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

TESIS DOCTORAL

A RATE CONTROL ALGORITHM FOR SCALABLE VIDEO CODING

Autor: Sergio Sanz Rodríguez-Escalona
Director: Dr. Fernando Díaz de María

Leganés, 2011

TESIS DOCTORAL:
A RATE CONTROL ALGORITHM FOR SCALABLE VIDEO CODING.

Autor:
SERGIO SANZ RODRÍGUEZ-ESCALONA

Director:
DR. FERNANDO DÍAZ DE MARÍA

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

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

acuerda otorgarle la calificación de

Leganés, a

Abstract

This thesis proposes a rate control (RC) algorithm for H.264/scalable video coding (SVC) specially designed for real-time variable bit rate (VBR) applications with buffer constraints. The VBR controller assumes that consecutive pictures within the same scene often exhibit similar degrees of complexity, and aims to prevent unnecessary quantization parameter (QP) fluctuations by allowing for just an incremental variation of QP with respect to that of the previous picture. In order to adapt this idea to H.264/SVC, a rate controller is located at each dependency layer (spatial or coarse grain scalability) so that each rate controller is responsible for determining the proper QP increment. Actually, one of the main contributions of the thesis is a QP increment regression model that is based on Gaussian processes. This model has been derived from some observations drawn from a discrete set of representative encoding states. Two real-time application scenarios were simulated to assess the performance of the VBR controller with respect to two well-known RC methods. The experimental results show that our proposal achieves an excellent performance in terms of quality consistency, buffer control, adjustment to the target bit rate, and computational complexity.

Moreover, unlike typical RC algorithms for SVC that only satisfy the hypothetical reference decoder (HRD) constraints for the highest temporal resolution sub-stream of each dependency layer, the proposed VBR controller also delivers HRD-compliant sub-streams with lower temporal resolutions. To this end, a novel approach that uses a set of buffers (one per temporal resolution sub-stream) within a dependency layer has been built on top of the RC algorithm. The proposed approach aims to simultaneously control the buffer levels for overflow and underflow prevention, while maximizing the reconstructed video quality of the corresponding sub-streams. This in-layer multi-buffer framework for rate-controlled SVC does not require additional dependency layers to deliver different HRD-compliant temporal resolutions for a given video source, thus improving the coding efficiency when compared to typical SVC encoder configurations since, for the same target bit rate, less layers are encoded.

Resumen extendido en castellano

Motivación

Durante estos últimos años, las aplicaciones de vídeo han crecido en popularidad debido a los grandes avances tecnológicos logrados en codificación de vídeo, estandarización, infraestructuras de red, capacidad de almacenamiento y capacidad computacional de los receptores multimedia. Las áreas de aplicación relacionadas con vídeo más conocidas en la actualidad incluyen mensajes multimedia, videoconferencia, *streaming*, televisión digital estándar (*standard-definition* TV, SDTV) y en alta definición (*high-definition* TV, HDTV), así como también almacenamiento multimedia en discos ópticos como el *Digital Versatile Disk* y el *Blu-Ray Disk*.

Algunas de estas aplicaciones, como las que ofrecen servicios de TV vía satélite, cable, o terrestre, hacen uso de los sistemas tradicionales de transmisión multimedia para enviar formatos espacio-temporales prefijados de la señal de vídeo (SDTV@25 Hz, HDTV@50 Hz, etc.). Esto significa que para otras resoluciones distintas el *bit stream* no puede ser directamente descodificado. Pese a que en general proporcionan una buena calidad de servicio (*quality of servic*e, QoS), los sistemas tradicionales presentan solamente dos estados de conexión: transmite o no transmite.

Sin embargo, los sistemas modernos basados en transmisión RTP/IP (*real-time transport protocol/Internet protocol*), tales como Internet o las redes inalámbricas, se caracterizan por una oferta mucho más amplia de QoS; oferta que se deriva, por un lado, de la propia red de transmisión de ancho de banda variable y, por otro lado, de

la demanda de servicios audiovisuales a terminales tan heterogéneos como pueden ser los teléfonos móviles, ordenadores personales o receptores HDTV. Normalmente estas QoS están caracterizadas por una tasa (o calidad) objetivo y una resolución spacio-temporal determinadas.

Dentro de este último marco tecnológico para la entrega de contenidos audiovisuales, la codificación de vídeo escalable (*scalable video coding*, SVC) proporciona una solución atractiva a las características inherentes de estos sistemas de transmisión RTP/IP. Concretamente, SVC permite que una determinada QoS pueda adaptarse a unas condiciones de red variables, necesidades o preferencias del destinatario final, todas ellas mediante el uso de ciertas herramientas escalables que modifan el grado de fidelidad o relación señal-ruido (*signal-to-noise ratio*, SNR) del vídeo reconstruido, el tamaño de imagen (resolución espacial) o la frecuencia de cuadro (resolución temporal). Otra propiedad de SVC es que la información codificada puede entregarse simultáneamente a una variedad de receptores, cada uno con su propia QoS.

Un *bit stream* es llamado escalable cuando partes del mismo pueden borrarse de manera que el *sub-stream* resultante forma otro *bit stream* válido para algún terminal objetivo, o sencillamente codificado a una tasa objetivo menor. Ese *sub-stream* representa un contenido de fuente codificado con una calidad de reconstrucción más baja que la correspondiente al *bit stream* completo, aunque más alta que la de cualquier otro *sub-stream* incluido en él. Los *bit streams* que no proporcionan propiedades escalables son denominados de una sola capa o *single-layer*.

Varios estándares de codificación de vídeo, tales como MPEG-2, H.263, MPEG-4 Visual, y H.264/*advanced video coding* (AVC), han incorporado herramientas escalables en sus últimas extensiones. Los modos de escalabilidad más comunes son los siguientes: espacial, temporal y en calidad (o SNR). La escalabilidad espacial proporciona *sub-streams* que representan el contenido de vídeo codificado con resoluciones espaciales menores. La escalabilidad temporal se encarga de generar *sub-streams* con resoluciones temporales menores para una resolución espacial dada. Por último, la escalabilidad SNR describe casos en que los *sub-streams* presentan la misma resolución

espacio-temporal que la correspondiente al *bit stream* completo, pero codificadas a calidades más bajas.

La extensión escalable de H.264/AVC, también llamado H.264/SVC, ha sido recientemente estardarizada [Schwarz et al., 2007, Wien et al., 2007b]. H.264/SVC es capaz de proporcionar una eficiencia de codificación y una complejidad de descodificación similares a los conseguidos por una codificación no escalable. Concretamente, en este estándar la escalabilidad espacial emplea una estructura codificación basada en capas para entregar versiones codificadas de una secuencia de vídeo con diferentes resoluciones espaciales. La capa base proporciona un *sub-stream* con la resolución espacial más baja y compatible con H.264/AVC, mientras que las capas realzadas codifican la fuente de vídeo con tamaños de imagen mayores.

Asimismo, cada capa espacial puede incluir escalabilidad temporal mediante el uso de estructuras de predicción jerárquicas que codifican los planos en base a capas. La capa temporal base representa una secuencia codificada a la frecuencia de cuadro más baja; la siguiente capa temporal añade los planos que faltan para formar una versión codificada a una frecuencia de cuadro igual al doble de la anterior; y así se van añadiendo sucesivamente capas hasta alcanzar la mayor resolución temporal para una determinada capa espacial.

La escalabilidad en calidad genera diferentes calidades de reconstrucción para una resolución