutes both an advantage and its main drawback when employed as smoothing filters for denoising. On the one hand, they are simple and effective in many situations but, on the other hand, its low-pass response removes noise as well as high-frequency signal details introducing blurring artifacts. Some techniques may help to deal with blurring artifacts.

In [Qi et al., 2006], a Gaussian pre-filter is applied prior to video coding. In order to prevent the appearance of blurring in important edges of the video frames, edge blocks are detected and each one is filtered or not according to a rate-distortion-based decision.

The blurring effect is alleviated in [Florencio, 2001] by applying the spatial filter only in static areas, in the understanding that moving objects are more probable candidates for capturing visual attention. In the same way, [Karlsson and Sjostrom, 2005] classifies the ROI and background of video frames, filtering more strongly those regions outside the ROI, considering that blurring effects can be ignored as soon as they appear in background areas.

Other works like [Lin and Ortega, 1997] suggest to control the amount of filtering according to the quantization scale to be employed by the subsequent encoding process. A higher q means that potential blurring artifacts introduced by the filter may be masked and, the encoding process could benefit from the corresponding simplification of the video sequence. In the same way, [Karunaratne et al., 2001] or [Segall et al., 2001] try to estimate the amount of quantization noise that will be introduced by the encoding process and then adapt the linear filter strength for keeping the filter distortion less noticeable than the effects of quantization.

In any case, blurring is unavoidable when real-world images or video sequences are linearly filtered and, therefore, the use of this kind of smoothing is only recommended for low-complexity constraints or when the filtered regions could benefit from masking effects.

4.2.2 Non-Linear Filters

As an alternative to linear filters, non-linear approaches are capable of reducing noise while avoiding blurring artifacts. For this reason, some of these techniques have become very popular for different denoising applications, from the simplest implementations in consumer cameras to the professional video editing or artistic applications.

The main characteristic of non-linear filters is that filtering weights depend on the values of the image to be filtered. Although there are many techniques, the most popular ones are based in order statistics (OS) like maximum and minimum filters or the different versions of median filters. Furthermore, bilateral filters claim to be the best option for image denoising. Finally, morphologic operators are more suitable for pattern recognition.

Order Statistics

Given a data sample $X : \{x(i)\}_{i=1..N}$, the OS are obtained by first ordering the values of the original sample in a new set $\left\{x_{(j)}\right\}_{j=1..N} : x_{(j_1)} > x_{(j_0)}, \forall j_1 > j_0$. The OS are then addressed by its index j in the ordered set so that certain interesting operations can be defined over them:

$$min(X) = x_{(1)} \tag{4.5}$$

$$max(X) = x_{(N)} \tag{4.6}$$

$$median(X) = \begin{cases} x_{\left(\frac{N+1}{2}\right)} & \text{, if N odd} \\ mean\left(x_{\left(\frac{N}{2}\right)} + x_{\left(\frac{N}{2}+1\right)}\right) & \text{, if N even} \end{cases} \tag{4.7}$$

An interesting property of these operators is that their result is always one of the values in the original set (except when calculating the median on an even number of elements). If we apply the operations of Equations 4.5, 4.6 and 4.7 on pixel values taken from a neighborhood of the current pixel, we obtain respectively the minimum, maximum, and median filters. The last one, described in Equation 4.7 is very popular for image smoothing because it is less sensitive to outliers.

$$\hat{I}\left(i,j\right) = median\left\{S\left(i+x,j+y\right) : \left(x,y\right) \in \Omega\right\}. \tag{4.8}$$

Weighted Median Filter

Basic median filters applied to image denoising tend to fail when noise levels are low and the image exhibits thin lines or sharp corners. In these situations, some important details tend to be removed by a conventional median filter. There are many ways for adapting the median filter to the actual signal characteristics in order to control the amount of smoothing, but probably the most widely used are variations of the Weighted Median Filter (WMF).

As described in Section 4.2.1, linear averaging filters can be improved by means of applying different weights for each element in the filter support, making the contribution of the central pixels more important than those near the mask borders. Gaussian filters, for example, are designed to accomplish this task by the calculation of filter coefficients towards a Gaussian bell function centered in the current position of the pixel to be filtered.

In the same way, giving more importance to the values of those pixels near the center of the support in OS filters is carried out, not by multiplying by a factor, but by repeating these values a certain number of times. For instance, if a 3x3 square support is used, the mask defined in Equation 4.9 could be employed for giving more importance to center and horizontal and vertical samples. Each element covered by this mask would be introduced in the ordered list of values as many times as the corresponding coefficient.

$$H = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{4.9}$$

This basic idea is employed for defining the WMF, as described in [Brownrigg, 1984]. In general, WMF allows for controlling the behavior of the filter and adjusting the trade-off between noise suppressed and blurring of original details. For example, this procedure is followed in [Ko and Lee, 1991] for giving more importance to the center point of the mask. This Center Weighted Median Filter can be adapted depending on the noise and signal power by only changing this central value in the mask.

Bilateral Filters

A particularly interesting class of non-linear filters are the bilateral filters. Described originally in [Tomasi and Manduchi, 1998], they perform two different filtering stages. The *domain* stage is a linear Gaussian filter whose weights are calculated according to the Euclidean distance between the point to be filtered and those in the filter support, like the one described in Equation 4.4. On the other hand, the *range* filter extends the concept of distance by assigning higher weights to those pixels in the filter support more similar to that in the center position. The combined bilateral filter is defined as follows:

$$\hat{I}\left(\mathbf{p_0}\right) = \frac{\sum\limits_{\mathbf{p}\in\Omega} S\left(\mathbf{p}\right) G\left(\left\|\mathbf{p}-\mathbf{p_0}\right\|, \sigma_d\right) G\left(\left|S(\mathbf{p})-S(\mathbf{p_0})\right|, \sigma_r\right)}{\sum\limits_{\mathbf{p}\in\Omega} G\left(\left\|\mathbf{p}-\mathbf{p_0}\right\|, \sigma_d\right) G\left(\left|S(\mathbf{p})-S(\mathbf{p_0})\right|, \sigma_r\right)}, \tag{4.10}$$

where $S\left(\mathbf{p}\right)$ is the pixel value in position $\mathbf{p}$ of the noisy image, $\left\|\cdot\right\|$ denotes the Euclidean distance, and $G\left(q,\sigma\right) = e^{-\frac{q^2}{2\sigma^2}}$ is a Gaussian function with variance σ. As can be seen, higher differences with respect to the center pixel value $\mathbf{p_0}$ produce a lower value of the range filter, that at the end makes possible to prevent the blurring of certain details in the image like lines or sharp corners.

The filter strength can be controlled by means of the domain and range filter variances, σ_d and σ_r respectively. For example, an adaptive bilateral filter is proposed in [Li and Xu, 2006] for simplifying video for encoding at low bit rate, whose values of σ_d and σ_r are controlled depending on the quantizer to be used in the encoding process. In a similar way, filter strength is controlled in [Xu et al., 2007] based on a ROI-oriented importance map extracted from detecting gaze in a robotic tele-surgery application. Finally, a simpler approach is employed in [Kim et al., 2010] relying on a motion-activity measure.

Although the results of bilateral filters for edge preserving are outstanding, it is to be noticed that range filter stage is nonlinear. The reason is that its weights depend on the image pixel values, which makes the bilateral filter non-separable, an important drawback for practical implementations. In this sense, in [Pham and van Vliet, 2005] a separable approximation of bilateral filters is proposed that is notably faster.

Morphologic Filters

Another important category of non-linear filters are the morphological operators. Nevertheless, they are more known for processing binary images (with only two values of luminance, black and white) and they are more popular in pattern recognition applications.

The most popular morphological filters are compositions of two basic operations, dilation and erosion. Their grayscale versions are based in the maximum and minimum operations described in Equations 4.5 and 4.6, defined over a certain Structuring Element (SE) similar to the previously defined concept of filter support. The only difference is that the definition of the SE may include offset parameters. These offsets are added to the corresponding pixels in the image before applying the corresponding operator, influencing the result. Grayscale dilation and erosion over a structuring element H are respectively defined as follows:

$$(I \oplus H)(i,j) = \underset{(x,y)\in H}{max} \{I(i+x, j+y) + H(x,y)\} \tag{4.11}$$

$$(I \ominus H)(i,j) = \underset{(x,y)\in H}{min} \{I(i+x, j+y) - H(x,y)\} \tag{4.12}$$

The basic morphological operators described in Equations 4.11 and 4.12 can be controlled by changing the shape and values of the SE. Nevertheless, the main use of these filters consists in their combination into more complex operators like *Opening* and *Closing*, of paramount importance for binary image processing and pattern recognition. Basically, an opening denotes the use of the same structuring element for performing an erosion followed by a dilation, and in binary imaging it is used to eliminate isolated '1' values. A closing operation is a dilation followed by an erosion with the same SE, and it is employed for filling the gaps in regions with predominant

’1’ values. However, the use of these operators for denoising is not very recommended given that the minimum and maximum filters perform even worse than median or mean filters for smoothing gray scale images.

4.2.3 Beyond the spatial support

For the sake of simplicity, the algorithms described along this Section have been formulated as 1D or 2D spatial image filters. But image denoising can also be performed in domains different than the spatial domain. For example, certain filters operate on transformed domains such as the Fourier space or on a multi-scale decomposition of the noisy image instead of filtering in the spatial domain. Others, when working with video sequences, incorporate the temporal dimension performing a 3D filtering.

Temporal Filters

When denoising video sequences, a simple approach could be independently filtering each frame in the sequence with one of the filters previously described. However, this approach would not take into account the temporal redundancy of video information, which is even higher than the spatial redundancy. Therefore, extending the filter support to different frame instants and considering the video sequence as a 3D signal leads to a variety of promising techniques for video denoising. In this way, it is possible to build temporal filters according to the same techniques described above. For example, a linear temporal filter could be applied to those co-located pixels of a certain number of past and/or successive frames in a temporal support as follows:

$$\hat{I}\left(i,j,n\right) = \sum_{k \in \Omega} h_k \cdot S\left(i,j,n-k\right) \tag{4.13}$$

However, the same nature of video information implies that the visual content of certain regions in a frame may change substantially with respect to co-located regions in neighboring frames because of the displacement of the objects in the scene or even due to camera motion or scene changes. In these situations, the filter described in Equation 4.13 would perform poorly, leading to blurring effects like those observed in object boundaries for linear spatial filtering or even to harmful temporal artifacts like those described in Section 4.1.2. Fortunately, there are many techniques to overcome this drawback.

First of all, in order to not rely only on the temporal processing, some filters are applied to a spatiotemporal support, constituting the 3D filters. For example, a 3D linear filter generalized from Equations 4.3 and 4.13 is defined as follows:

$$\hat{I}\left(i,j,n\right) = \sum_{x,y,k \in \Omega} h_{xyk} \cdot S\left(i-x, j-y, n-k\right), \tag{4.14}$$

where n is the current frame index and the different k indexes correspond to previous or subsequent frames in the video sequence.

This basic scheme can be further improved by a proper selection of the filter coefficients h_{xyk} depending on the movement expected or detected for the current position (i, j) in consecutive frames. For example, in [Ozkan et al., 1993] a complete motion estimation procedure is performed in order to modify the temporal extent of the 3D support using motion-compensated positions instead of taking the co-located position. Other options avoid performing motion estimation and rely on simpler motion activity measures such as frame differences to classify which regions in each frame are in motion and which ones are static in order to use only spatial filtering or a 3D filter, respectively.

Multi-Scale Filters

Multi-scale image representations, or the more general approach of multi-resolution analysis, have provided to image processing with powerful techniques. The basic idea is the use of a pyramid representation of images by iteratively smoothing and sub-sampling the original image. This idea is employed, for example, in JPEG-2000 for image coding. Moreover, in the field of image denoising, pyramid representation enables to better differentiate between signal and noise. Based in this premise, a meta-procedure for adapting any kind of denoising filter to a multi-scale image representation is described in [Burger and Harmeling, 2011].

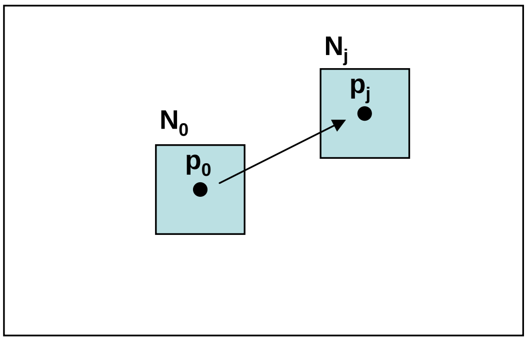


Figure 4.5: Regions compared to compute the weight assigned to pixel in position $\mathbf{p_j}$ in the calculation of the filtered pixel corresponding to position $\mathbf{p_0}$.

Basically, in high-frequency bands of the pyramid representation, noise coefficients are dominant with respect to signal coefficients (in the same sense that image pixels exhibit a higher spatial correlation than noise values in natural images). As described in [Rajashekar and Simoncelli, 2009], denoising can be performed without a noticeable distortion of the original image by abruptly removing some of the high-frequency sub-bands composed mainly by noise, or by re-synthesizing the image by the weighting of each sub-band by a factor depending on the noise and signal energy of the sub-band.

Other Filters

Following the same reasoning that adaptive temporal and bilateral filters, the Non-Local Means (NLM) Filter is proposed in [Buades et al., 2005]. NLM Filter is, in principle, a 2D linear spatial filter like the one described by Equation 4.3, whose support may cover the entire image. When calculating the filtered value for the position $\mathbf{p_0}$ in the frame, the filter weight corresponding to position $\mathbf{p_j}$ is calculated according to the similarity between a region N_0 around $\mathbf{p_0}$ and a region N_j around $\mathbf{p_j}$ for a certain window size, as shown in Figure 4.5. This similarity is measured by means of a Gaussian weighted Euclidean distance similar to that employed in the Range Filter description in Section 4.2.2. A more complex approach is adopted in [Balle, 2010] for introducing frequency-selection in the NLM or bilateral filter approaches. The patch-based criterion for weight calculation is substituted by a formulation in the frequency domain. The proposed Gabor space coefficients represent the energy located in a certain position within the frame with a certain frequency and orientation. Filter parameters allow for selecting which textures are to be removed and which should be preserved.

Chapter 5

Pre-filtering for low bit rate video coding

Science never solves a problem without creating ten more. (G. Bernard Shaw)

5.1 Introduction

As seen in Chapter 2, the output bit rate in a hybrid video encoder is controlled by means of the proper variation of the encoding parameters, above all the quantizer scale. The general approach to obtain the proper QP for a certain rate constraint goes through two stages: 1) a bit allocation process, in which the available bits are distributed among the different frames or smaller basic units and a particular bit budget is determined for each of them; and 2) a QP selection stage, in which the proper quantizer scale of each basic unit is selected from the previously allocated target bits. Although the encoder is designed to reduce non-relevant information when operating at medium to high bit rates, a low bit rate constraint requires to severely drop the transmitted data and, therefore, employ higher quantization parameters, also removing perceptually relevant details. Moreover, encoding with a large quantization step, as described in Section 4.1, increases the risk of appearance of harmful artifacts such as ringing, blocking, mosaic or mosquito effects.

A number of noise sources affect the visual data records prior to the encoding process itself, coming for example from the film grain of analog sources or from the noise present in the electronic devices employed for the capture of visual information. The presence of this noise not only reduces the perceived quality but also implies a more complex representation of the visual data that makes more difficult the task of image or video encoding. Different filtering techniques traditionally applied to image and video denoising have shown themselves to be an alternative to the increment of QP for low bit rate encoding or, at least, a useful tool for alleviating the encoder from high-frequency details that would otherwise increase the output bit rate.

In this PhD Thesis, these techniques for video simplification are studied in-depth and a video simplification mechanism is proposed for low bit rate video coding aimed at satisfying the following design conditions:

- The algorithm should remove the less relevant high-frequency information to make possible the encoding of the image sequence with lower QP values for the particular rate constraints, in the hope that this QP reduction will also avoid artifacts as ringing, blocking or mosquito in the reconstructed images.

- This process must preserve relevant details of the original sequence and, as far as possible, it should not introduce additional artifacts or harmful effects in the reconstructed frames.

- It should be simple enough to be run together with the encoder itself in real-time applications.

- Given the time-varying nature of video information, the algorithm should adapt to the content by means of configurable parameters, becoming robust against changes in the video content such as scene changes or camera motion.

- Additionally, it is desirable to employ an algorithm adaptable to different video codecs.

Section 5.2 will be devoted to the study of the state-of-the-art techniques in video filtering for simplification, some of them already introduced in Section 4.2, in order to ground later assumptions and design decisions for the proposed filter described in Section 5.3. As a crucial part of the proposed design, Section 5.4 will deal separately with the problem of coupling a filter control to the rate control. Finally, some experiments described in Section 5.5 will lead us to a discussion about the potential use of these kind of tools in practical video codec implementations in Section 5.6.

5.2 Related Work

5.2.1 Pre-filtering for simplification

As mentioned before, filtering schemes can be used as tools for simplifying images or video sequences in a way that the subsequent encoding of these data could achieve a better RD performance. The basic idea behind this assertion is the reduction of noise and, in general, of a great part of the high-frequency components of the image or video sequence, which are typically a burden for the encoding performance. The joint design of simplification filters and encoders was proposed by Lin and Ortega in their works on the late nineties (see [Lin and Ortega, 1997]).

Nevertheless, there are some differences between using a filter scheme for denois-

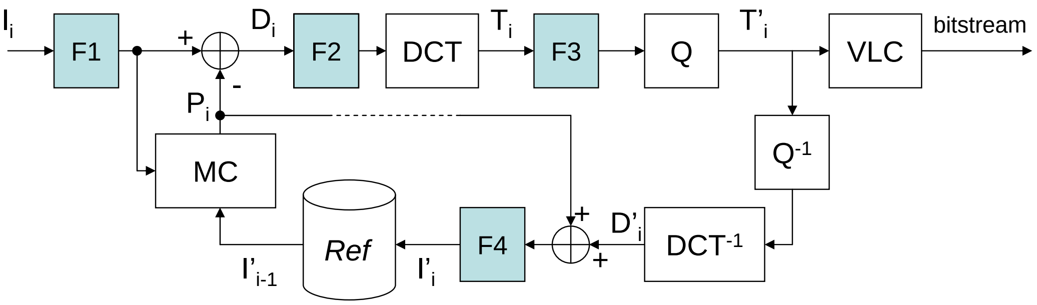


Figure 5.1: Different locations for filtering in a hybrid video encoder.

ing and using it for simplifying video content. Although a blind denoising of the original sequence would surely benefit the encoding process, the joint design of both processes and the adaptivity of the filters to the encoding conditions could achieve a much better performance in terms of reducing the encoding artifacts at a given bit rate. For example, the introduction of filtering artifacts such as blurring in certain situations could be tolerable as long as the characteristic artifacts of low bit rate video coding were reduced. Although, as pointed out by [Lin and Ortega, 1997], there is no evidence that the human vision was less sensitive to one artifact or the others, the fact is that the quality of smoothed sequences is more consistent than that of sequences corrupted by low bit rate encoding artifacts such as blocking, ringing or mosquito effect.

Video simplification can be equally performed on the whole frame or can be applied unequally based on a perceptual criterion. The so-called perceptual video coding constitutes an important field in which filters similar to those introduced in previous sections have an important role. The basic idea of perceptual coding is to distribute the encoding resources (namely bits) in such a way that the distortion perceived by a human observer is reduced with respect to an even distribution of the same resources. To this end, different criteria are employed that were already mentioned in Section 4.1.1. Some of them are the luminance, frequency and texture masking, and above all, the foveation-based techniques that are the basis of saliency coding.

Filtering has in these applications an important role to play. It allows for allocating bits unequally within the frames: employing less bits in non-relevant, non-salient or masked regions. The alternative of variating the QP on a macroblock basis implies a risk of blocking or mosaic artifacts due to the quantization of adjacent blocks with different steps. For some examples of foveation-based techniques, see [Karlsson and Sjostrom, 2005], [Gopalan, 2009] or [Lu and Zhang, 2011]. In these proposals, a saliency map is constructed (representing an estimate of the ROI), which determines the strength of a Gaussian filter simulating the foveation effect: the loss of sensitivity of the human eye for those objects projected away from the fovea. However, many other schemes include some kind of perceptual considerations to discriminate those regions in which filtering can save bits without noticeable distortions.

Moreover, the pre-filtering approach for denoising described in Section 4.2 is only one of the possible schemes that may be employed to combine the filtering and encoding processes, as described in [Segall and Katsaggelos, 2000]. As can be seen in Figure 5.1 for a typical hybrid video encoder scheme, the alternatives to classic pre-filtering (F1) could be the residue filtering (F2), transform-domain filtering (F3) and in-loop post-filtering (F4), which will be discussed later. On the other hand, out-of-loop post-filtering is not included in this relation as it is not related to the encoding process itself, but employed at the output of the decoding process for enhancing the reconstructed sequence prior to display. Moreover, an additional location could be defined at the output of the quantization process for simplifying the quantized transformed residue but this possible simplification could also be considered as part of the quantization process design itself and it is out of the scope of this document.

Pre-filtering for video compression

The most broadly-used configuration for simplifying the encoding process by means of filtering is carried out prior to the encoding process itself. As previously mentioned, the use of this scheme comes from the denoising applications and the mechanism is in essence the same as the one employed for that task: to consider some non-relevant details in the image sequence as noise and to filter this noise keeping the relevant visual details unaltered. Pre-filtering the high-frequency bands introduces an almost non-noticeable distortion and, at the same time, reduces the encoder output bit rate. This is the case of the traditional spatial smoothing filters included in many proposals related to the topic. The differences between them usually lie in the particular filter architecture employed and the mechanism to control the amount of filtering.

The first example is the 2D Gaussian filter employed in [Lin and Ortega, 1997]. This proposal suggests to control the filter strength s according to the quantizer q employed by the H.263 encoder (in which the filter is implemented). The filter control is then integrated into the rate-distortion framework and the RD cost function for the decision of (q,s) is modified in order to penalize the blocking effect. It is to be noticed that the amount of filtering is modified only in the Intra frames in order to not be detrimental to the motion compensation performance. Finally, some perceptual considerations are taken into account on a MB basis that prevent to employ large q values in flat regions in order to reduce blocking effects. Regardless of the PSNR loss, the algorithm achieves a significant blocking artifact reduction, producing a smoother output.

A slightly different approach can be found in [Kawada et al., 2006]. This proposal employs a pre-analysis stage in order to adaptively decide the proper amount of filtering for the original frames in the sequence. An estimate of the expected encoding distortion of the non-filtered sequence is compared to another estimate of the distortion in case the pre-filtering scheme is employed. A decision is then made on a frame basis in order to determine whether the filter is employed or not. The complexity is the main drawback of this solution, which needs to perform an additional motion compensation as a part of the pre-analysis stage. It is also to be noticed that the distortion estimations rely on pixel-wise measures and, therefore, the potential benefit of this proposal is PSNR increment, which is not necessarily related to subjective quality.

In other examples, the proper amount of filtering is selected according to certain spatiotemporal activity measures, on the understanding that it is important to preserve object edges and moving objects. This last concern is related to the concept of ROI in the sense that moving objects are strong candidates for attracting visual attention. In [Song et al., 2004], for example, a modified 1D Gaussian filter is applied in order to simplify sequences prior to an H.264/AVC encoding process, and an activity measure guarantees that no object boundaries or salient objects are blurred. In the same sense, the proposal in [Florencio, 2001] employs a motion-aware filter for pre-processing the video sequence.

As described in Section 4.2, a drawback of many pre-filtering schemes is the blurring of relevant details of the sequence, above all in the linear filters. For this reason, some non-linear approaches have been employed for video simplification. For example, the proposal in [Kim et al., 2010] uses a motion-aware bilateral filtering (similar to those described in Section 4.2) for edge preservation with two different intensities: a deep filtering for stationary regions and a light filtering for moving objects.

As described in previous paragraphs, the typical pre-filtering schemes for video simplification are mostly simple enough to be incorporated to video encoders and, although there are other more complex proposals for video and image filtering that achieve better results in terms of noise suppression and absence of blurring artifacts (see [Balle, 2010] for a good example), they fail to comply with the constraints of a video coding scenario in terms of complexity or delay.

Motion Adaptive Bilateral Filter

A motion-compensated bilateral filter for video simplification was proposed in [de Frutos-Lopez et al., 2012] by our research group. The bilateral filter configuration was chosen in order to preserve edges and important details from a potential blurring. In essence, it simplifies more strongly those static areas of the original frames in order to save bits to moving objects, which are identified as the potential ROI. To this end, the filter is applied to a spatial support but its parameters are selected for each pixel relying on the comparison of a small region around the pixel with respect to a camera-motion-compensated position in the previous frame. The motion estimation carried out in this proposal is robust to camera motion, i.e., the camera motion is compensated from the motion vectors field. When camera motion is present, the displacement of an object between consecutive frames is not necessarily related to its saliency, as it occurs in absence of camera motion. Moreover, those objects static with respect to camera (describing the same motion trajectory than that of the camera) seem to attract the attention of both the camera and the observer. After these considerations, the final results of the MABF when applied to a low bit rate encoding scenario show an improvement in visual quality (reduction of coding artifacts). However, it was very difficult to obtain a proper set of filter parameters for all the sequences, given that different aspects such as the presence of camera motion or the size of the ROI can notably alter the potential bit savings from filtering the background. Apparently, the particular structure of motion compensation in hybrid video coding may explain these mismatches. A compromise solution was adopted in this work, but we think it could be outperformed by a residue filtering scheme.

Residue filtering

The same filtering schemes proposed as pre-filters in previous paragraphs could be applied to filter the motion-compensation residue or the intra-prediction residue. For example, a Gaussian smoothing filter is applied in [Segall et al., 2001] over the motion-compensation residue. Again, as in [Lin and Ortega, 1997], filtering should maintain a trade-off between blurring and low bit rate coding artifacts such as blocking or ringing. In order to deal with this trade-off, coupling filter and encoder is suggested, an a mechanism for obtaining the proper amount of filtering from the expected coding distortion is proposed. As will be described later, there are some reasons for filtering the motion-compensation residue rather than the original or the reconstructed image, but the main argument is an easier coupling between rate control and filter control.

In-loop post-filter

Probably the most broadly-used filter techniques are those applied as a post-processing to the reconstructed sequence, particularly because of the popularity achieved by deblocking filters like the one described in Section 2.3.3. However, it is to be noticed that placing a post-processing filter after the reconstruction of each frame in the sequence could not be considered as a simplification procedure, but an enhancement stage for the experience of observers. On the other hand, if the filter is placed in-loop, i.e., in a way that reconstructed frames to be used as references for inter prediction are previously filtered, it is possible to reduce blockiness in reference frames and consequently some temporal artifacts in the subsequent frames, or even to improve the overall rate-distortion performance.

The main drawback of this kind of approaches is that in-loop filters must be applied in both the encoder and decoder side in order to reconstruct identical reference frames for the operation of motion compensation. This means that any filter configuration must be known *a priori* in both sides or must be communicated from the encoder to the decoder. Given the outstanding quality improvement demonstrated, deblocking filter was incorporated in the H.264/AVC standard as a normative part of the decoding process.

Furthermore, with the increase in computational resources boosted by the development of multi-processor architectures, some more complex techniques has been proposed for an adaptive in-loop filtering of the reconstructed images in hybrid video codecs, mainly oriented to the coming HEVC standard (see [Narroschke, 2010] or [Siekmann et al., 2010]). Instead of dealing with a particular artifact, these techniques aim at reducing the pixel-wise differences between original and reconstructed images by means of an adaptive Wiener filter, and sending the filter coefficients to the decoder. The effect of these general-purpose correction filters comes in the form of an overall rate-distortion improvement, even with the drawback of sending some filter parameters to the decoder side. They can also be combined with the deblocking filter.

5.3 Proposed algorithm: residue filtering for video simplification

A novel filtering scheme for video sequence simplification was proposed based on the previous works described in Section 4.2 and with the objectives outlined in Section 5.1. Apart from the works found in the literature, the proposal described in [de Frutos-Lopez et al., 2012] and summarized in Section 5.2.1, was our first on video filtering for simplification, and some of its conclusions grounded the design decisions made in this PhD Thesis.

The main objective of the algorithm is to reduce the amount and strength of low bit rate artifacts, while keeping the blurring effect consequence of the smoothing filters under control. To this end, a good estimate of the visibility of these artifacts should be constructed. Although the blocking effect is the main artifact related to low bit rate coding and it is relatively easy to detect by pixel-wise measures (like the one described in [Lin and Ortega, 1997]), other artifacts have also a considerable influence in the perceptual quality of the encoded video and their appearance is not so easy to detect. Therefore, it is necessary to find a distortion metric or estimation nearer the subjective perceptual quality.

Some of the solutions described in Section 4.2 rely on a RD decision to determine the proper amount of filtering. On the other hand, as described in Section 2.3.5, typical RC algorithms designed for H.264/AVC and based on empirical models estimate the QP value for a basic unit before encoding it, avoiding the *chicken-and-egg* dilemma and, therefore, decoupling the task of the RC from the RD problem. In the same way, the amount of filtering could be calculated prior to the encoding process by means of a model relating QP and filtering.

Finally, pre-filtering schemes benefit from a certain independence of the encoding process itself, making them appropriate regardless of the video codec to be employed. However, this advantage could turn to a drawback considering that a filter control mechanism is necessary in order to deal with the trade-off between coding artifacts and blurring, as firstly suggested in [Lin and Ortega, 1997]. This mechanism could be more precise by filtering of prediction residue and, as described in the following section, other benefits are to be expected.

5.3.1 Rationale of Residue Filtering

Our previous experience described in [de Frutos-Lopez et al., 2012] highlighted the importance of integrating the pre-filtering stage more deeply in the encoder architecture, specifically after the motion compensation. There is a number of potential benefits of this technique, some of them suggested by [Segall et al., 2001]:

1. The prediction residue is the only information to be quantized and therefore its encoding is the only source of artifacts in absence of channel errors. Those pixels properly predicted by the motion-compensation are not relevant from

the filter point of view, given that the encoding process will deal with them without noticeable distortion.

2. Filtering the original image does not always imply a bit rate reduction. For example, in static regions of Inter frames, filtering original pixels could even make the prediction worse and increase the bit-rate. On the other hand, when filtering the residue, its transform will show less high-frequency components. This kind of direct relationship between bit-rate reduction and filter strength makes easier to find the proper amount of filtering for a certain situation.

3. While in-loop post-filter operations must be shared by encoder and decoder, residue filtering can be done with independence of the decoding process.

4. The filter control and the rate control can be integrated in a unique control of QP and filter strength.

5. Integrating residue filter and RDO mode decision allows for predicting whether a filtered residue block would achieve a better R-D performance than its corresponding non-filtered version.

6. When filtering the original image, if the proper amount of filtering is selected by an iterative process, motion estimation should be repeated for each iteration. On the other hand, filtering the residue with different filter parameters does not entail repeating the motion estimation, which is a previous step to obtain the actual residue of the block.

7. Residue filtering makes possible to filter successive frames with different filter strengths without noticing any artifact. This is not the case when the original frames of the sequence are filtered, in which a different amount of filtering, apart from increasing the differences between a frame and its references, it may entail the appearance of flickering artifacts.

Of course, there are also some advantages of the traditional pre-filtering scheme and drawbacks of the residue filtering approach:

1. Residue filtering can not be applied off-line (outside of the encoding process)

2. When applied too coarsely, residue filtering tends to alter the mode decision in a way that the number of skipped blocks increases due to the simplification of the prediction residue (one of the conditions for selecting a block as SKIP block is that it could be sent without any residue). This affects in the same way that a higher λ_{mode} parameter in mode decision would do, and an increase in the number of skipped blocks may appear. In this way, the encoder could decide for a short-term bit rate reduction at a cost of worsen the reference frames for future motion compensation.

3. Given the lack of correlation between residue patterns of neighboring blocks, residue filter support should be limited to the actual block size selected for motion-compensation and each residue block should be filtered separately. On the other hand, filtering the original pixels can be accomplished with larger filter masks and without taking care of any block boundary.

4. When working on original frames, temporal pre-filters can be extended to both previous and future frames (if a certain delay is accepted), while it is impossible to access to future residue samples with the residue filtering approach.

There are some reasoning to overlook the abovementioned problems. First of all, one of the characteristics of the particular target application states that the filtering algorithm should be capable of running together with the encoder making use of the rate control information as far as possible, so the first drawback is not relevant. Second, the nature of the motion-compensation prediction suggests that there is not a high correlation between residue samples of adjacent blocks, then there is no need of large filter supports. In the same way, the correlation between consecutive residue frames is also lower than the correlation between the corresponding consecutive original frames and there would little gain in employing temporal supports. Finally, a better control of the impact of filtered information can be achieved by filtering the residue, given that it actually simplifies the data to be sent to the decoder. This is of

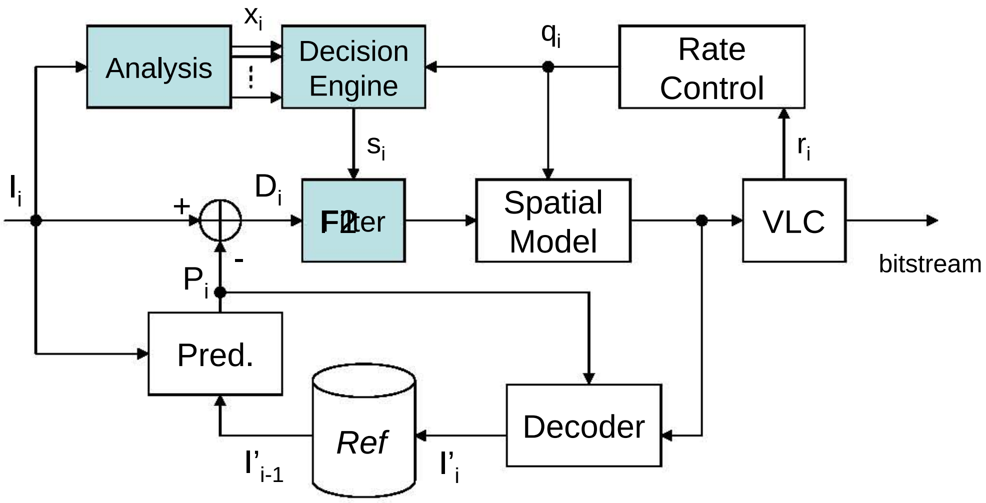


Figure 5.2: Architecture of the proposed algorithm.

paramount importance in the design of a filter control coupled with the rate control algorithm.

5.3.2 System Architecture

As a result of the discussion in previous section, the proposed scheme is shown in Figure 5.2, in which the filtering stage is placed after the calculation of the prediction residue in order to integrate filtering and encoding processes.

However, according to the RDO mode decision performed in the H.264/AVC video codecs and described in Section 2.3.4, two options for placing the residue filtering stage were considered. The first one involves performing the mode decision independently of filtering: the best mode for motion compensation is decided according to the R-D performance and subsequently the motion-compensated residue of the best mode is filtered.

The second option is the integration of the filter into the mode decision engine, filtering the motion-compensated residue corresponding to each block size being tested prior to the calculation of rate and distortion for mode decision. Obviously, the mode decision corresponding to this second option is expected to be more accurate given that the R-D performance data used for deciding the block size matches up with the actual values to be observed at the decoder. On the other hand, the cost of the filtering stage would be consequently multiplied by the number of possible partitions per macroblock.

Faced with the risk of performing mode decisions with non-real data and being aware of the computational cost of filters, the second option was chosen: the filtering is performed independently for each block and block size to be tested in the RDO mode decision. As a first consequence, given that an H.264/AVC encoder incorporates intra-predicted blocks within inter-coded frames, it is necessary to apply the filter also to the intra prediction residue. On the contrary, the inter-intra mode decision would not be fair and an inadequate rise of intra modes could be observed. Filtering the residue limits the extent of the filtering support to the size of the predicted block. In this way, for *Intra16x16* and *Inter16x16* modes, in which there exist a correlation between samples for the entire macroblock, the filter will be applied to the entire residue macroblock, while for the rest of the modes it is applied on a 8x8 pixel-block basis.

Concerning the filter algorithm itself, after assessing different configurations, a Gaussian smoothing filter with a square support of 5x5 pixels was selected due to its simplicity and ability to reduce high frequencies in residual information. The details of its configuration and the process of filter selection will be discussed in Section 5.3.3.

Finally, a filter control was placed parallel to the encoding process that analyzes video sequence and decides on the proper filter strength based on the expected coding artifacts for the available bit rate. Given that the visibility of video coding artifacts is directly related to the QP employed for residue quantization and with the spatiotemporal characteristics of the video sequence, these parameters help to decide the degree of smoothing to be applied. In order to tune this control algorithm for filter control, some subjective quality assessments over a set of training sequences at different bit rates and filter strengths were employed, as described in Section 5.4.

5.3.3 Filter configuration

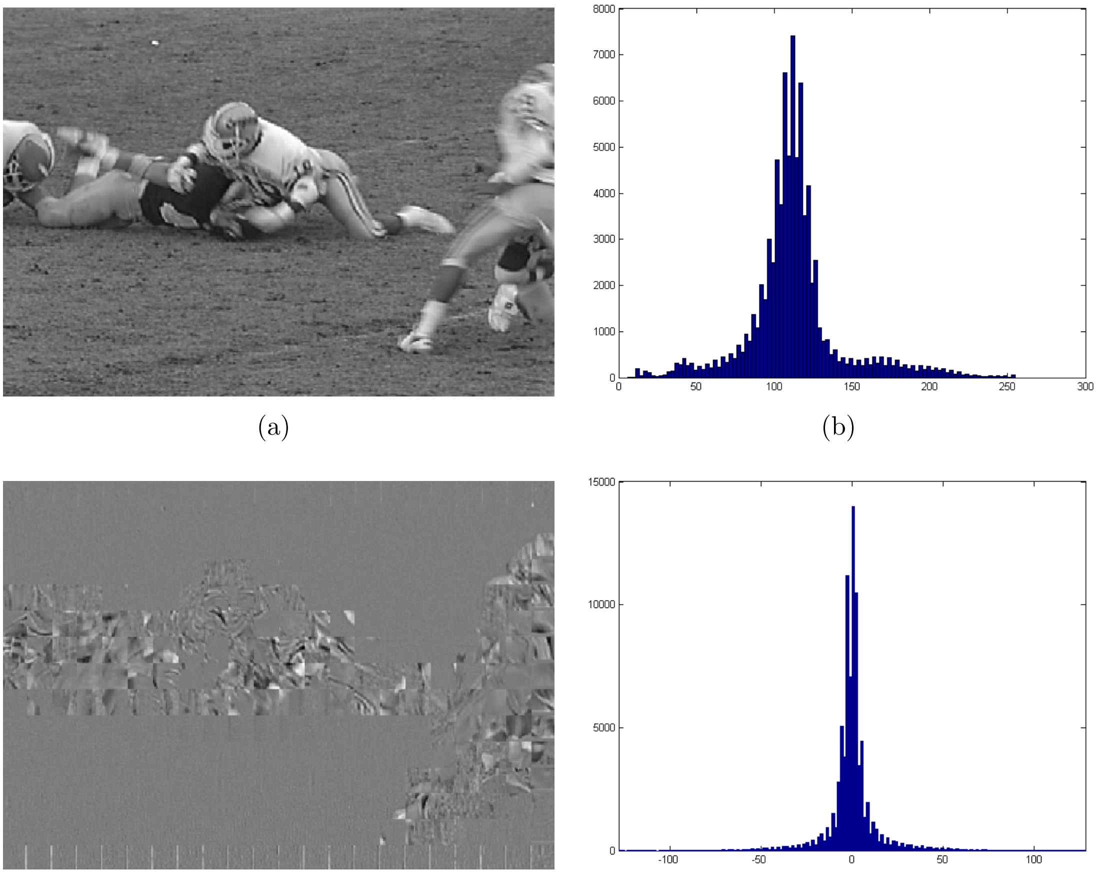
The core of the proposed algorithm is the filtering stage. According to the architecture shown in Figure 5.2, each motion-compensation or intra-prediction residue block (D_i) is simplified facilitating the encoding process due to the reduction of the high frequency components. Although almost every filtering scheme of those described in Section 4.2 could be employed for the task of video simplification, the particular characteristics of the residue signal as well as the characteristics of the H.264/AVC standard should be taken into account to select the type and configuration of the filtering algorithm.

Residue statistics

In Figure 5.3.3, a luminance frame of the sequence *football* has been depicted together with its motion-compensation residue. In order to highlight the differences in the statistics of these data, the corresponding histograms of the original image and the residue are also included. As can be observed, the prediction residue is a zero-mean signal with double dynamic range than that of the original image, but more concentrated around the mean value. As a way of an example, the variance relative to the histogram in Figure 5.3(b) is around 950, while the variance corresponding to the residue and histogram in Figure 5.3(d) is 245. Residue values also exhibit a lower entropy.

As can be seen in Figure 5.3(c), the higher residue values are concentrated on those objects appearing for the first time in the frame (as it is the case of the football players entering in the framing from both sides), and in the edges of moving objects, given that these areas are the most difficult to predict.

Given that motion estimation is performed independently for each macroblock, even more for smaller macroblock partitions up to 4x4-pixel blocks, residue samples in a non-skipped block are independent from those in neighboring blocks or in co-located blocks from previous or subsequent frames. For example, in Figure 5.3(c), the macroblock boundaries are evident due to the differences between the encoding modes of neighboring blocks.



(a)

(b)

(c)

(d)

Figure 5.3: Analysis of motion compensation residue for the 2_{nd} frame of the sequence *football*: (a) Original frame; (b) Histogram of luminance values; (c) Residual frame; and (d) Histogram of residue values.

It should not be forgotten that the objective of the simplification stage is to lighten the subsequent encoding process. The H.264/AVC hybrid coding includes a spatial model stage after motion compensation that, as discussed in Section 2.3.4, performs the DCT plus a quantization stage and a zig-zag reordering of residual block samples to conform a vector of quantized transform coefficients. This vector is then entropy encoded and packed in the bitstream structure.

Analyzing the process inversely, the entropy encoder deals efficiently (in terms of produced bits) with vectors containing some relevant coefficients in the first positions and small values or zeros in the rest of the elements. On the contrary, vectors with higher values in its last positions imply important bit expenses. This means that the transformed block prior to the zig-zag reordering should not show relevant coefficients apart from those corresponding to low frequencies, the most top-left samples in the transformed block. Therefore, in order to lighten the encoding process and save a considerable amount of bits, a smoothing filter should be applied to the residual samples for reducing high-frequency components in the transformed block.

Filter pre-selection

Given the particular conditions of the scenario, a certain number of considerations were made for defining a first set of candidate filters:

- The particular configuration of the rate-distortion mode decision in H.264/AVC motivated the inclusion of the filter mechanism after the residue calculation of each evaluated mode. This constitutes an important complexity constraint given the relatively large amount of modes to be tested. Even avoiding the use of inter modes smaller than 8x8-pixel blocks, there are still 4 inter modes to be tested (16x16, 16x8, 8x16 and 8x8) together with the corresponding intra 16x16 and 8x8 modes, which means that each macroblock is filtered 6 times. Therefore, the most complex approaches from the filter mechanisms described in Section 4.2 must be rejected.

- In order to integrate the filter into the encoding process, a mechanism for filter strength control is mandatory and the fixed averaging filter masks should be avoided.

- As can be seen in Figure 5.3(c), the low correlation of the residue values with respect to those in neighboring blocks (because they use different encoding modes or exhibit different texture patterns), prevent against the use of large filter supports or multi-scale analysis, as well as against the NLM filters.

- In real-time implementations of the H.264/AVC encoder, residue data corresponding to subsequent frames can not be calculated. Temporal supports extending to subsequent frames are therefore avoided. Moreover, the low correlation of the residue values with respect to those in previous and subsequent frames advise against any temporal filtering.

Still there are some simple filter approaches that comply with the abovementioned constraints. The basic Gaussian smoothing filter, that has already demonstrated to be useful for video simplification in previous works as [Lin and Ortega, 1997] or [Segall et al., 2001], was our first candidate. The system could benefit from its simple structure and the filter strength could be tuned with only a single parameter: the variance of the bell-shaped weight map.

Of course, given that certain details in residue blocks need to be preserved in order to properly reconstruct each image, a certain edge preserving potential is desirable in the filter mechanism. For this reason, median filter was also included as a candidate filter. For a small support, the operations needed for data ordering imply only a moderate increase in computational cost.

In the same sense, bilateral filter could be counted as a candidate. It combines the smoothing potential of a spatial domain filter with the edge preserving component of the range filter. The use of two parameters for controlling filter strength introduces an additional degree of freedom for filter tuning.

Due to its relative simplicity, the gray scale morphological operators could also be considered. As described in 4.2.2, these filters are mainly employed in applications related to pattern recognition and for the construction and processing of binary maps.

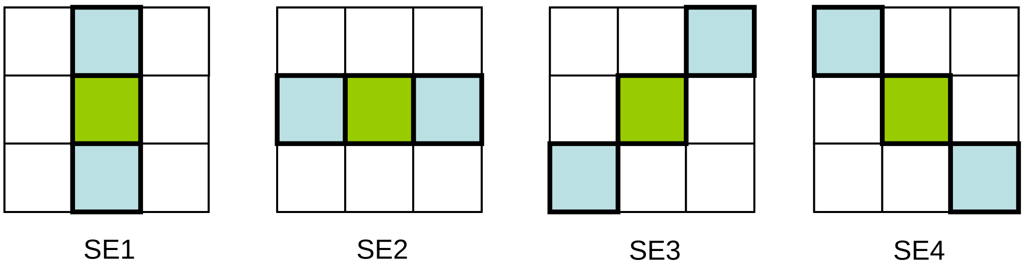


Figure 5.4: Directional SE for erosion filtering.

In any case, given the structure of residual information, the ability of the closing operation to remove high isolated values could be of some help for reducing high-frequency components. The erosion operator could also be considered as a candidate.

In summary, the filter algorithms selected as candidates for simplifying video residual were the Gaussian (G), Median (M) and Bilateral (B) filters together with Erosion (E) and Closing (C) morphologic operators. Different filter supports were tested as structural elements of the morphologic operators: the square supports of sizes 3x3 and 5x5, together with some directional structural elements depicted in Figure 5.4. For Gaussian and Bilateral filters, 3x3 and 5x5 masks were employed, although the 3x3 support could imply for some values of the filter strength a significant drop in the filter response at the support borders, that may introduce artifacts.

Several values of standard deviation were employed for the Gaussian filter configuration, from $\sigma_g = 0.1$ to $\sigma_g = 10$. These same values were employed for the standard deviation parameter of the domain filter component σ_d in the Bilateral filter, together with different versions of the range filter component with σ_r values ranged from 0.1 to 15.

Some initial tests were carried out for assessing the effects of these filters in the residual data. First of all, a set of typical video sequences was encoded using the H.264/AVC reference software version JM18.0 available on-line[1] with QP=20. The motion-compensation residue of the first inter predicted frame was employed for our experiments to determine the potential for residue smoothing of the different filter

[1] K.Sühring. H.264 software coordination. http://iphome.hhi.de/suehring/tml/download/old_jm/

Filter type:	NF	G5x5	B5x5	M3x3	M5x5	E3x3	C3x3
Residue variance:	208	119	119	154	121	52	204

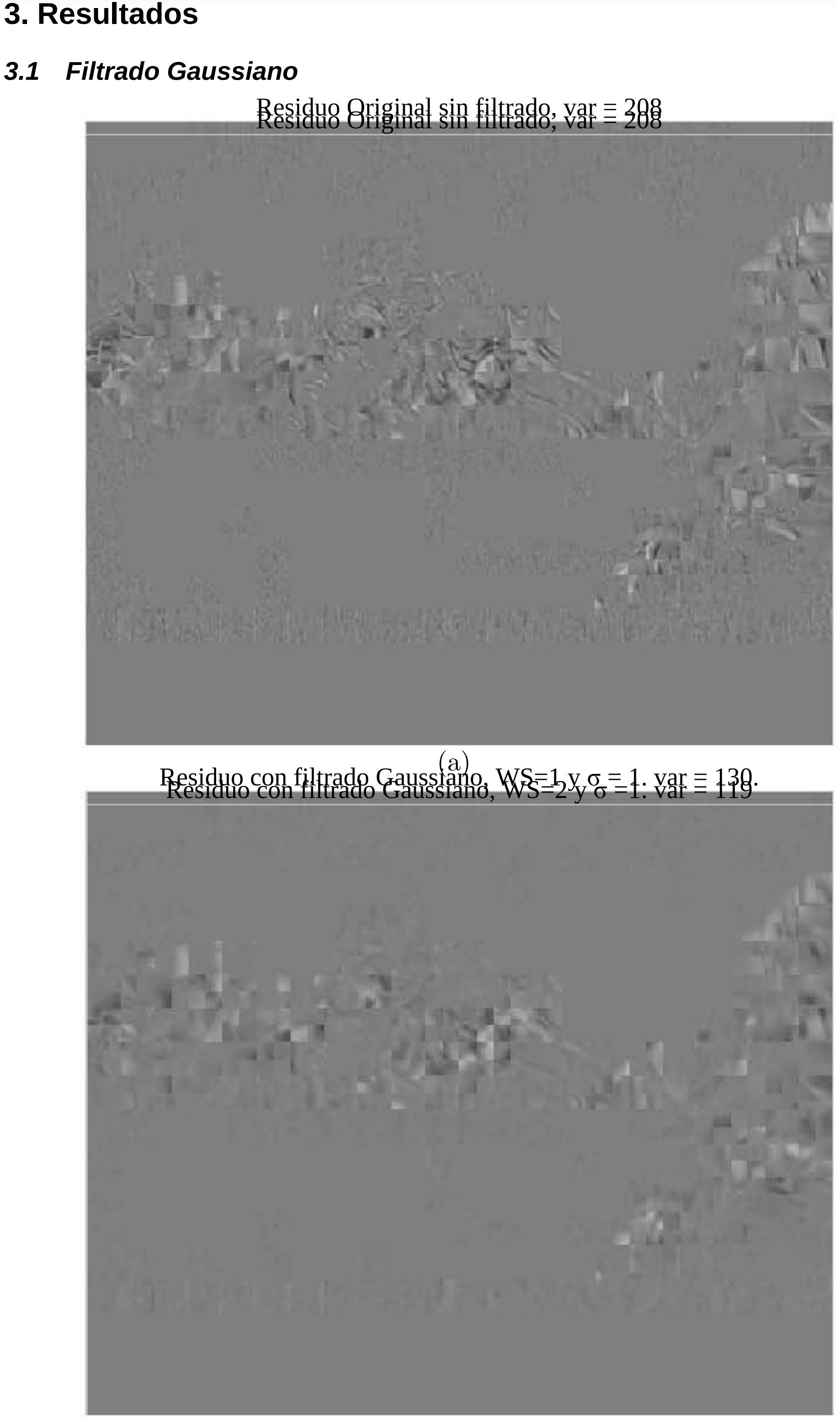
Table 5.1: Variance of residue samples for frame number 2 in *football* sequence for different filter algorithms.

options. Filtering was applied to 8x8-pixel blocks and the resulting variance of the residue signal was calculated for the entire frame. It is to be noticed that the variance itself has nothing to do with the quality performance of the filter, but with the ability to reduce the bit rate. The lower the residue variance after filtering, the more bit savings. But concerning quality comparison, other measures were considered given that the filtering process could introduce an important distortion.

Table 5.1 lists the corresponding values of variance of the residue with the different filters. In this case, Gaussian 5x5 filter (G5x5) was configured with $\sigma_g = 1$, and Bilateral 5x5 filter (B5x5) employed $\sigma_d = 1$ and $\sigma_r = 1$. Figures 5.5 to 5.8 show the motion-compensation residue frames before and after filtering with some of the mentioned configurations.

From the data in Table 5.1, it can be seen that Gaussian and Median filters achieved a similar variance reduction for the same 5x5 mask size, although the resulting distribution of residue samples is quite different, as can be seen in Figures 5.5 and 5.6. For this reason, the behavior of the spatial model in the H.264/AVC encoder was expected to be also different with both filters. In the same way, Bilateral filters performed similarly to Gaussian filters when $\sigma_d = \sigma_g$ and large σ_r values were employed; however, as σ_r was reduced, the smoothing strength of the bilateral filter dropped, and higher values were achieved for the variance of the residue, reducing the potential bit rate savings.

Concerning the morphological operators, the erosion filter achieved, for the square 3x3 SE, the best variance reduction due to an aggressive pruning of those residue samples surrounded by zeros, as can be seen in Figure 5.7. This also means that it removes a great amount of information and some way of controlling this filter is mandatory to employ an erosion filter as a simplification algorithm. The results also


3. Resultados
3.1 Filtrado Gaussiano
Residuo Original sin filtrado, var = 208
Residuo Original sin filtrado, var = 208
Residuo con filtrado Gaussiano, WS=1 y σ = 1. var = 130.
Residuo con filtrado Gaussiano, WS=2 y σ =1. var = 119

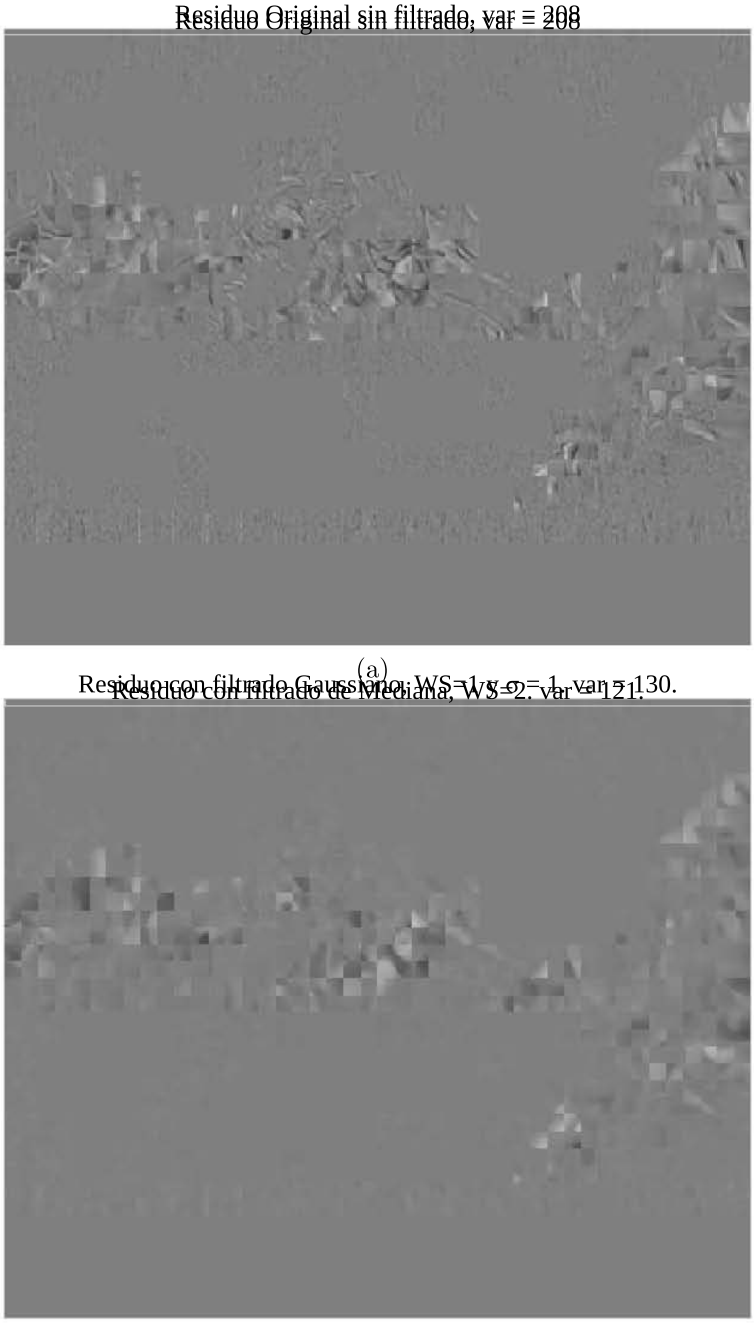
(a)

(b)

Figure 5.5: (a) Motion-compensation residue for $2nd$ frame of *football*; and (b) G5x5 filtered version (with $\sigma_g = 1$).

3. Resultados

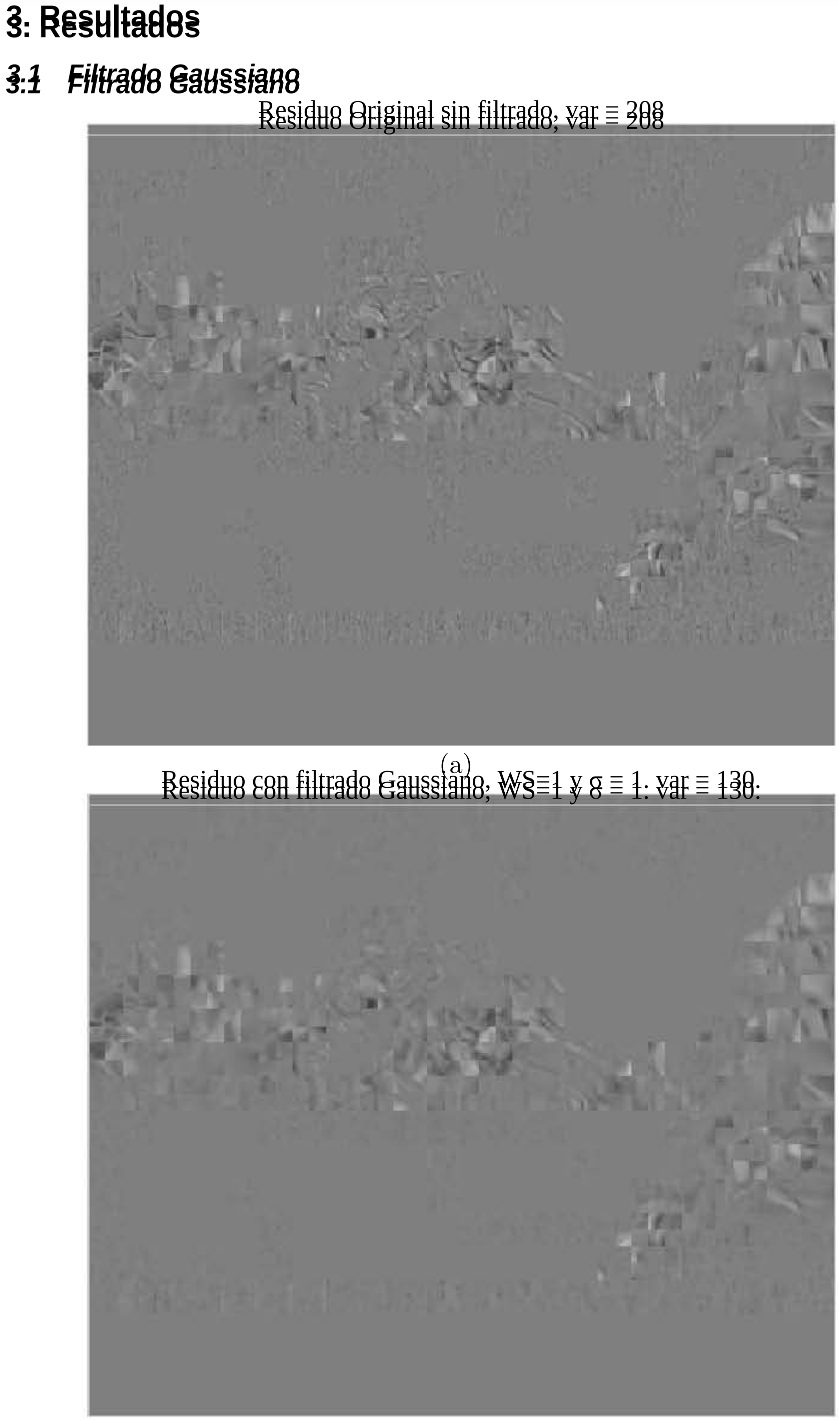
3.1 Filtrado Gaussiano


Residuo Original sin filtrado, var = 208
Residuo Original sin filtrado, var = 208
Residuo con filtrado Gaussiano, WS=1 y σ = 1, var = 130.
Residuo con filtrado de Mediana, WS=2. var = 121.

(a)

(b)

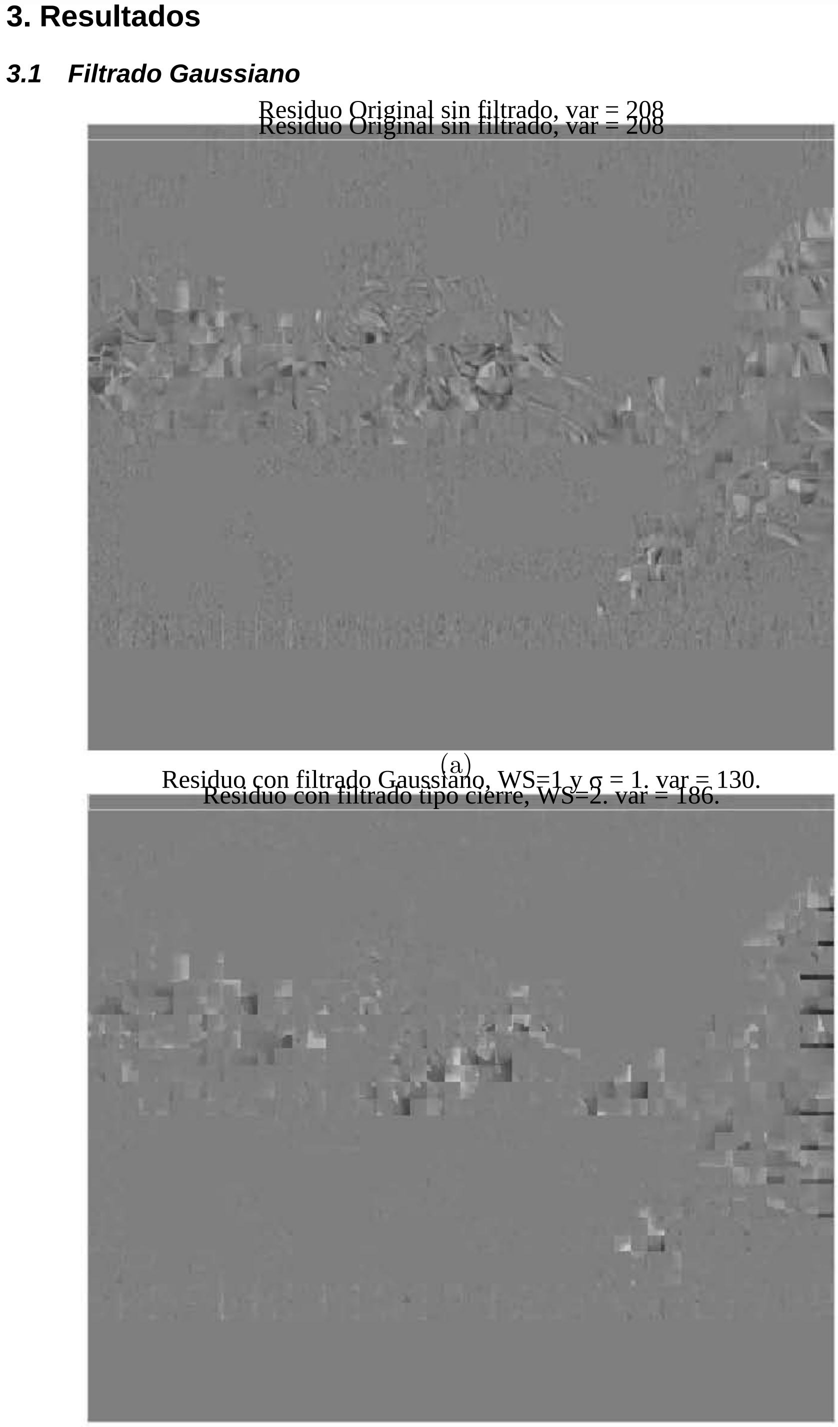
Figure 5.6: (a) Motion-compensation residue for $2nd$ frame of *football*; and (b) M5x5 filtered version.


3. Resultados
3.1 Filtrado Gaussiano
Residuo Original sin filtrado; var = 208
Residuo con filtrado Gaussiano, WS=1 y σ = 1: var = 130.

(a)

(b)

Figure 5.7: (a) Motion-compensation residue for $2nd$ frame of *football*; and (b) E3x3 filtered version.


3. Resultados
3.1 Filtrado Gaussiano
Residuo Original sin filtrado, var = 208
Residuo Original sin filtrado, var = 208
Residuo con filtrado Gaussiano, WS=1 y σ = 1. var = 130.
Residuo con filtrado tipo cierre, WS=2. var = 186.

(a)

(b)

Figure 5.8: (a) Motion-compensation residue for $2nd$ frame of *football*; and (b) C5x5 filtered version.

showed that the closing filter (C5x5) achieved a certain variance reduction at a cost of some artifacts in the residue structure that in the end could make it unfit for our task of residue simplification, as can be seen in Figure 5.8.

To this point, some conclusions were ventured about the behavior of the different filter approaches for residue simplification at the sight of the filtered residue values and its variance. However, as mentioned before, a reduction of the variance of the residue samples does not necessarily guarantee a good performance of the filters when working as part of the proposed framework. For this reason, the abovementioned algorithms were integrated into the H.264/AVC reference software for a complete testing. As described in Section 5.3.2, the filtering stage was implemented after the prediction carried out by the encoder for each possible mode (inter or intra) and before making the RDO mode decision. From the complete set of block partitioning modes for Inter coding, only the 16x16 size was employed in this first approximation. Together with the Intra modes with block sizes 16x16 and 8x8, and the SKIP mode, they conformed the pool of enabled modes for these experiments.

The rest of the encoder configuration parameters employed in the tests were:

- IP...P pattern with 100 frames at 25 fps (only the first frame is an Intra)
- Rate-distortion Optimized decision
- Rate Control ON: CBR at 128/256 kbps
- 8x8 DCT
- One reference frame for motion estimation
- CIF (352x288) test sequences with different spatial and temporal complexities

The figure of merit selected to compare the different filter configurations should have been the quality observed at a fixed bit rate, with the rate control mechanism ensuring that the bitstream size is approximately the same for all the configurations. Unfortunately, as discussed in 2.4, the problem of quality assessment is far from being solved. In later tests, subjective scores were introduced as the figure of merit but, concerning filter selection, other indexes listed next were employed:

- The PSNR is employed as an estimate of subjective quality in many applications due to its easy implementation. However, it performs poorly for the particular task of assessing the quality of filtered video, given that the general opinion score at the view of a blurred image can be considerably higher than what its PSNR suggests. Therefore, the PSNR score was calculated but taken with caution in these experiments.

- The average QP was taken into account given that high QP values rise the risk of low bit rate artifacts like blocking, ringing, etc.

- Finally, the distribution of the block modes was also be monitored in order to verify that the filtering process does not substantially influence the mode decision.

With the help of these encoding results, an additional refinement of the list of filter candidates was performed. The first consequence was that some filters, whose variance tests were promising, appeared as a poor option for residue simplification according to the encoding results. The distortions introduced in a frame due to the residue filter turned it into a bad reference for the subsequent picture in the sequence, worsening the performance of its motion compensation and increasing the energy or the residue. The short-term bit rate savings in a frame come at a cost of higher residue values in the next frame. This *greedy effect* is observed in some filters like the median filter, and others with a strong residue reduction, like the 5x5 erosion filter, and advises against their use for residue simplification. That was the reason for excluding this configurations from the candidate list as soon as the first encoding results came into light. The second consequence of dealing with real encoding results was the discarding of the erosion filters with directional SE depicted in Figure 5.4, due to their low performance when embedded into the pre-filter framework.

Final results and decision

Tables 5.2 and 5.3 show the results at 128 kbps for the reduced set of filter techniques derived from previous analysis. As shown in Table 5.2, the filtering stage reduces

	No Filter $PSNR(dB)$	Gaussian $\Delta PSNR$	Bilateral $\Delta PSNR$	Erosion $\Delta PSNR$
bus	24.14	-0.52	-0.47	-1.3
coastguard	27.13	-0.65	-0.56	-1.3
football	22.8	-0.12	-0.08	-0.68
foreman	31.63	-0.56	-0.5	-1.68
mobile	22.42	-0.29	-0.25	-1.06
paris	27.76	-0.97	-0.86	-1.63
stefan	24.8	-0.6	-0.58	-2.01
tempete	26.06	-0.52	-0.47	-1.54

Table 5.2: Average PSNR values for the reference non-filtered version and the corresponding PSNR difference values for the candidate filters and test sequences at 128 kbps.

the reconstructed average PSNR of the video sequence regardless of the employed algorithm, as expected with this kind of measures. Nevertheless, probably the most interesting information is in Table 5.3 (and also in Table 5.5, which will be presented next), the average QP obtained for the different filter options. The lower the QP values for the tests at 128 kbps, the lower the risk of harmful artifacts such as blocking, ringing, etc. As can be seen in Table 5.3, the higher QP reductions (up to more than 4 units) can be achieved with the erosion filters for the complete set of sequences. However, the visualization of reconstructed sequences highlights the risk of certain encoding artifacts around edges in moving objects[2].

As can be seen in Table 5.5 for the tests at 256 kbps, the average QP values (the operating points of the encoder) are reduced in about 4 or 5 units with respect to the 128 kbps tests for the same filter versions. This means that the risk of quantization artifacts is also reduced when the available bit rate is doubled. Therefore, obtaining a QP reduction at this rates does not necessarily improves the subjective quality of reconstructed sequences, and the aggressive filters that were suitable for 128 kbps become unfit for working at 256 kbps, except for certain sequences with very high

[2] These reconstructed sequences can be found in http://www.tsc.uc3m.es/~mfrutos/Thesis/FilterSelection2/ for Gaussian 5x5 filter (G5x5), bilateral (B5x5) 5x5 filter, erosion 3x3 filter (E3x3), and reference non-filtered version (NF), at 128 and 256 kbps.

	No Filter QP	Gaussian ΔQP	Bilateral ΔQP	Erosion ΔQP
bus	44.15	-2.9	-2.8	-4.4
coastguard	39.92	-3.3	-3.1	-4.7
football	47.09	-1.8	-1.7	-3.1
foreman	37.05	-2.6	-2.5	-3.6
mobile	44.14	-2.8	-2.8	-3.4
paris	39.87	-4.1	-4	-4.8
stefan	43.54	-2.6	-2.5	-3.7
tempete	40.29	-2.8	-2.8	-4.3

Table 5.3: Average QP values for the reference non-filtered version and the corresponding QP difference values for the candidate filters and test sequences at 128 kbps.

	No Filter $PSNR(dB)$	Gaussian $\Delta PSNR$	Bilateral $\Delta PSNR$	Erosion $\Delta PSNR$
bus	26.87	-1.22	-1.11	-2.51
coastguard	28.98	-1.2	-1.07	-2.18
football	24.98	-0.61	-0.55	-1.6
foreman	34.25	-1.5	-1.35	-3.2
mobile	25.13	-1.36	-1.26	-2.45
paris	30.18	-2.25	-2.04	-3.13
stefan	28.15	-1.75	-1.6	-3.87
tempete	28.35	-1.36	-1.28	-2.47

Table 5.4: Average PSNR values for the reference non-filtered version and the corresponding PSNR difference values for the candidate filters and test sequences at 256 kbps.

	No Filter	Gaussian	Bilateral	Erosion
	QP	ΔQP	ΔQP	ΔQP
bus	39.6	-4.5	-4.3	-4.9
coastguard	36.7	-4.3	-4.1	-5.3
football	42.9	-3.4	-3.3	-4.8
foreman	32	-3.2	-3.3	-3
mobile	39.4	-4.5	-4.4	-4.2
paris	35.6	-4.7	-5	-3.2
stefan	38.2	-3.9	-3.7	-2.7
tempete	36.3	-4.6	-4.4	-4.9

Table 5.5: Average QP values for the reference non-filtered version and the corresponding QP difference values for the candidate filters and test sequences at 256 kbps.

motion complexity like *football* or *bus*, in which the average QP is still high enough for causing harmful artifacts.

It is to be noticed that there is a certain deviation of the output bit rate with respect to the target bit rate imposed to the rate control algorithm when some filters are used. While this deviation is under 1% for the non-filtered version, the rate control algorithm fails to predict the amount of bits and a $2 - 3\%$ mismatch is observed for strongly filtered residues in some sequences. If this mismatch means a problem for certain scenarios, a possible solution could be to redefine the R-Q model in the rate control in order to take into accoung the amount of filtering.

Concerning the differences between filters, for all but some particular cases, the erosion filter applied to the residue frame tends to introduce severe blurring artifacts in the reconstructed sequence. The QP reduction achieved, although suitable for reducing low bit rate artifacts, comes then at a cost of an excessive blurring. The subjective results are much better for the gaussian and bilateral filters.

As also shown in these and previous tests, the response of bilateral filters depends highly on σ_r, the range filter variance. When this parameter is high enough, bilateral filters tend to perform like Gaussian filters and, when σ_r is decreased, they tend to filter too softly and its behavior is more similar to that of the non-filter configuration.

Could be this property of the range filter variance employed to control the amount of filtering? The answer is no. Range filters are incorporated to bilateral filters in order to preserve edges in the image, an invaluable behavior for certain scenarios but unsuitable for the task of residue filtering. The simplification introduced in the residue frame drops fast when smaller values of σ_r are employed and it becomes difficult to select the appropriate combination of σ_d and σ_r. It is easier to perform a Gaussian filtering and to control the filter strength with a single σ parameter.

The theoretical analysis of the different filter mechanisms, the particular conditions of the scenario, and the successive tests carried out for assessing the filters performance, suggest the use of the Gaussian filter for simplifying the residue. A complete summary of the reasoning is stated here:

- Gaussian filters are simple enough for being implemented into a real-time encoding engine. They can even be separated in two 1D linear filters to speed up the filtering process.

- The filter strength can be controlled by a unique parameter σ for a fixed support size, which is an important advantage for integrating a filter control into the rate control mechanism. Other filters have more than one parameter to control (like bilateral filters) or need to be adjusted by changing the filter support adaptively (like median or morphologic filters).

- Linearly smoothing the residue is a good way for reducing the encoder output bit rate and its corresponding reconstructed video sequence exhibits less artifacts than reconstructed sequences filtered with morphological or median filters.

- Bilateral filters combine a Gaussian domain filter with a range filter for preserving edges, but the residue is basically an edge signal and the bit rate savings drop abruptly as the range filter strength becomes important.

- The benefit obtained from filtering residue frames with a certain configuration at 128 kbps is not the same as in 256 kbps. Filter strength should be adapted at least to the quantizer for a proper trade-off between blurring and coarse-quantization artifacts.

5.4 Filter control

The objective of this block is to properly select the filter strength for adapting the filter to the subsequent spatial model (DCT+quantization) in the hybrid encoder. This adaptation can be performed by the proper selection of the σ_g parameter (denoted simply as σ from now on) in the Gaussian filter selected as the core of the proposed framework. As observed in previous section, the proper amount of filtering depends on the characteristics of the sequence and the target bit rate, in the search of a trade-off between blurring and low bit rate artifacts.

In this section, a procedure to control the filtering strength is proposed that obtains the best σ value for a given sequence and bit rate from a perceptual quality point of view. In the next sections, the problem of perceptual quality assessment will be addressed and a solution will be proposed for obtaining the optimum σ value by means of a linear regression procedure. The extensive testing of the final solution and the corresponding conclusions will be discussed later in Sections 5.5 and 5.6.

5.4.1 The Perceptual Distortion

Some extensive tests were performed in order to find a relationship between the filter strength and the perceived quality in the reconstructed sequences. The objective is to employ this relationship for obtaining a proper value of the filter strength for each sequence and target bit rate. To this end, up to 18 CIF video sequences were encoded at 4 different target bit rates (128, 256, 384 and 512 kbps) for a total of 72 test samples. After motion-compensation of each frame, and prior to encoding the prediction residue, these sequences were filtered with 5 different values of the Gaussian filter strength (0, 0.6, 1, 1.25 and 1.5) in order to obtain an approximation of the best σ value for each sample (sequence and rate). The quality of the resulting reconstructed sequences was assessed by determining the point of minimum distortion

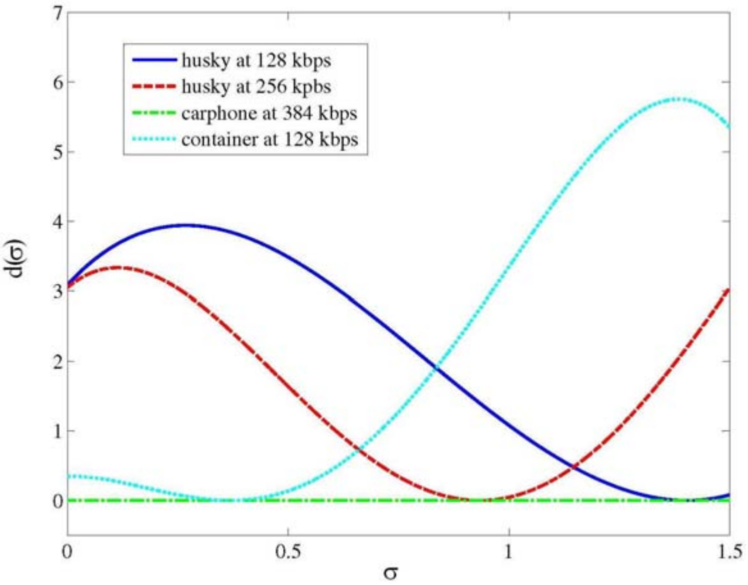


Figure 5.9: Examples of subjective distortion curves $d(\sigma)$ with respect to filter strength σ.

σ_{opt} and then classifying the rest of the versions according to the amount of distortion with respect to that of σ_{opt}. The following qualitative scale was used whose right side values correspond to the actual distortion score assigned:

Minimum distortion (σ_{opt})	-	0
Imperceptible	-	0
Perceptible but not annoying	-	1
Annoying	-	3
Very annoying	-	5

The results show that the common behavior of the system at low bit rate is a reduction of the harmful artifacts such as blocking, ringing or mosquito as σ is increased until a certain point in which the blurring effect becomes evident and the visual quality drops. In Figure 5.9, four distortion curves corresponding to different test samples are depicted. Each sample corresponds to a certain sequence and bit rate, and each curve was obtained by interpolating the subjective distortion scores at the selected filter operating points (σ values). An offset has been added to the curves in order to ensure non-negative values. For this reason, although the nominal values of quality are in the range from

0 to 5, some values in Figure 5.9 are greater than 5.

As can be seen, the $d_j(\sigma)$ curves are different for every sample j, but all of them exhibit a unique minimum value (or a unique region in which the minimum value is likely to be found, like in *carphone* at 384 kbps, in which all the filtered and unfiltered versions are indistinguishable). Therefore, the differences between distortion curves are mainly related to the position of the minimum. For example, certain complex sequences exhibit noticeable coarse-quantization artifacts at low bit rate that can be alleviated by the use of a higher σ in the filtering stage, admitting a more strong filtering and then displacing the minimum distortion to σ_{opt} values up to 1.5, like observed in the Figure 5.9 for *husky* sequence at 128 kbps. On the other hand, working at higher bit rates and/or with simpler sequences displaces the minimum distortion to smaller values of σ or even to 0, like in Figure 5.9 for *container* at 128 kbps. Moreover, in some cases like in *carphone* at 384 kbps, it is not possible to determine a unique value of σ_{opt} (any σ value would do the job), while in some others the distortion abruptly rises when the amount of filtering is increased or decreased with respect to the point of minimum distortion.

The differences observed in the behavior of the distortion function for different sequences and rates imply that the estimation error depends on the particular sample and, therefore, in order to find the best estimate it is necessary to take into account not only the actual value of σ_{opt} determined for each sample in the experiments but also the particular shape of the distortion function.

In order to deal with this, our proposal consisted in training an estimator of σ_{opt} with a modified error function that depends on the shape of the distortion curve. Given that the definition of a tailored error function for each sample in the training set is not possible, a weighted least squares (WLS) method (to be introduced in the following section) was designed that employs an importance index for weighting each of the training samples prior to the calculation of the LS estimation.

5.4.2 Linear Regression

As mentioned in previous section, each sequence and encoder operating point yielded a different perceptual distortion function that should be taken into account for obtaining a proper estimator. The formulation of the linear estimate of the filter standard deviation could be as follows:

$$\hat{\sigma} = w_0 + \sum_{i=1}^{N} w_i \cdot x\left(i\right) = w_0 + \mathbf{w}^{\mathbf{T}} \cdot \mathbf{x}, \tag{5.1}$$

where $\mathbf{x}$ is the input characteristic vector and N is the number of characteristics. The weights of the linear estimate are calculated by linear regression from the data in the training set:

$$\mathbf{w} = \arg\min_{\mathbf{w}} \left\{ \sum_{j=1}^{M} f_{e,j}\left(\sigma, w_0 + \mathbf{w}^{\mathbf{T}} \cdot \mathbf{x}(j)\right) \right\}, \tag{5.2}$$

where $\mathbf{x}(j)$ is the input data vector for the jth training dample, representing a particular sequence and bit rate in the training set of M samples. The complete set of training input data $\mathbf{x}$(j) can be rewritten as a matrix $\mathbf{X}$ whose element x_{ij} corresponds to the jth characteristic of the ith input sample (each sample is a row of the matrix $\mathbf{X}$). Finally, the function $f_{e,j}$ is the error function that ideally has a different shape for each data index j.

This error function should match the particular perceptual distortion curve (like those in Figure 5.9) for the jth sample but, in order to obtain a practical solution, the error function is taken as the square error function: $f_{e,j}\left(\sigma, \hat{\sigma}\right) = \left(\sigma - \hat{\sigma}\right)^2$ for every j.

Furthermore, the information concerning the different perceptual distortion functions in the training stage has been addressed by means of an importance parameter, h, that weights each error value and can be introduced in the LS solution, which can be written as a WLS:

$$\mathbf{w}_{\mathbf{WLS}} = \left(\mathbf{X}^T \mathbf{D}_h \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{D}_h^{-1} \mathbf{y}, \tag{5.3}$$

where $\mathbf{D}_h$ is an MxM diagonal matrix whose elements are the importance indexes

h_j extracted from the perceptual distortion curves of each input sample, and $\mathbf{y}$ is the vector of optimal σ values.

It is to be noticed that the optimum σ value for a certain training sample could be different from those employed in the tests (those in the set $\{0, 0.6, 1, 1.25, 1.5\}$). Given that a finer sampling of the output variable is unavoidable due to practical reasons, each optimum output value y_j has been found as the σ value that achieves the minimum distortion using a cubic interpolation of the distortion values achieved for the considered set of σ values 0, 0.6, 1, 1.25, 1.5.

Importance Index

The importance index h should incorporate to the estimation problem the particular sample-dependent shape of the perceptual distortion curves. In general terms, those samples for which the distortion values are low, are less important from the average error point of view than the samples in which the perceptual distortion reaches higher values. Moreover, a soft slope of the distortion curve around its minimum also makes a certain sample less important, in the sense that a slight deviation with respect to the optimum σ comes at a lower cost than if the slope around the minimum were steeper. With both points in mind, an importance measure has been proposed and calculated for each sample. In particular, the importance index for every sample has been defined from its corresponding perceptual distortion curve by averaging the distortion values and measuring the slope around the minimum as follows:

$$h\,(j) = \mu_{d,j} + p_{d,j}, \tag{5.4}$$

where $\mu_{d,j}$ is the mean distortion for the jth sample (the average of the $d_j\,(\sigma)$ function), and $p_{d,j}$ is an slope steepness index described as:

$$p_{d,j} = \begin{cases} 4\,(d_j\,(\sigma_+) - d_j\,(\sigma_{opt})) & , \sigma_{opt} < 0.25 \\ 2\,(d_j\,(\sigma_+) + d_j\,(\sigma_-) - 2d_j\,(\sigma_{opt})) & , 0.25 < \sigma_{opt} < 1.25 \\ 4\,(d_j\,(\sigma_+) - d_j\,(\sigma_{opt})) & , \sigma_{opt} > 1.25 \end{cases}, \tag{5.5}$$

husky 128	husky 256	carphone 384	container 128
5.34	5.47	0.22	4.67

Table 5.6: Values of parameter $h(j)$ for the curves in Figure 5.9.

where σ_{opt} is the position of the minimum in the distortion curve $d_j(\sigma)$ and the slope is calculated from two points σ_+ and σ_- near the minimum ($\sigma_+ = (\sigma_{opt} + 0.25)$ and $\sigma_- = (\sigma_{opt} - 0.25)$).

The values of $h(j)$ for the curves depicted in Figure 5.9 are listed in Table 5.6:

Tentative input Features

The experiments described in Section 5.4.1 highlighted the dependency of the optimum filter strength with respect to certain characteristics of the video sequence and, above everything else, with respect to the encoder operating point. There is a clear relationship between the available bit rate (or QP) and the optimum amount of filtering. An increase in the available bit rate reduces the risk of blocking, ringing and other low bit rate artifacts, making the filtering process unnecessary. On the other hand, the stronger the filter, the better the perceptual quality of the reconstructed sequence when encoding with a lower bit rate (or larger QP).

Moreover, for simple sequences (i.e. those easy to predict) like *akiyo*, motion compensation yields a good encoding performance even for very low bit rate, making the filtering stage unnecessary; while more complex sequences like *football* could benefit from it even at relatively high bit rates. This reasoning highlights the fact that the optimum filter strength is sequence-dependent in the sense that high motion or texture complexity or even the presence of camera motion changes the way the distortions appear and are perceived.

In order to take into account all these dependence, a certain number of basic input characteristics of the problem has been collected, attempting to model the complexity of the sequence and the encoder operating point:

- QP: Quantization parameter. It accounts for the encoder operating point. The target bit rate was also tested as an input characteristic but it is less correlated with the presence of artifacts and the results were worse.

- SI: Spatial Information index. As described in [ITU-T, 2008], it makes use of a *Sobel* edge detector applied to every pixel in the frame. Then, the standard deviation of the modulus of these gradient vectors is calculated for each frame as a measure of its spatial complexity. The spatial complexity of the sequence is obtained as the maximum of this frame measure along the whole sequence.

- TI: Temporal Information index. Defined also in [ITU-T, 2008], it is a measure of the average absolute differences between successive frames in a sequence, evaluating the degree of motion. Again, the temporal complexity of a sequence is obtained as the maximum of this motion measure along the entire sequence.

- X_{cm}: Camera motion. It is a measure of the average camera motion vector, extracted from a complex hierarchical motion estimation analysis.

Additionally, it is known from the preliminary tests that the relationship between the risk of low bit rate artifacts and the QP could be defined as non-linear (only at very high values of QP the artifacts become evident). In order to introduce this non-linearity in a linear estimator, QP^2 is considered as an additional input characteristic. Following the same reasoning, the product of SI and TI is also included, as a measure of the spatiotemporal complexity.

K-fold Validation

In order to decide which of these inputs brings more information on the desired output, a cross-validation procedure has been designed to try different combinations of these characteristics and obtain the combination achieving the best results for the training set. Given the small size of the training set, and the huge amount of time for obtaining new samples due to the quality assessment process, only 72 samples were available for training and validating the estimation, leaving out 14 samples for validation purposes. On one hand, this means that all the possible configurations of input characteristics can be tested with little CPU time. On the other hand, with such a small validation set, the results are too dependent of the particular set chosen and methods like the *K-fold* cross-validation become a must.

The K-fold cross-validation takes the original data set, divides it in K subsets and repeats as many times the training phase of the estimator. For each of these *folds* a different subset is employed as the validation set while the remaining K-1 subsets are used for training. Averaging the estimation error over the validation sets for the K iterations, a unique error measure is obtained for the whole original data set for comparison purposes. In our case, given the particular nature and structure of the data, 5 subsets were selected by putting together all the different bit rate versions of the same original sequence: 4 of the subsets consisted of 4 sequences with their 4 rate versions (16 sample points each), and the remaining subset consisted only of 2 sequences (8 sample points). With this distribution, the presence of different versions (at different bit rates) of the same sequence in the train and test sets of each one of the K iterations is avoided.

Feature Selection

The estimation error, understood as a function of the difference between the desired output and the estimate, is not a good performance metric in this case. The objective of the estimator is to find the proper amount of filtering for a sequence in order to maximize the perceived quality. Therefore, the performance metric for the estimator should be the perceptual quality observed in the reconstructed sequences after being filtered with the estimated σ parameter and encoded. The final tests with the best of the estimates $\hat{\sigma}$, in which some tests sequences will be encoded with and without residue filtering and compared, will be described in Section 5.5. In this section and concerning the initial training set, a different performance metric will be employed.

Nevertheless, the feature selection is based on an error measure of the σ estimation calculated by means of the cubic interpolation distortion curves, considering the average error of the K iterations in the K-fold validation. The final result after trying all the possible combinations of inputs for the WLS algorithm was obtained by the following estimator:

$$\hat{\sigma}_{WLS} = \mathbf{w}^{\mathbf{T}} \cdot x = w_0 + w_1 QP + w_2 SI, \tag{5.6}$$

where the vector of optimum weighting factors is $w^T = [-0.1057, 1.7302, -0.5987]$.

As can be seen in the final solution, the proper σ value is mainly related to the operation point of the encoder through the quantizer, and a certain contribution of the spatial complexity is also observed. This does not mean that other parameters are irrelevant to the decision of σ, but their influence may be somehow redundant when considering the final particular feature selection. For example, it is clear that temporal complexity has an impact in the appearance of blocking effects but, given that it also implies the use of higher QP values for a given bit rate, Equation 5.6 could still provide a reasonable result.

Obviously, these results cannot be considered as optimum given that the perceived quality measure has been coarsely approximated for practical purposes.

5.4.3 Filter Control Integration

Once the particular solution for the proper amount of filtering has been derived, the result in Equation 5.6 can be integrated in the H.264/AVC encoder for testing. However, some considerations need to be done regarding this integration. In the experiments described within this Section, the particular values of spatial complexity were calculated for the whole sequence as the maximum of the SI values obtained on a frame basis. In a real-time video encoder scenario, future values are not available and a practical method is proposed instead. In particular, the on-line implementation calculates this maximum over a window of 10 past frames (future frames are not available) simplifying the measure and maintaining a good overall estimate.

In this way, the values of SI and QP are obtained from the pre-analysis stage and the rate control decision, respectively, and the filter strength is calculated by means of Equation 5.6. The obtained σ value is restricted to the range $[0, 1.5]$.

Old sequences	husky, news, paris, soccer
New sequences	city, flower, hall, ice, irene, mother&daughter, silent, students

Table 5.7: List of test CIF sequences into two groups: sequences already employed in the training stage and new sequences.

5.5 Experiments and Results

In order to measure the performance of the complete system, an extensive test including subjective quality assessment was carried out. The objective is to demonstrate that the residue filtering technique achieves a good quality performance when compared to the non-filtered version for the same sequence and target bit rate. The benefits of filtering should be more evident at lower bit rates and complex sequences, but at higher rates or simpler sequences the quality of the proposed scheme should not be worse than the non-filtered version due to a proper estimation of the filter strength.

In the following sections, the experiment setup will be described for the quality assessment of both the proposed algorithm and the reference version. After that, a complete summary of the results obtained in the subjective test will be given and its validity discussed before extracting some conclusions about the performance of the complete system in Section 5.6.

5.5.1 Experimental setup

Encoder configuration

A total of 12 CIF sequences, listed in Table 5.7 have been employed for the subjective tests: 4 sequences extracted from those employed in the filter control training stage and 8 new sequences. All the sequences have been encoded with an H.264/AVC encoder, modified according to the filtering scheme described in Section 5.3 and the filter control in Section 5.4, working at target bit rates of 128, 256, 384 and 512 kbps.

As described in Section 4.1.2, the differences in the kind of processing carried out in inter and intra frames can be noticeable between a P or B frame and the successive I frame of a new GOP. In this way, a periodic pattern of 25 or 30 frames per GOP motivates the appearance of the so-called *flicker* artifact, which can become very annoying when higher QPs are employed. In order to prevent this effect to bias the subjective opinion of the viewers, the test bench has been generated in a way that only the first picture is an Intra frame. The rest of the encoding parameters are summarized here:

- IP...P pattern with 100 frames at 25 fps
- Rate-distortion Optimization enabled
- Rate Control ON: CBR at 128, 256, 384, and 512 kbps
- 8x8 DCT
- SKIP, Inter16x16, Intra16x16, and Intra8x8 modes
- One reference frame for motion estimation
- CIF (352x288) test sequences with different spatial and temporal contents

Additionally, the same sequences and rates are employed for obtaining the corresponding reference non-filtered reconstructed sequences. The complete test bench is then composed by a total of 48 pairs of filtered and unfiltered sequences (12 sequences and 4 bit rates)[3].

Pair comparison setup

The subjective experiment design is a variation of the Pair Comparison (PC) method described in [ITU-T, 2008] and summarized in Section 2.4.1. There was no reference to be shown. Instead of this, both reconstructed sequences were shown simultaneously without any clue of their nature. The subject was asked to decide which one

[3] The complete set of reconstructed sequences (both filtered and non-filtered versions) corresponding to these tests can be found in `http://www.tsc.uc3m.es/~mfrutos/Thesis/TestSequences/` at different bit rates (128, 256, 384 and 512 kbps).

of the sequences was perceived as *with higher quality* and then a categorical scale was employed for determining the perceived impairment between them (or distortion measure):

0	-	Imperceptible
1	-	Perceptible but not annoying
2	-	Slightly annoying
3	-	Annoying
4	-	Very annoying

In order to comply with some of the setup conditions described in [ITU-T, 2008] for subjective quality assessment, these sequences were displayed in 17" monitors with low background room illumination and a proper viewing distance. Each pair of test sequences must be shown twice in order to display them in both possible orders: each one was shown once in the left and once in the right side of the screen. For this reason, there was no need of repeating a subset of the experiments for testing the consistency in the answers for a PC method. Furthermore, a subset of the test pairs was employed as a calibrating set at the beginning of the experiment as required in the recommendation. The results of these test pairs were not taken into account.

A number of assessors from 4 to 40 is suggested in [ITU-T, 2008] for obtaining statistically relevant data. In our experiments, a number of 8 assessors contributed with their opinions. In order to determine how the previous experience of a subject influences its answers to the subjective test, 4 assessors were naive and the other 4 assessors were experts in the fields of image or video coding, having taken part in previous subjective tests.

Result handling

The assessors were provided with a test sheet in which the instructions were summarized and the results could be checked. The results were processed as follows. First of all, the results for the calibrating test sequences were discarded. Then, a consistency test was carried out: if an assessor marked sequence A (the one at the left side) as the best of a pair of test sequences, as soon as this pair is displayed again in reverse order, the observer should see a better quality in sequence B (the one at the right side). The following considerations were applied:

- Consistency: if the answers were identical, the response was consistent and processed without change.
- Slight discrepancy: if the answers were different but in one of them the impairment was considered as imperceptible, the response in which the impairment was perceived was taken as the valid answer and the distortion score was averaged.
- Severe discrepancy: if a noticeable impairment was detected in both repetitions of a basic experiment but the winner sequence was different, the response was discarded.
- Discarding: If more than 20% of the responses by an assessor were discarded due to severe discrepancy, the assessor would be rejected.

The remaining valid data were analyzed and a differential quality score (DMOS) was computed by averaging the distortion values assigned by the assessors (from 0 to 4). For those cases in which the filtered version was considered to be the winner, the contribution to the DMOS was taken as positive, while for those in which the non-filtered version was considered to be the best one, the contribution to the DMOS was taken as negative.

5.5.2 Results

Objective Tests

Before analyzing the data corresponding to the subjective experiment, it is possible to make also an interesting analysis of the results obtained from the encoding of the sequences. Table 5.8 shows, for each target bit rate, the average values of the standard deviation employed for filtering (σ) and the PSNR and QP differences over the reference encoded version. A negative value in $\overline{\Delta PSNR}$ means that the filtered

Rate	σ_{ave}	$\overline{\Delta PSNR}$	$\overline{\Delta QP}$
128	0.65	-0.25	-2.0
256	0.47	-0.23	-1.8
384	0.37	-0.16	-1.5
512	0.30	-0.18	-1.2

Table 5.8: Average results for all the test sequences at different bit rates.

version achieved a lower pixel-wise quality than the reference non-filtered version. Likewise, a negative value in $\overline{\Delta QP}$ means that a lower QP was employed by the RC for obtaining the same target bit rate in the filtered version.

As can be seen in Table 5.8, higher target bit rates mean that, in average, a lower amount of filtering was employed in average. The average PSNR difference shows a reduction of the pixel-wise quality for all the bit rates but, given that the PSNR does not correlate well with the perceived quality in many situations, it was to be expected. The objective was to reduce the low bit rate artifacts at the expense of a slight decrement in the average PSNR due to the smoothing. On the other hand, an average QP reduction is achieved by the proposed algorithm for all the target bit rates that in some of the sequences can even reach a reduction of 4 units. This was indeed an important result given that, as mentioned in Section 4.1.2, the use of large quantizers lead to the appearance of most of the low bit rate encoding artifacts. In this sense, the mechanism of residue filtering helped to reduce QP from 1 to 2 units in average.

Subjective Tests

A detailed summary of the responses to the test is shown in Table 5.9 for the target bit rates. First, the number of valid results is stated, which corresponds to those answers with slight or no discrepancy. It is to be noticed that none of the assessors was rejected due to severe discrepancy in the results, and the higher severe discrepancy rate was less than 17%. Then, the valid data were collected and divided into 9 different categories depending on the impairment score and the sign of this impairment with

Rate(kbps)	Valid	#hits per DMOS category								
		-4	-3	-2	-1	0	+1	+2	+3	+4
128	62	0	2	1	5	14	10	16	13	1
256	66	0	2	2	7	22	17	8	6	2
384	65	0	0	0	5	40	9	9	0	2
512	67	0	0	0	9	41	10	2	3	2

Table 5.9: Itemized number of valid DMOS responses per target bit rate. Positive DMOS means that the filtered version was considered better than the non-filtered one by the assessors.

Rate(kbps)	$\overline{DMOS}$	σ	CI	%FSB	%FCB	%NSB	%NCB
128	1.18	0.45	±0.4	42	23	10	3
256	0.6	0.56	±0.36	38	12	14	3
384	0.48	0.24	±0.26	28	3	8	0
512	0.35	0.32	±0.27	18	7	13	0

Table 5.10: Average DMOS and other statistical data from subjective tests. Positive DMOS means that the filtered version was considered better than the non-filtered one by the assessors.

respect to the hypothesis of a better quality of the filtered version. As can be seen in Table 5.9, the frequency of positive values is higher than that of negative scores, specially at low bit rates, confirming that the filtered version was perceives to be higher quality by the average assessor.

The final average DMOS values, the main result of the experiment comparing the quality of the filtered sequences against the non-filtered reference versions, are summarized in Table 5.10 and depicted in Figure 5.10 for the different target bit rates. This values of DMOS were calculated by averaging the impairment scores obtained in the subjective test and listed in Table 5.9 for all the sequences and assessors at each bit rate. As can be seen, in the assessors' opinion, there is an overall benefit in employing the proposed scheme for residue filtering at low bit rates. This benefit decreases as the bit rate increases, as a consequence of lowering the filter strength, but it still maintains a good performance up to 512 kbps. Additionally, the standard

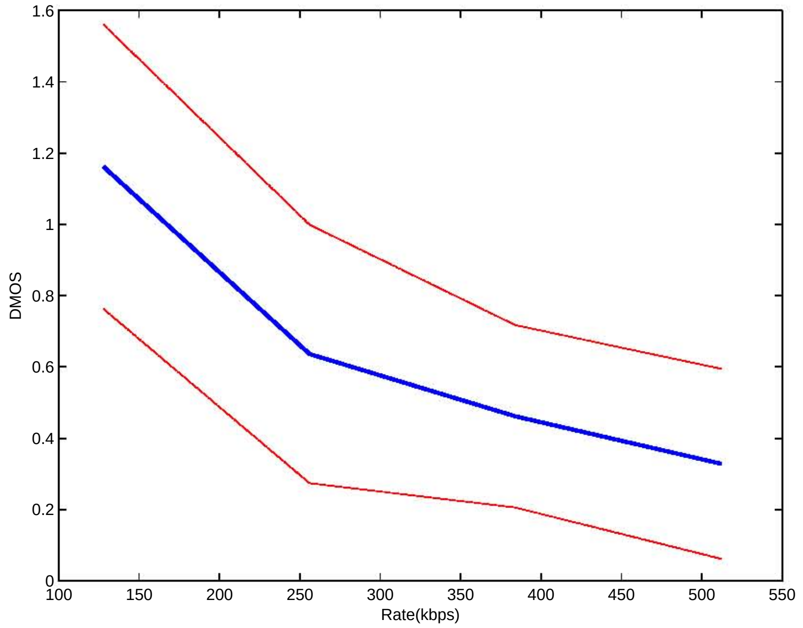


Figure 5.10: DMOS values at different bit rates with their corresponding confidence intervals. Positive DMOS means that the filtered sequence was considered better than the non-filtered version by the assessors. Dotted lines show the 95% confidence intervals.

deviation and 95% confidence intervals of the estimated DMOS are also listed in Table 5.10 in order to assure that the results are statistically representative. The rest of the values in the table are described here:

- Filtered version Slightly Better ratio (FSB): is the percentage of cases in which the filtered version is considered as the better one and the impairment score for the non-filtered version corresponds to *a perceptible but not annoying distortion* or *a slightly annoying distortion* (DMOS values between +1 and +2).

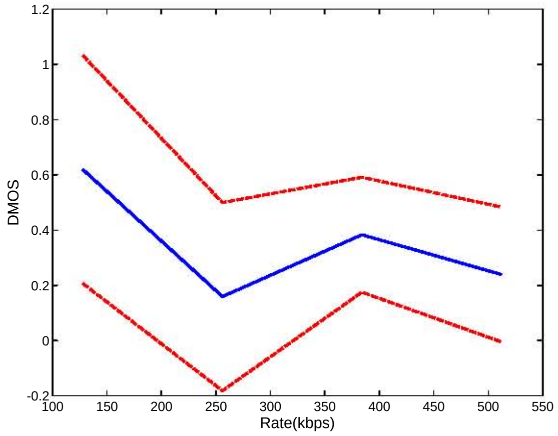
- Filtered version Clearly Better ratio (FCB): in this case, is the percentage of tests in which the filtered sequence is the preferred version but the impairment observed is *annoying* or *very annoying* (DMOS values between +3 and +4).

- Non-filtered version Slightly Better ratio (NSB): percentage of tests in which the non-filtered version is preferred with low values of the impairment (DMOS between -1 and -2).

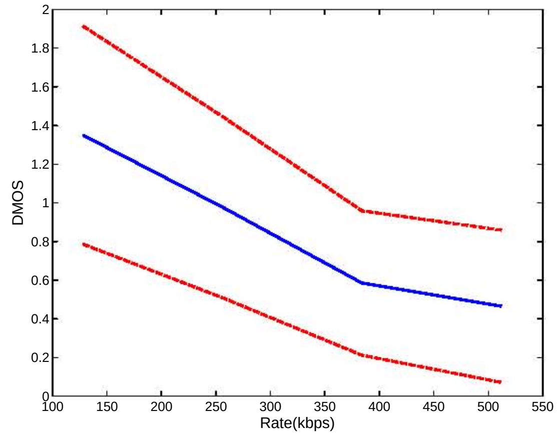
- Non-filtered version Clearly Better ratio (NCB): percentage of tests in which the non-filtered version is preferred with high values of the impairment (DMOS equals -3 or -4).

From the data summarized in Table 5.10 it is observed that NSB and NCB are below 15% and 3% respectively for all the tested bit rates, corresponding with a low degree of rejection of the proposed technique. On the other hand and according to the results, there is a high percentage of acceptance of the technique: the addition of FSB and FCB gives a 65% of positive answers at 128 kbps, while it slowly decreases down to 25% at 512 kbps.

Additionally, the subjective quality results obtained from the experiment have been analyzed by separating those corresponding to naive subjects from those corresponding to experts. Figure 5.11 shows that, although the positive results can be observed in both subsets, naive subjects tend to grade the impairment between filtered and non-filtered version with lower values, while experts tend to assign larger scores to the impairment. The most likely cause of this behavior is a better training of



(a)



(b)

Figure 5.11: DMOS values at different bit rates with their corresponding confidence intervals for (a)naive subjects, and (b)expert subjects. Positive DMOS means that the filtered sequence was considered better than the non-filtered version by the assessors. Dotted lines show the 95% confidence intervals.

the expert assessors for detecting impairments in encoded video, but some differences in the tolerance to artifacts could also influence this results.

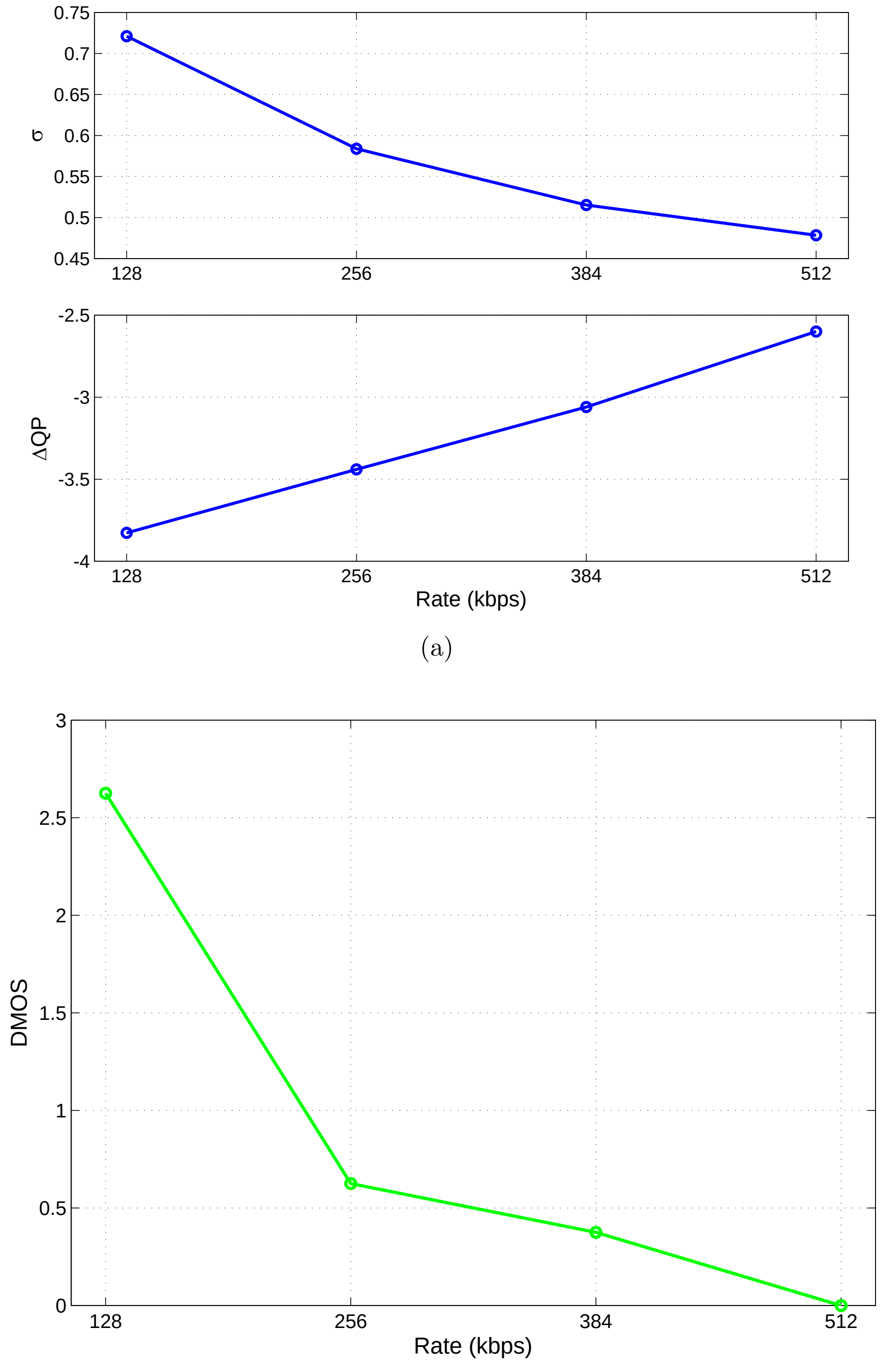
Finally, for the sake of brevity, only the results of the subjective test for a representative sequence have been included in this Section and those corresponding to the rest of the sequences are included in Appendix B. In Figure 5.12(b), the subjective scores for the sequence *city* at different bit rates are depicted as a good sample of the average behavior of the system. As can be seen, there is a noticeable improvement in subjective quality at low bit rates that slightly decreases as the bit rate rises. As mentioned before, the use of lower quantization steps at higher bit rates reduces the risk of artifacts, and therefore, the need for filtering. In this sense, it can be seen in Figure 5.12(a) that the values of σ selected at higher bit rates are lower than those employed at low bit rates, obtaining only moderate QP reductions.

Complexity considerations

With respect to the number of operations necessary to perform the residue filtering algorithm, the following list summarizes the extra calculations in each part of the system:

- Complexity estimate: the complexity parameter SI must be calculated on a frame basis. To this end, as described in Section 5.4.2, a Sobel gradient operator is applied to each pixel in the frame. The calculation of the squared modulus of the gradient takes 25 operations (products and additions) per pixel, a number very far from the number of operations required to test a certain position for motion estimation or a certain direction for intra prediction, for mentioning only two examples. Moreover, the results of these calculations could be of some help in other tasks within the encoding process. For example, the spatial complexity measure SI could be employed by the rate control for bit allocation in Intra frames, as in [Jing et al., 2008]. Another option is to reuse these gradients for predicting the edges directions, reducing the amount of intra modes to be tested in a block, as described in [de Frutos-López et al., 2010].

- Filter complexity calculation: The calculation of the σ parameter is carried out



(a)

(b)

Figure 5.12: Results for the sequence *city* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.

only once per frame and, therefore, its complexity cost is negligible. Given that the same filter is employed for all the blocks in the picture, the corresponding filter weights care updated by means of Equation 4.4, only once per frame.

- Pixel filtering: The filtering process described in Equation 4.3 is carried out for each pixel employing a 5x5 square mask as spatial support Ω. This means that 50 operations (25 additions and 25 products) are employed for the calculation of each filtered pixel. Given that the filters are applied to the residue of each candidate mode, and for a minimum block size of 8x8 pixels, each pixel is filtered a total of 14 times (1 inter mode, 4 intra 16x16 modes, and 9 intra 8x8 modes). The result, about 700 operations per pixel, means a notable increase in CPU consumption, much lower than the cost of motion estimation but higher than the complexity of intra prediction. Nevertheless, for low complexity encoding devices, the algorithm could be simplified in a way that only the best intra mode (direction) for each block size was filtered, reducing from 17 to 3 the number of times each pixel is filtered (and the number of operations from 700 to 150).

5.6 Conclusions

In this chapter, a residue filtering scheme has been proposed that aims at simplifying the operation of a video encoder by smoothing high-frequency components of the prediction residue prior to the DCT transform. The method includes a perceptually-tuned filter control capable of modulating the filter strength depending on the sequence to be encoded and the encoder operating point. The proposal has been integrated in an H.264/AVC encoder and its performance has been subjectively tested by means of a normative-compliant procedure for quality assessment. The results show a significant improvement in the perceived quality over the reference video encoder for a wide range of sequences and for low bit rate conditions.

The main contribution of this proposal is the incorporation of a filter control based on subjective quality. The objective was to find the proper standard deviation of a 5x5

Gaussian filter employed for smoothing the prediction residue in an H.264 encoder. To do so, a method for estimating the subjective distortion of a certain sequence at a certain bit rate was first defined as a function the amount of residue filtering. This subjective distortion was chosen as the criterion to find a compromise between the low bit rate encoding artifacts and the blurring introduced by the smoothing filter.

With the data extracted from a number of sequences, a linear regression (WLS) method was trained to estimate the value of σ_{opt} for residue filtering. The final estimator obtains a filter strength that is dependent of 1)the operation point of the encoder, towards the QP value; and 2)the particular sequence to be processed, by means of a measure of the spatial complexity. In order to validate the expression obtained for σ_{opt}, a subjective test was conducted according to the recommendation in [ITU-T, 2008]. The results show that the filtered sequences obtain higher quality scores than the corresponding non-filtered reference versions.

The complexity of the method is high if the filtering process is applied to every prediction mode in H.264, especially if the residues of every directional intra modes are filtered prior to the mode decision. This configuration corresponds to the results shown in Section 5.5, and entails an increase of 50% with respect to the encoding time of the reference software version JM18.0. However, it is to be noticed that no optimization has been performed for the implementation of the residue filtering algorithm. Furthermore, there is room for simplificating the algorithm without important changes in its performance.

Although it has been implemented in an H.264 encoder, the system can be integrated in any hybrid encoder. The only restriction could be found in the intra frames. While the H.264/AVC standard includes an intra prediction procedure whose prediction residue has similar statistical properties than that of motion compensation, not every hybrid encoder includes an intra prediction. In case the intra frames were not predicted, filtering could be inadequate. However, if the proposed algorithm is implemented in a different codec, the particular form of $\widehat{\sigma}$, the estimator of the residue filter strength, must be re-evaluated, selecting the proper combination of input parameters for the best subjective score.

The experiments have been carried out with CIF sequences but the algorithm could be employed no matter the resolution of the video sequences. The resolution could be employed as an additional input variable to be used for the estimation of the proper filter strength.

5.7 Further work

Like other works described in the literature, the filtering scheme proposed in this PhD Thesis highlights the advantages of the simplification of video sequences by pre-filtering. However, there is still room for obtaining a more general solution capable of achieving a good performance in every situation. These are some of the future lines brought into light by this PhD Thesis:

- Many algorithms based in pre-filtering for simplification include a *filtering allocation* procedure that makes use of perceptual considerations for a better distribution of the resources. In case of filtering the prediction residue this aspect becomes less relevant, given that only those regions in the residue frame with significant values will contribute to decrement in the bit rate. However, it could be interesting, even for our approach, to include some considerations about ROI in order to not to filter the residue of crucial regions in the scene such as faces, labels or other important details. Instead of using a unique value of σ, a map of σ values related to the ROI and/or other aspects of the HVS described in Section 4.1.1 could be employed.

- Although the linear model relating the proper amount of filtering to both the operation point of the encoder (the QP) and the complexity of the sequence (through the SI parameter) performs quite well for the set of sequences under study, a better approach could be found by using non-linear models. This line of work was partially followed and some experiments were carried out with Gaussian Processes for non-linear regression. However, these experiments were not included in the final version of this PhD. Thesis because we did not find a practical way of introducing a point-wise error function like the importance index described in Section 5.4.2. Different configurations of the set of input characteristics were tested without achieving a better performance than the linear model employed in the final solution, in terms of the error in the estimation of the σ_{opt} determined in the training phase. Instead of giving up with this solution, it could be a good idea to explore other non-linear regression algorithms capable of being modified according to a measure similar to the importance index for error calculation.

- On the other hand, some tests could be useful to determine whether the algorithm performs well regardless of the encoding pattern and GOP size. As mentioned before, a different filter strength in successive frames should not mean the appearance of any artifact, but this does not mean that our proposal perform well when different frame types are employed. Some adjustments could be necessary, for example, when using hierarchical patterns in which the residue of the B frames in the deepest layers is highly reduced with respect to that of upper layers.

- The training of the linear estimator could benefit from employing an exhaustive quality assessment procedure based in subjective tests with different subjects. In this way, a possible bias in the perceptual distortion functions could be compensated by averaging different quality scores for the same training sequences. In the same way, if enough resources were available (in terms of time and personal), the cross-validation process carried out by means of the K-fold method for the feature selection step could also be improved by employing real assessors for determining the subjective distortion.

- Concerning the distribution of modes selected when the residue filter is employed, a certain increment in SKIP blocks was observed in detriment of other modes representing operating points with higher rate and quality. This means that there could exist a discrepancy between the operating point selected by the rate control by means of a certain QP and the corresponding λ value for RD optimization. A slightly better solution could be achieved by re-calculating the relationship between λ and QP for the H.264 encoder including the proposed residue filtering mechanism. The process to follow would be similar to that discussed in [Sullivan and Wiegand, 1998].

Chapter 6

Discussion

Wisdom begins at the end. (Daniel Webster)

In this final chapter, the conclusions of both parts of this PhD Thesis are summarized: in the first place, the contributions of the author to both reconfigurable and low bit rate video coding will be highlighted; next, those conclusions derived in previous chapters will be summed up; and finally, some interesting research lines, opened during the development of this PhD Thesis, will be discussed.

6.1 Contributions of this PhD Thesis

6.1.1 Contributions to Reconfigurable Video Coding

These are the contributions of this PhD Thesis in the field of reconfigurable video coding:

- Collaboration in the design and implementation of a reconfigurable codec prototype, a hybrid video codec, capable of changing the transform during the encoding session and sending the corresponding new inverse transform to the decoder. Above all, this prototype demonstrated the potential benefit of the on-the-fly reconfiguration of the video encoder for adapting to the video content and the application scenario. The main contribution of the author to this prototype was the design of a syntax for transmitting the inverse transforms to the decoder. This syntax, described in [Bystrom et al., 2010] as a part of the prototype codec, was based on a reduced set of instructions that allow for converting the flow graph representations of the inverse transforms into a compact binary representation. Apart from its use into the reconfigurable codec prototype, this syntax for describing graph representations might has some value by itself as a compact description of certain transforms in other scenarios.

- After the proof of concept, a complete syntax for defining a video decoder configuration was developed. The so-called Decoder Description Syntax (DDS) included a set of basic functions for defining the decoder functionality as well as the protocol for sending the decoder description in the video bitstream. The author contributed to the design and optimization of the DDS in order to reduce the overhead due to the reconfiguration information, i.e., Decoder Description Object (DDO), as described in [Kannangara et al., 2010]. The resulting DDO size to transmit an H.264-like decoder was reduced to 1351 bytes, which could be an affordable overhead for the complete description of a decoder.

- Once we showed that the on-the-fly codec reconfiguration obtained some benefit from the Rate-Distortion point of view, a number of challenges for the implementation of the FCVC paradigm were identified during the development of this prototype, as well as the differences between the FCVC and the RVC standard. The author of this PhD Thesis contributed to the discussion of these issues and, finally, our effort was directed to the convergence of some of the FCVC ideas with the RVC standard, still under development. Specifically, a proposal was made by some members of the FCVC group at the MPEG meeting in July 2009 (see [Richardson et al., 2009c]) for translating the RVC functional units into FCVC-DDS in order to incorporate fine-grain reconfiguration to RVC.

6.1.2 Residue filtering for video simplification

Concerning the second part of the Thesis, the objective was to simplify video sequences so that lower QP values were required to operate at low bit rates, reducing the risk of appearance of harmful encoding artifacts. In this sense, although some previous works were aimed in the same direction, a more practical solution was proposed based in two main ideas: filtering the prediction residue and controlling the filter strength based on perceptual considerations. The main contributions of this second part of the Thesis are:

- An algorithm for filtering the prediction residue in an H.264/AVC encoder. Although the residue filtering was suggested before in the literature as a way of simplifying video, the integration of such an algorithm into an H.264/AVC has been performed in this Thesis for the first time.

- A filter selection process for simplifying the prediction residue. A number of candidate filters and configurations were tested for selecting the best algorithm in terms of 1) low complexity, 2) easy control, and 3) absence of noticeable artifacts in the reconstructed sequence. These experiments might be useful for future works related to prediction residue filtering.

- A Filter Control integrated with the Rate Control algorithm. This sub-system, the main contribution of this part of the Thesis, aims at estimating the proper amount of filtering for each sequence and encoder operating point. The estimate was learned from the subjective distortion observed for different filter strengths, train video sequences, and target bit rates. The final expression of the optimum filter strength was implemented in a modified version of the H.264 standard video encoder in order to test the performance. A subjective quality assessment experiment was conducted following the ITU-T P-910 recommendation and the results showed a noticeable improvement in the quality of the reconstructed sequences at low bit rates when using the proposed technique. The perceptual-based design of the filter control could be easily extended to any hybrid video encoder different from H.264/AVC.

6.2 Conclusions

During the last years, the author of this PhD Thesis has contributed to the field of video coding with several research papers dealing with different blocks of the hybrid DCT/DPCM video encoder. From the first works related to mode decision and fast motion estimation for H.264, his interests moved to the field of rate control. The works related to RC helped to look at the video coding process from a wider perspective, which somehow inspired his interest in reconfigurable video coding. As a result, the first part of this Thesis was devoted to question the big picture of video coding standards in the search of a more flexible way of defining video codecs: the fully configurable video coding.

In this PhD Thesis, the proposal of a fully configurable video coding was described, and compared to the RVC standard from MPEG, discussing the pros and cons of both approaches. Two main differences between RVC and FCVC have been identified as novel contributions of the FCVC group: 1) the on-the-fly reconfiguration; and 2) the fine-grain reconfiguration.

In our opinion, the potential benefit of the on-the-fly codec reconfiguration was proced by the results obtained by two different FCVC prototypes. The adaptive-transform codecs proposed obtained a RD benefit by only changing one transform for another and communicating the corresponding changes to the decoder, i.e, the R-D performance obtained after considering the overhead generated by the transmission of the new decoder configuration, was better than that of a non-reconfigurable solution. Possibly, as a result of these efforts, the RVC included some kind of on-the-fly reconfiguration in later versions of the standard that was not originally conceived.

Concerning the fine-grain reconfiguration, it is more difficult to compare its performance with that of the solution proposed by the RVC standard, which consists in the combination of standard building blocks (or functional units) for conforming a new decoder. Intuitively, even if these FUs were small enough, a method capable of reconfiguring each sentence in the codec software would exhibit a considerably higher flexibility and, therefore, a potential gain in terms of rate and distortion with respect with the building-block-based solution. Additionally, the use of new selfdefined video processing tools is inherent to FCVC, and no standardization process is needed. Nevertheless, as highlighted by our experiments, the use of the FCVC approach to fine-grain reconfiguration entails some issues: the decisions to be made at the encoder become much more complex; there is a risk of malicious behavior or malfunction related to an inappropriate reconfiguration of the decoder; new tools implementations can not be platform-optimized at the decoder side.

Besides questioning the potential drawbacks of video coding standards, a different research theme was explored related to rate control techniques. In this PhD Thesis, some methods for enhancing the performance of state-of-the-art video codecs at low bit rates were also explored. Specifically, the second part of the Thesis proposed solutions for video simplification in order to reduce the output bit rate while avoiding typical low bit rate encoding artifacts. The target application would be the coding of real-time video in wireless environments, in which both the rate and the complexity available are restricted.

The proposed method simplifies the prediction residue in a hybrid video encoder (the one chosen for the experiments was an H.264 encoder), by means of a Gaussian filter with an adaptive filter strength. By filtering the residue instead of the original pixels of the sequence, a more precise control of the output bit rate can be achieved, given that the filter is integrated with the motion compensation process. Moreover, by adaptively selecting the filter strength according to the characteristics of the sequence and the operating point of the encoder, a good trade-off between the blurring effect, inherent to smoothing filters, and low bit rate encoding artifacts, such as blocking or ringing, can be obtained.

The first challenges addressed during the implementation of the residue filtering algorithm were its integration in the rate-distortion-optimized mode decision of the H.264 video encoder, and the selection of the filter structure and configuration. After numerous experiments, and having in mind the complexity and bit rate constraints of the target scenario, a simple Gaussian smoothing filter was selected as the best option for residue filtering. One of the criteria for selecting this filter was that it could be controlled through a unique parameter.

The filter control was then implemented by looking for a relationship between the amount of filtering and the appearance of harmful artifacts in some training subjective tests. When working at low bit rates (for a particular sequence), the non-filtered version usually exhibited some blocking or ringing artifacts that could be reduced by using a Gaussian filtering of the residue. As the value of the variance of this filter was increased, some blurring artifacts might appear in certain moving regions, while the QP value was decreased and the artifacts related to a higher quantization step were reduced. The best filter strength, σ_{opt}, was then subjectively selected among the different filtered versions (from $\sigma = 0$ meaning no filter, to $\sigma = 1.5$), as the best trade-off between blurring and large QP artifacts. In this way, a distortion (or error) measure was obtained for each training sequence, bit rate, and σ, that enabled to train a simple linear model to estimate σ_{opt} from certain characteristics of the sequence and the operating point of the encoder.

The proposed adaptive residue filtering obtained an important QP reduction when integrated in an H.264 encoder, while controlling the appearance of blurring, which is typical from pre-filter schemes. Specifically, it was tested by means of a normative subjective quality assessment experiment (following ITU P-910 recommendation). The average score of the impairment (DMOS) obtained between the non-filtered reference sequences and the residue-filtered sequences rises up to 1, in a range from 0 to 5, for low bit rates, meaning that the average non-filtered version is considered as having a noticeable distortion (although not too harmful) with respect to the filtered version.

The proposed solution can be implemented even in low-complexity scenarios given that the algorithms involved are quite simple. Moreover, with a proper adaptation of the training stage and the filter implementation, this algorithm can be implemented in any hybrid video encoder.

Summarizing, this PhD Thesis has made contributions to the solution of two relevant issues arisen from the development of modern multimedia applications. The heterogeneous scenarios, such as wireless channels, demand more and more adaptivity to the content and to the devices involved in the transmission process. Such an adaptivity is harder and harder to obtain by means of monolithic standard codecs structures. At the same time, traditional concerns in the field of video coding such as the lack of bit rate or computational resources are still an issue, given that multimedia applications come hand to hand with the development of the new handheld and resource-constrained devices.

The solutions proposed are the fruits of an in-depth study of modern video coding and processing techniques, and they try to shed some light on the two addressed problems.

6.3 Further work

Some of the issues of modern video coding described in this PhD Thesis are too complex for a simple and unique solution. Moreover, the proposals included in this document are far from being closed, and some research lines, or even new fields, keep open. The following open lines are considered as the most interesting from those highlighted during the development of this PhD Thesis.

First of all, we consider that the development of reconfigurable solutions for the definition of video codecs is still a matter of interest, even with the advances in the RVC standard. The ability of a video coding platform for reconfiguring its structure in a fine-grain basis, without the concurrence of standard building blocks, could still be an important field for future research. The issues of FCVC, evidenced by our experiments, could be overcome, and the platform for fine-grain reconfigurable video coding could be re-designed. This platform would be useful for research purposes, or could be employed for defining and testing new tools in order to include them into the RVC-toolbox, as part of the RVC standard. Could the RVC standard include in the future a fully configurable functional unit with a special syntax for research purposes?

Concerning the proposal of a prediction residue filter for video simplification at low bit rate, our experiments have demonstrated that some important improvements can be achieved over pre-filter solutions by using a subjective assessment of the influence of low bit rate artifacts, rather than estimating this subjective scores from objective measures that are far from reliable. In the same way that subjective quality assessment experiments are conducted for the selection of the set of video coding tools conforming a new video coding standard when a call for proposals is made, a similar set of experiments could be useful for designing, under different circumstances and with a variety of input sequences, an algorithm devoted to simplify video sequences (or prediction residue) in order to reduce the low bit rate artifacts while introducing no negligible blurring, such as the algorithm proposed in this document.

There is still room for improvement of the proposed solution. The filter control could be enhanced by introducing a proper non-linear estimator, with some other input characteristics, or by modifying the perceptual distortion measure which has been employed as error function for obtaining the linear estimator. Additionally, instead of filtering the entire frame with the same filter strength, some benefit (from the subjective quality point of view) could be obtained from unequal filtering of the frames based, for example, on ROI considerations.

During the course of the next years we will see, whether or not techniques such as those described in this PhD Thesis will play an important role in the oncoming generation of video applications.

Part III

Appendices

Appendix A

Documents for Subjective Test

Subjective Test. Instructions

- Open test video sequences by pairs beginning with File001_A.cif and File001_B.cif, and finishing with File0104_A.cif and File0104_B.cif.

- Put both display windows at middle height. Take care of putting always the A test sequence in the left side and the B version in the right side of the window.

- Keep at a normal viewing distance of the display in a low illuminated environment (turn off the lights if possible) and avoid light reflections in the display.

- Run one sequence after the other and decide which one is preferred. A second run of both sequences is allowed to decide for difficult cases.

- Put a mark in the results sheet in the box corresponding to your preferred version. If none of the sequences seems better than the other, just mark one of them at random (the result will be ignored due to the following item).

- Finally, try to rate the perceived impairment between both sequences and mark it in the results sheet according to the following table:

0	Imperceptible
1	Perceptible but not annoying
2	Slightly annoying
3	Annoying
4	Very annoying

- In order to avoid tiredness, it is recommended to take breaks of about 5 minutes after 20 minutes of test but it is also recommended to run the complete test bench in only one session.

Thank you for your cooperation!

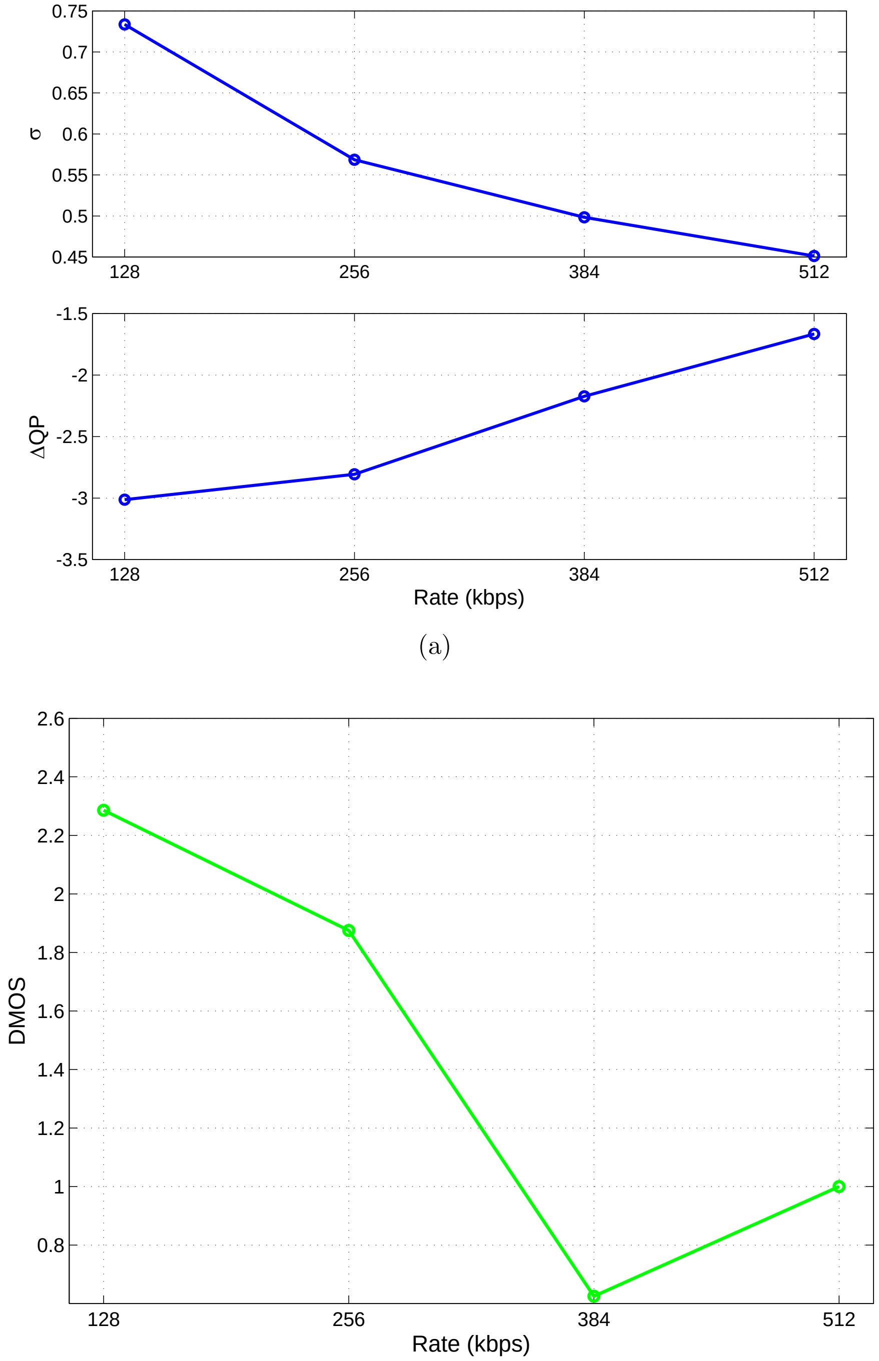
Subjective Test. Results Sheet

Subjective Test ST1. Results Sheet

Test 001:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 002:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 003:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 004:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 005:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 006:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 007:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 008:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 009:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 010:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 011:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 012:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 013:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 014:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 015:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 016:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 017:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □
Test 018:	Best A □ Best B □	**Imp. Score**	0: □ 1: □ 2: □ 3: □ 4: □

Appendix B

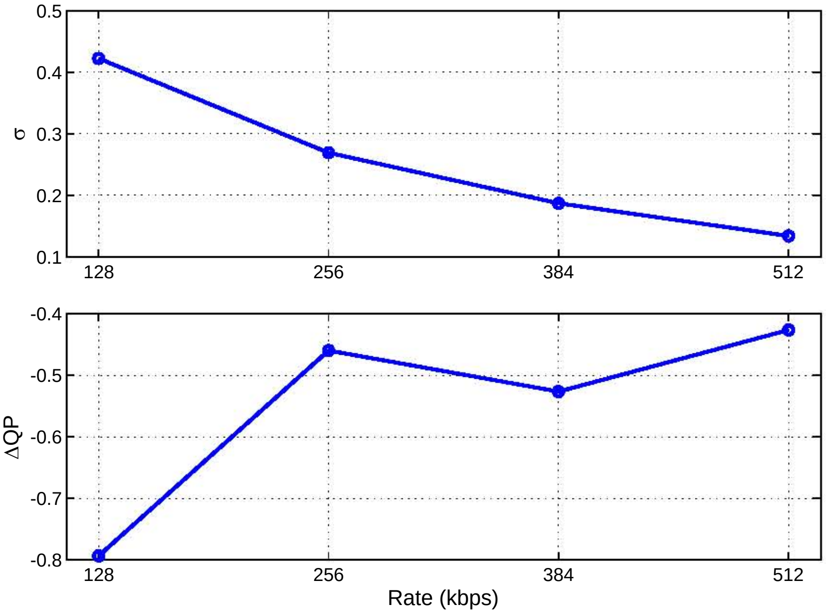
Subjective Test Results per sequence



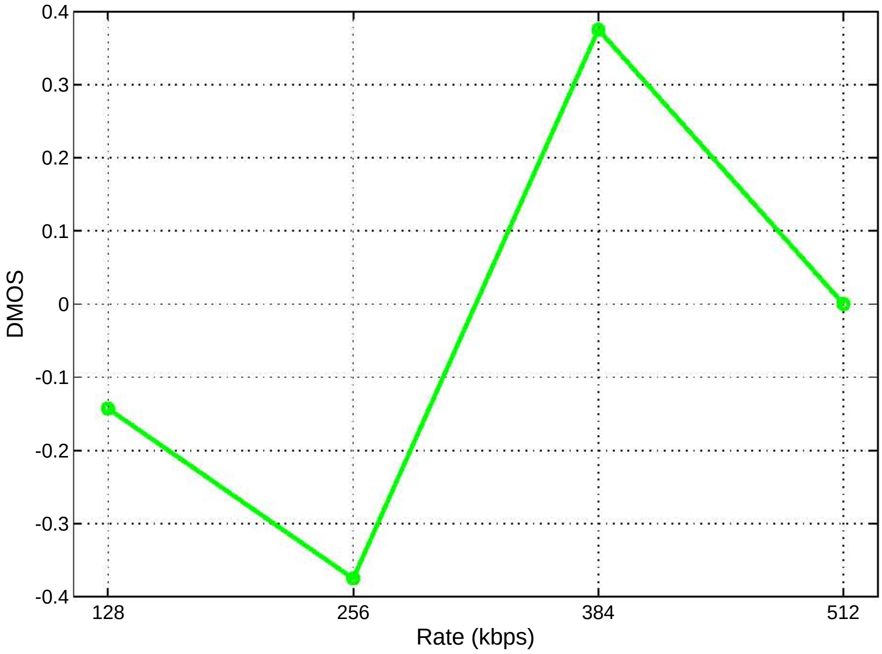
(a)

(b)

Figure B.1: Results for the sequence *flower* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.

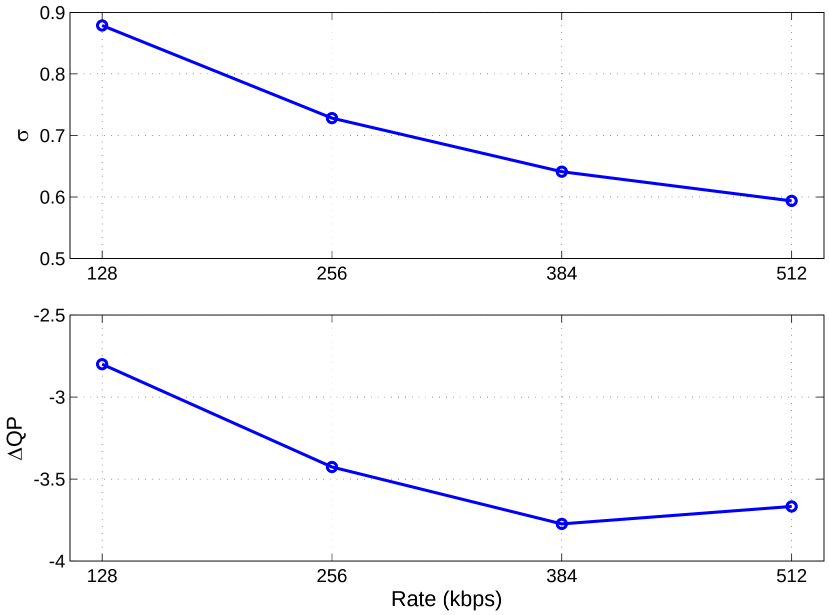


(a)

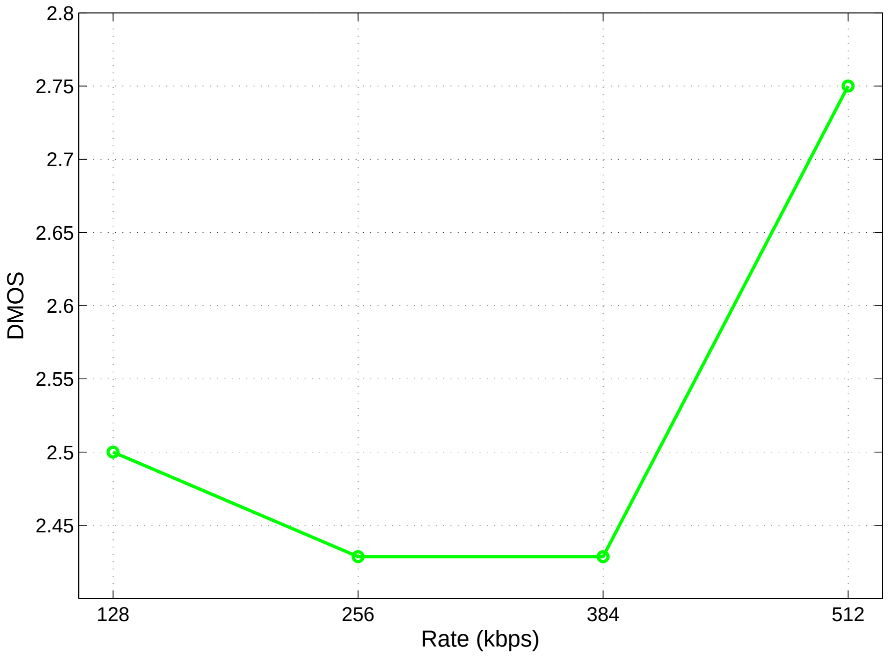


(b)

Figure B.2: Results for the sequence *hall* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.

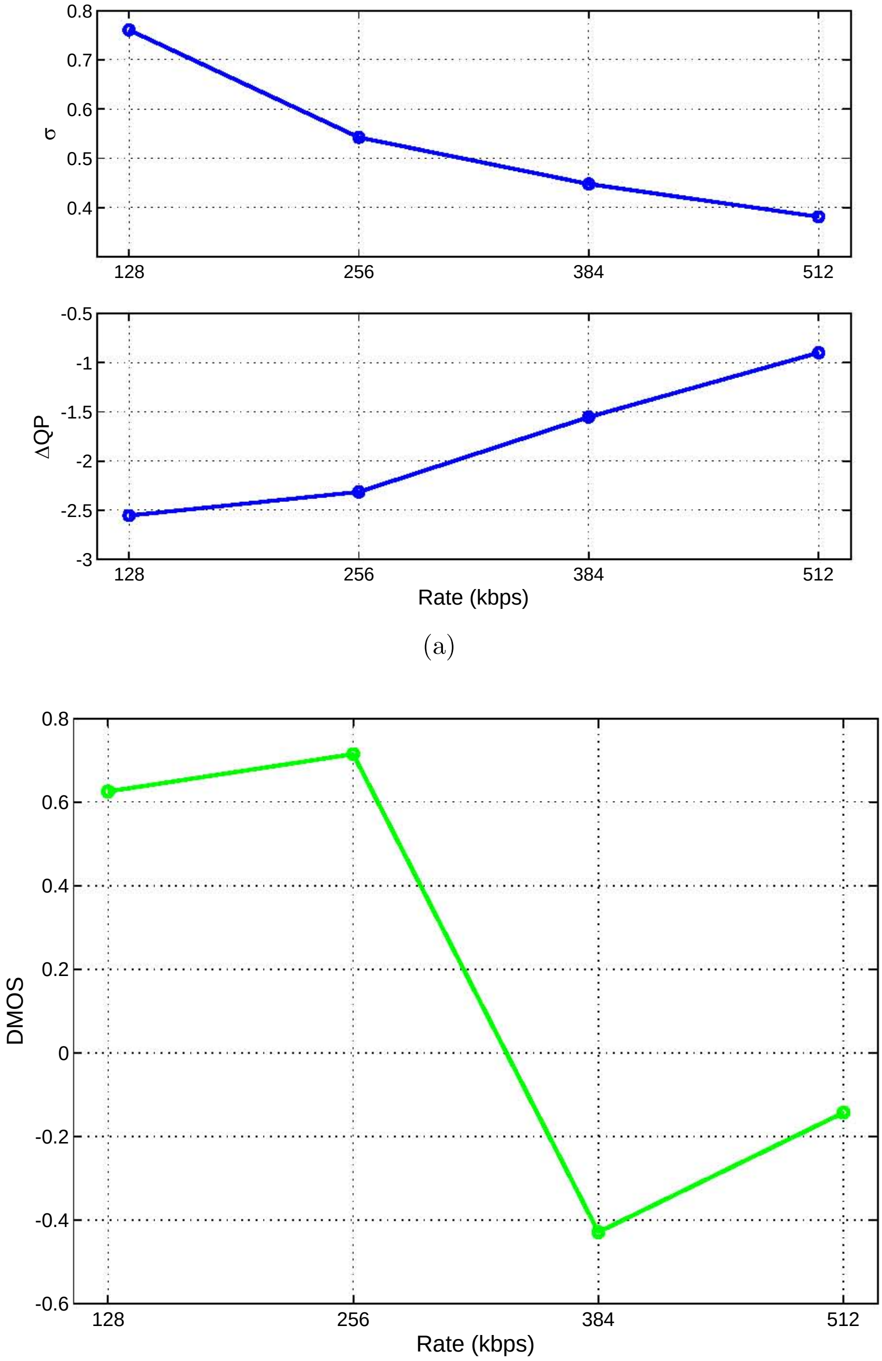


(a)



(b)

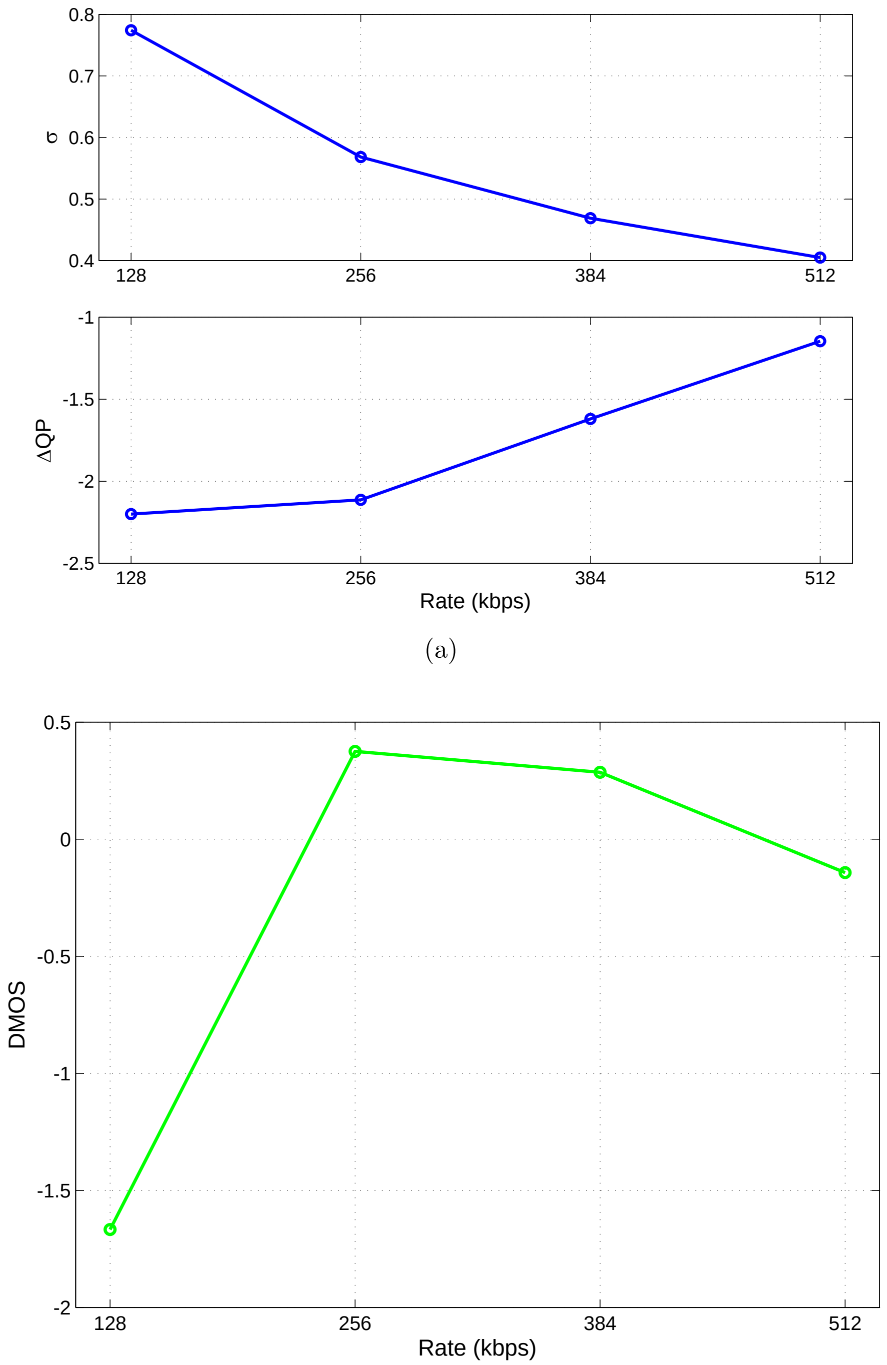
Figure B.3: Results for the sequence *husky* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.



(a)

(b)

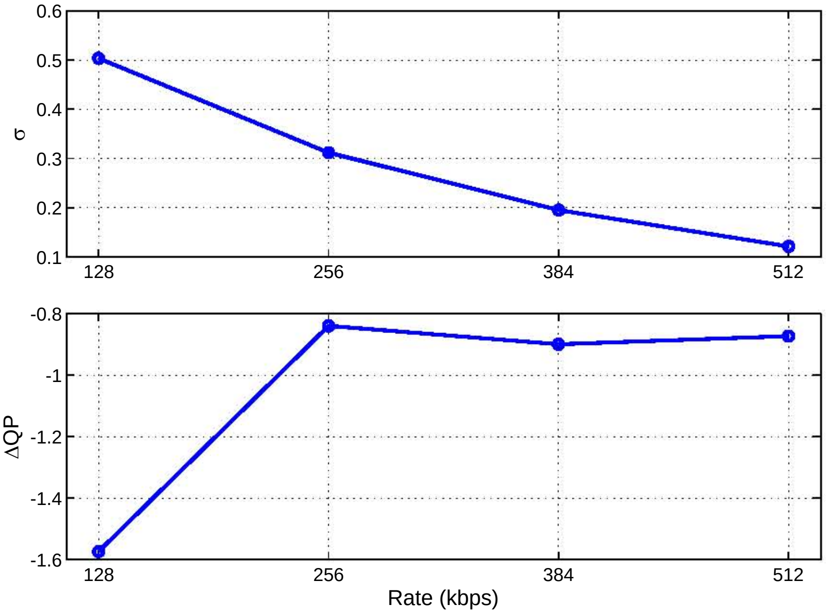
Figure B.4: Results for the sequence *ice* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.



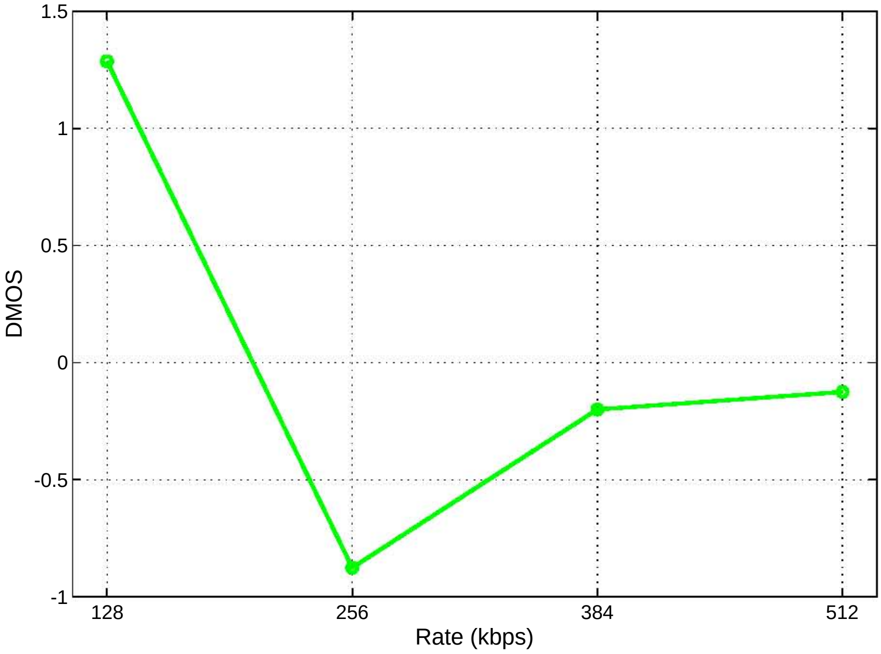
(a)

(b)

Figure B.5: Results for the sequence *irene* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.

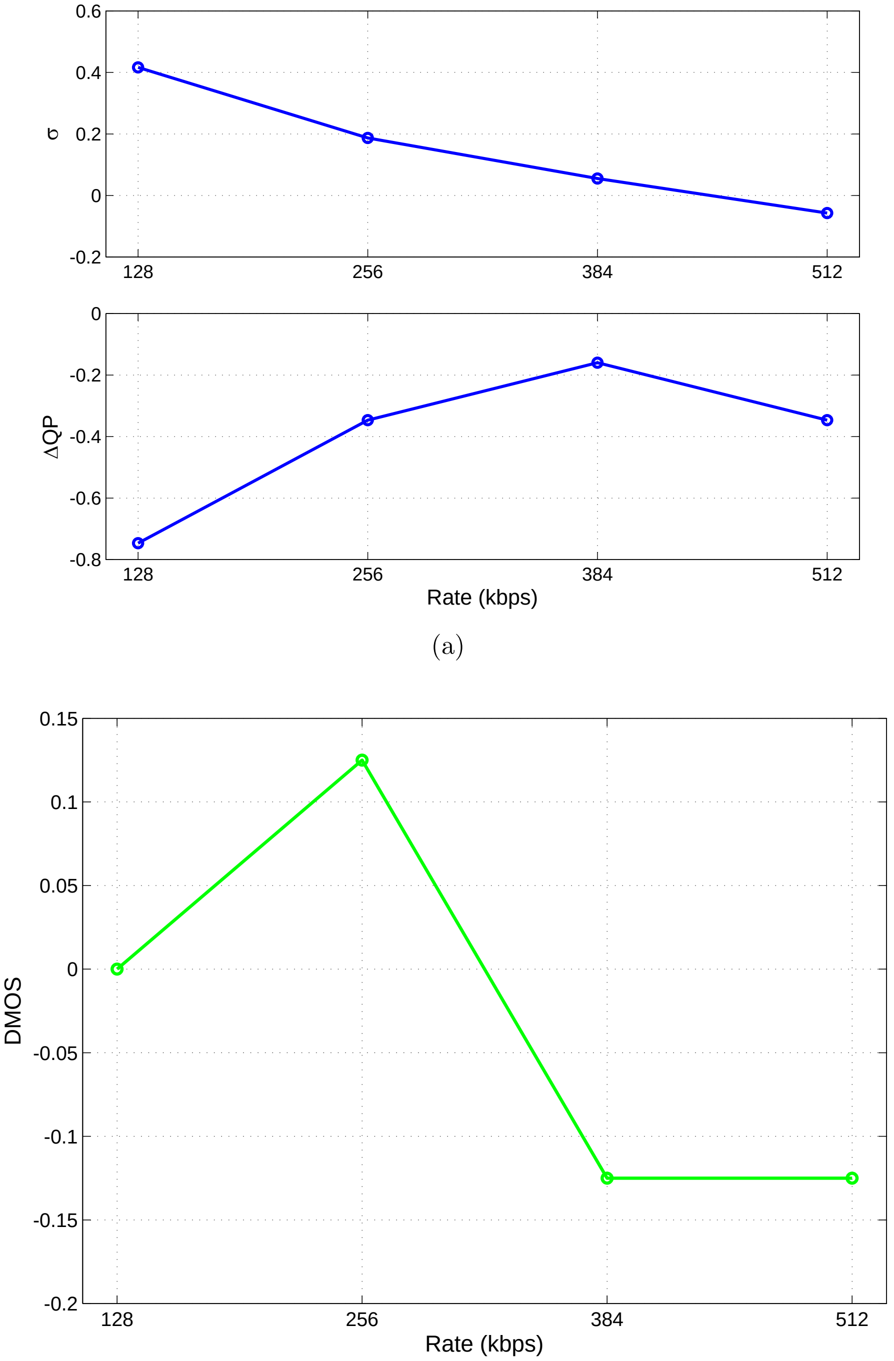


(a)



(b)

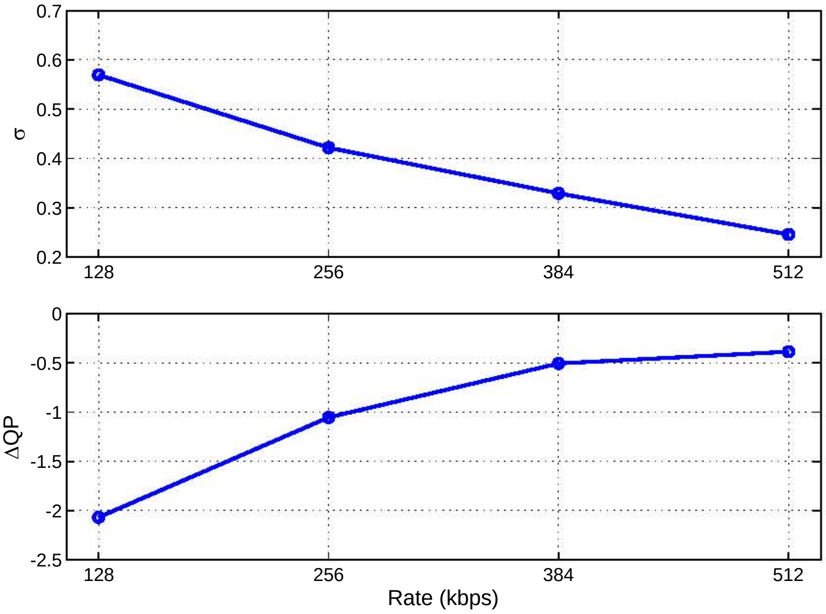
Figure B.6: Results for the sequence *mother&daughter* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.



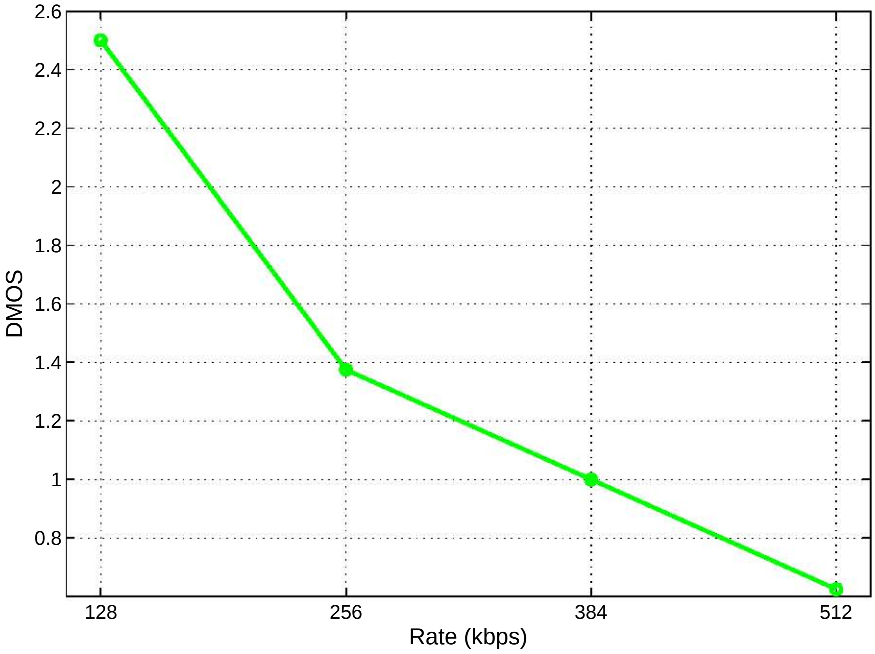
(a)

(b)

Figure B.7: Results for the sequence *news* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.

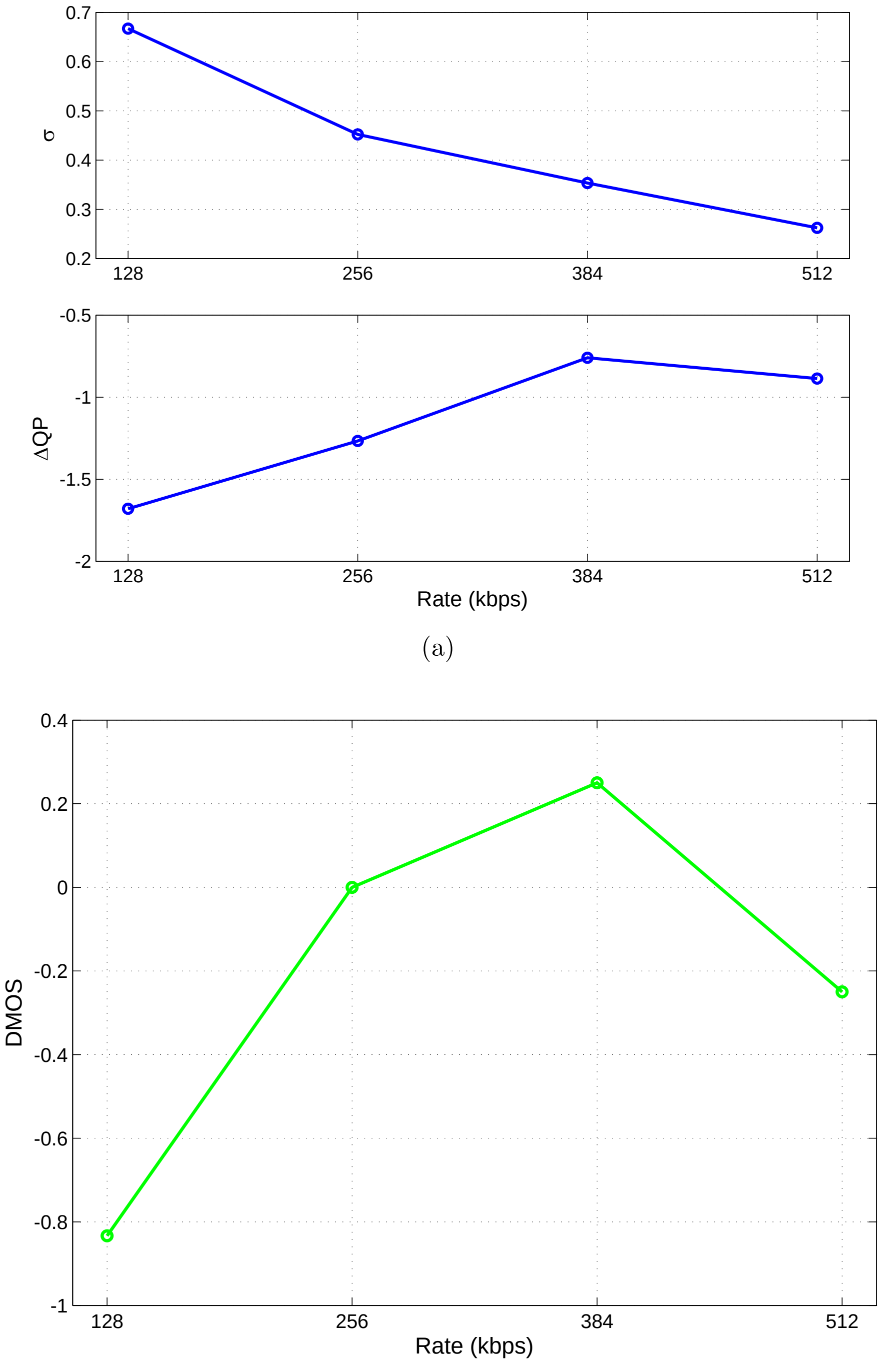


(a)



(b)

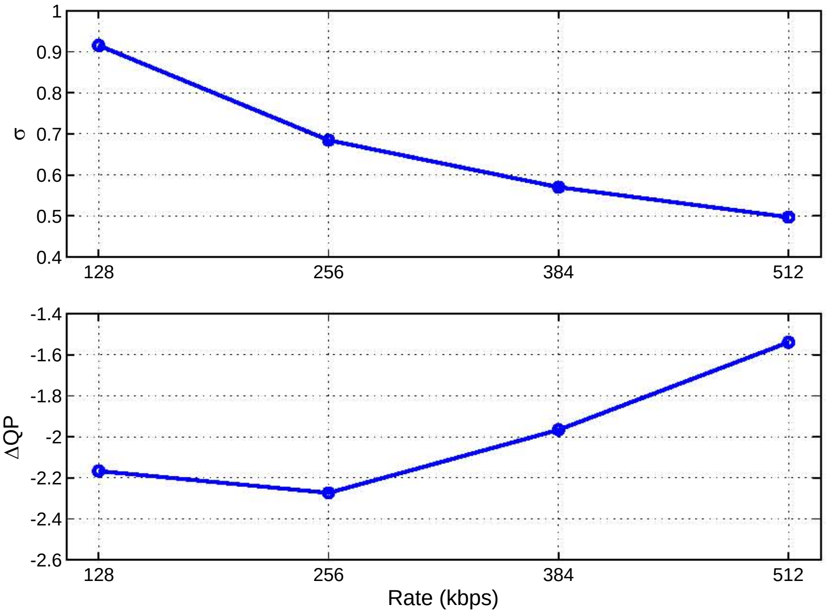
Figure B.8: Results for the sequence *paris* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.



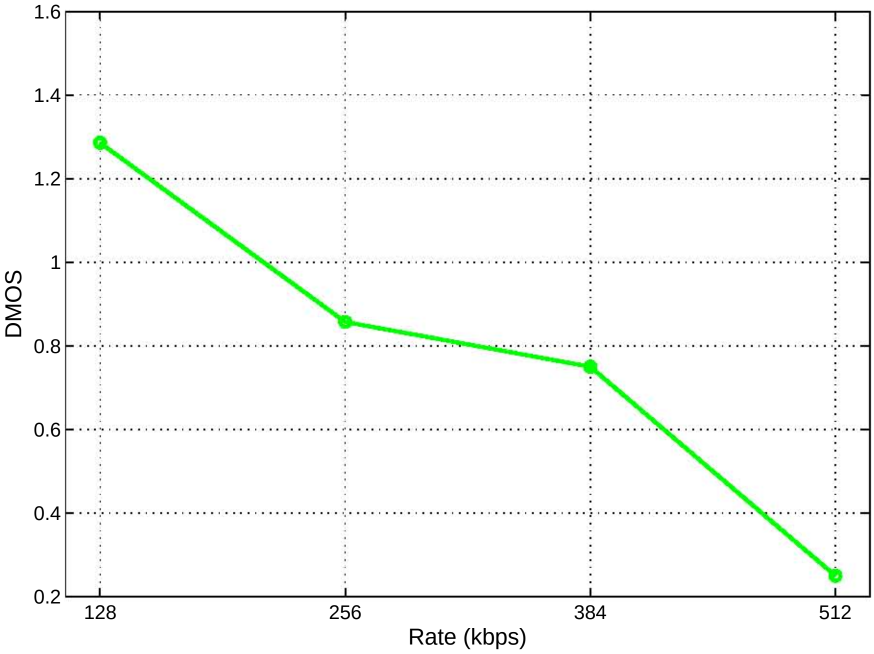
(a)

(b)

Figure B.9: Results for the sequence *silent* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.

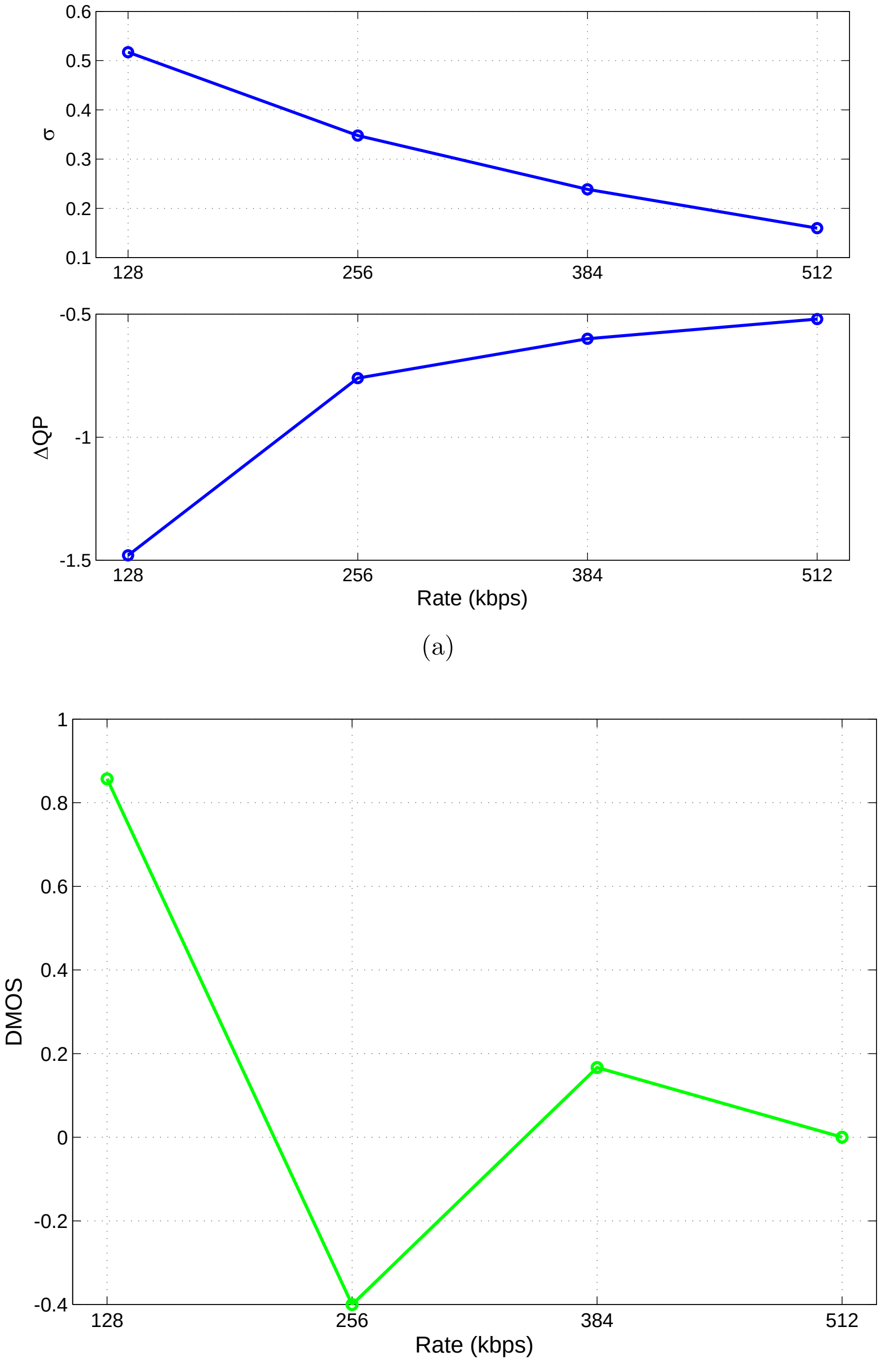


(a)



(b)

Figure B.10: Results for the sequence *soccer* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.



(a)

(b)

Figure B.11: Results for the sequence *students* at different bit rates: (a) filter strength and QP reduction; (b) average DMOS score with respect to non-filtered version.

Bibliography

[Aase et al., 1999] Aase, S., Husoy, J., and Waldemar, P. (1999). A critique of svd-based image coding systems. In *Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on*, volume 4, pages 13–16.

[ATSC, 2008] ATSC (2008). *Mobile/Handheld Standard A/153. Part 7: AVC and SVC Video System Characteristics.*

[Avaro et al., 1997] Avaro, O., Chou, P. A., Eleftheriadis, A., Herpel, C., Reader, C., and Signès, J. (1997). The MPEG-4 systems and description languages: A way ahead in audio visual information representation. *Signal Processing: Image Communication*, 9(4):385–431.

[Balle, 2010] Balle, J. (2010). Image simplification by frequency-selective means filtering. In *Picture Coding Symposium (PCS), 2010*, pages 126–129.

[Bovik, 2009] Bovik, A. C. (2009). *The Essential Guide to Image Processing.* Academic Press.

[Brownrigg, 1984] Brownrigg, D. R. K. (1984). The weighted median filter. *Commun. ACM*, 27(8):807–818.

[Buades et al., 2005] Buades, A., Coll, B., and Morel, J. M. (2005). A Review of Image Denoising Algorithms, with a New One. *Multiescale Model. Simul*, 4:490–530.

[Burger and Harmeling, 2011] Burger, H. and Harmeling, S. (2011). Improving denoising algorithms via a multi-scale meta-procedure.

[Bystrom et al., 2010] Bystrom, M., Richardson, I., Kannangara, S., and de Frutos-Lopez, M. (2010). Dynamic replacement of video coding elements. *Signal Processing: Image Communication*, 25(4):303–313.

[Chen et al., 1977] Chen, W.-H., Smith, C., and Fralick, S. (1977). A fast computational algorithm for the Discrete Cosine Transform. *Communications, IEEE Transactions on*, 25(9):1004–1009.

[de Frutos-Lopez et al., 2010] de Frutos-Lopez, M., del Ama-Esteban, O., Sanz-Rodriguez, S., and Diaz-de Maria, F. (2010). A two-level sliding-window VBR controller for real-time hierarchical video coding. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 4217–4220.

[de Frutos-Lopez et al., 2012] de Frutos-Lopez, M., Medina-Chanca, H., Sanz-Rodríguez, S., Peláez-Moreno, C., and Díaz-de María, F. (2012). Perceptually-aware bilateral filtering for quality improvement in low bit rate video coding. In *Picture Coding Symposium (PCS), 2012*.

[de Frutos-López et al., 2010] de Frutos-López, M., Orellana-Quirós, D., Pujol-Alcolado, J. C., and Díaz-de María, F. (2010). An improved fast mode decision algorithm for intraprediction in H.264/AVC video coding. *Signal Processing: Image Communication*, 25(10):709–716.

[Ding et al., 2009] Ding, D., Qi, H., Yu, L., Huang, T., and Gao, W. (2009). Reconfigurable video coding framework and decoder reconfiguration instantiation of AVS. *Sig. Proc.: Image Comm.*, pages 287–299.

[Dony and Haykin, 1995] Dony, R. and Haykin, S. (1995). Optimally adaptive transform coding. *Image Processing, IEEE Transactions on*, 4(10):1358–1370.

[Effros and Chou, 1995] Effros, M. and Chou, P. (1995). Weighted Universal Tansform Coding: universal image compression with the Karhunen-Loeve transform. In *Image Processing, 1995. Proceedings., International Conference on*, volume 2, pages 61–64 vol.2.

[Eker and Janneck, 2003] Eker, J. and Janneck, J. W. (2003). CAL language report specification of the CAL Actor Language. Technical Report UCB/ERL M03/48, EECS Department, University of California, Berkeley.

[Florencio, 2001] Florencio, D. (2001). Motion sensitive pre-processing for video. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 399–402.

[Gopalan, 2009] Gopalan, R. (2009). Exploiting region of interest for improved video coding. Master's thesis, Electrical and Computer Engineering, Ohio State University.

[Gorin et al., 2011a] Gorin, J., Wipliez, M., Prêteux, F., and Raulet, M. (2011a). LLVM-based and scalable MPEG-RVC decoder. *J. Real-Time Image Process.*, 6(1):59–70.

[Gorin et al., 2011b] Gorin, J., Yviquel, H., Prêteux, F., and Raulet, M. (2011b). Just-in-time adaptive decoder engine: a universal video decoder based on MPEG RVC. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 711–714, New York, NY, USA. ACM.

[ISO/IEC, 1994] ISO/IEC (1994). Generic coding of Moving pictures and associated audio information - Part 2: Video. *ITU-T Recommendation H.262-ISO/IEC 13818-2, MPEG-2.*

[ITU-T, 1993] ITU-T (1993). *ITU-T Recommendation H.261: Line Transmission of Non-Telephone Signals : Video Codec for Audiovisual Services at p x 64 kbits.* International Telecommunication Union.

[ITU-T, 1995] ITU-T (1995). *Video coding for low bitrate communication.* International Telecommunication Union.

[ITU-T, 2008] ITU-T (2008). P-910: Subjective video quality assessment methods for multimedia applications. Technical report, International Telecommunication Union.

[Jing et al., 2008] Jing, X., Chau, L.-P., and Siu, W.-C. (2008). Frame complexity-based Rate-Quantization model for H.264/AVC intraframe Rate Control. *Signal Processing Letters, IEEE*, 15:373–376.

[JVT, 2003] JVT (2003). Advanced Video Coding for generic audiovisual services. *ITU-T Recommendation International Standard of Joint Video Specification, ITU-T Rec. H.264/ISO/IEC 14496-10 AVC, Version 1, JVT-G50.*

[Kamaci and Altunbasak, 2005] Kamaci, N. and Altunbasak, Y. (2005). Frame bit allocation for H.264 using Cauchy-distribution based source modelling. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 2, pages 57–60.

[Kamisli and Lim, 2009] Kamisli, F. and Lim, J. (2009). Transforms for the motion compensation residual. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 789–792.

[Kannangara et al., 2010] Kannangara, C., Philp, J., Richardson, I., Bystrom, M., and de Frutos Lopez, M. (2010). A syntax for defining, communicating, and implementing video decoder function and structure. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(9):1176–1186.

[Kannangara et al., 2008] Kannangara, S., Richardson, I., Bystrom, M., and de Frutos, M. (2008). Fast, dynamic configuration of transforms for video coding. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1592–1595.

[Karlsson and Sjostrom, 2005] Karlsson, L. and Sjostrom, M. (2005). Improved ROI video coding using variable Gaussian pre-filters and variance in intensity. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–313–16.

[Karunaratne et al., 2001] Karunaratne, P., Segall, C., and Katsaggelos, A. (2001). A rate-distortion optimal video pre-processing algorithm. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 1, pages 481–484.

[Kawada et al., 2006] Kawada, R., Koike, A., and Nakajima, Y. (2006). Prefilter Control scheme for low bitrate TV distribution. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 769–772.

[Kim et al., 2010] Kim, J.-H., Lee, J. W., Park, R.-H., and Park, M.-H. (2010). Adaptive edge-preserving smoothing and detail enhancement for video preprocessing of H.263. In *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on*, pages 337–338.

[Ko and Lee, 1991] Ko, S.-J. and Lee, Y. (1991). Center weighted median filters and their applications to image enhancement. *Circuits and Systems, IEEE Transactions on*, 38(9):984–993.

[Li and Xu, 2006] Li, M.-q. and Xu, Z.-q. (2006). An adaptive preprocessing algorithm for low bitrate video coding. *Journal of Zhejiang University - Science A*, 7(12):2057–2062.

[Lin and Ortega, 1997] Lin, L.-J. and Ortega, A. (1997). Perceptually based video rate control using pre-filtering and predicted rate-distortion characteristics. In *Image Processing, 1997. Proceedings., International Conference on*, volume 2, pages 57–60.

[List et al., 2003] List, P., Joch, A., Lainema, J., Bjontegaard, G., and Karczewicz, M. (2003). Adaptive deblocking filter. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):614–619.

[Lu and Zhang, 2011] Lu, S.-P. and Zhang, S.-H. (2011). Saliency-based fidelity adaptation preprocessing for video coding. *Journal of Computer Science and Technology*, 26(1):195–202.

[Ma et al., 2003] Ma, S., Li, Z., and We, F. (2003). Proposed draft of Adaptive Rate Control. *JVT-H017, 8th JVT Meeting.*

[Mattavelli et al., 2010] Mattavelli, M., Amer, I., and Raulet, M. (2010). The reconfigurable video coding standard [standards in a nutshell]. *Signal Processing Magazine, IEEE*, 27(3):159–167.

[Minoo and Nguyen, 2005] Minoo, K. and Nguyen, T. (2005). Perceptual video coding with H.264. *Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on*, pages 741–745.

[Narroschke, 2010] Narroschke, M. (2010). Quantization noise reduction in hybrid video coding by a system of three adaptive filters. In *Picture Coding Symposium (PCS), 2010*, pages 314–317.

[Ortega, 2000] Ortega, A. (2000). *Compressed Video over Networks*, chapter Variable Bit-Rate Video Coding, pages 343–382. Marcel Dekker, New York, NY,USA.

[Ortega and Ramchandran, 1998] Ortega, A. and Ramchandran, K. (1998). Rate-Distortion methods for image and video compression. *Signal Processing Magazine, IEEE*, 15(6):23–50.

[Ozkan et al., 1993] Ozkan, M., Sezan, M., and Tekalp, A. (1993). Adaptive motion-compensated filtering of noisy image sequences. *Circuits and Systems for Video Technology, IEEE Transactions on*, 3(4):277–290.

[Pham and van Vliet, 2005] Pham, T. and van Vliet, L. (2005). Separable bilateral filtering for fast video preprocessing. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 1–4.

[Philp et al., 2009] Philp, J. M., Kannangara, C. S., Bystrom, M., de Frutos Lopez, M., and Richardson, I. E. (2009). Decoder Description Syntax for Fully Configurable Video Coding. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 769–772.

[Qi et al., 2006] Qi, Y., HUANG, Y.-G., and QI, H.-G. (2006). Pre-processing for video coduing with Rate-Distortion Optimization decision. *The Journal of China Universities of Posts and Telecommunications*, 13(2):79–83.

[Rajashekar and Simoncelli, 2009] Rajashekar, U. and Simoncelli, E. P. (2009). Multiscale denoising of photographic images. In Bovik, A., editor, *The Essential Guide to Image Processing (Second Edition)*, pages 241–261. Academic Press, Boston.

[Richardson et al., 2008] Richardson, I., Bystrom, M., de Frutos, M., and Kannangara, S. (2008). Dynamic transform replacement in an H.264 codec. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 2108–2111.

[Richardson et al., 2009a] Richardson, I., Kannangara, S., Bystrom, M., Philp, J., and de Frutos Lopez, M. (2009a). A framework for Fully Configurable Video Coding. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1–4.

[Richardson et al., 2009b] Richardson, I., Kannangara, S., Bystrom, M., Philp, J., and de Frutos Lopez, M. (2009b). Implementing Fully Configurable Video Coding. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 765–768.

[Richardson et al., 2009c] Richardson, I., Kannangara, S., Bystrom, M., Philp, J., and Y., Z. (2009c). *Fully Configurable Video Coding, Part 2: A Proposed Platform for Reconfigurable Video.* ISO/IEC JTC1/SC29/WG11.

[Richardson, 2003] Richardson, I. E. (2003). *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia.* John Wiley and Sons.

[Roeser and Jernigan, 1982] Roeser, P. and Jernigan, M. (1982). Fast Haar Transform algorithms. *Computers, IEEE Transactions on*, C-31(2):175–177.

[RVC, 2011a] RVC, I. (2011a). ISO/IEC 23001-4:2011: Information Technology: MPEG systems technologies-Part 4: Codec Configuration Representation. Technical report, ISO/IEC.

[RVC, 2011b] RVC, I. (2011b). ISO/IEC 23002-4:2010/amd 1:2011: Information Technology: MPEG video technologies-Part 4: Video Tool Library. Technical report, ISO/IEC.

[Sanz-Rodríguez et al., 2010] Sanz-Rodríguez, S., del-Ama-Esteban, O., de-Frutos-López, M., and Díaz-de-María, F. (2010). Cauchy-density-based basic unit layer rate controller for H.264/AVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(8):1139–1143.

[Segall and Katsaggelos, 2000] Segall, C. and Katsaggelos, A. (2000). Pre- and post-processing algorithms for compressed video enhancement. In *Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on*, volume 2, pages 1369–1373.

[Segall et al., 2001] Segall, C. A., Karunaratne, P. V., and Katsaggelos, A. K. (2001). Preprocessing of compressed digital video. In *VCIP*, pages 163–174.

[Siekmann et al., 2010] Siekmann, M., Bosse, S., Schwarz, H., and Wiegand, T. (2010). Separable wiener filter based adaptive in-loop filter for video coding. In *Picture Coding Symposium (PCS), 2010*, pages 70–73.

[Song et al., 2004] Song, W.-S., Kim, D.-R., Lee, S., and Hong, M.-C. (2004). A modified Gaussian model-based low complexity pre-processing algorithm for H.264 video coding standard. In Roca, V. and Rousseau, F., editors, *Interactive Multimedia and Next Generation Networks*, volume 3311 of *Lecture Notes in Computer Science*, pages 61–71. Springer Berlin Heidelberg.

[Sullivan and Wiegand, 1998] Sullivan, G. and Wiegand, T. (1998). Rate-Distortion Optimization for video compression. *Signal Processing Magazine, IEEE*, 15(6):74–90.

[Sullivan, 2004] Sullivan, G. J. (2004). The H.264/AVC Advanced Video Coding standard: overview and introduction to the Fidelity Range Extensions. *Proceedings of SPIE*, 5558:454–474.

[Tan et al., 1996] Tan, S. H., Pang, K., and Ngan, K. (1996). Classified perceptual coding with adaptive quantization. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(4):375–388.

[Tomasi and Manduchi, 1998] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846.

[van den Branden Lambrecht and Verscheure, 1996] van den Branden Lambrecht, C. J. and Verscheure, O. (1996). Perceptual quality measure using a spatiotemporal model of the human visual system. In Bhaskaran, V., Sijstermans, F., and Panchanathan, S., editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2668 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 450–461.

[Wang et al., 2004a] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004a). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612.

[Wang et al., 2004b] Wang, Z., Lu, L., and Bovik, A. C. (2004b). Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication*, 19(2):121–132.

[Watson et al., 2001] Watson, A., Hu, Q., and McGowan, J. (2001). Digital Video Quality Metric based on human vision. *Journal of Electronic Imaging*, 10:20–29.

[Wiegand et al., 2003] Wiegand, T., Schwarz, H., Joch, A., Kossentini, F., and Sullivan, G. (2003). Rate-constrained coder control and comparison of video coding standards. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):688–703.

[Winkler, 1999] Winkler, S. (1999). Issues in vision modeling for perceptual Video Quality Assessment. *Signal Processing*, 78(2):231–252.

[Wu and Rao, 2005] Wu, H. R. and Rao, K. R. (2005). *Digital Video Image Quality and Perceptual Coding (Signal Processing and Communications)*. CRC Press, Inc., Boca Raton, FL, USA.

[Xu et al., 2007] Xu, J., Sclabassi, R., Liu, Q., Chaparro, L., Marchessault, R., and Sun, M. (2007). Human perception based video preprocessing for telesurgery. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 3086–3089.

[Yim and Bovik, 2011] Yim, C. and Bovik, A. C. (2011). Evaluation of temporal variation of video quality in packet loss networks. *Signal Processing: Image Communication*, 26(1):24–38.

[Yokoyama and Ooi, 1999] Yokoyama, Y. and Ooi, Y. (1999). A scene-adaptive one-pass Variable Bit Rate video coding method for storage media. In *Image Processing, 1999. ICIP 99*, volume 3, pages 827–831.

[Yu et al., 2005] Yu, H., Pan, F., Lin, Z., and Sun, Y. (2005). A perceptual bit allocation scheme for H.264. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 1–4.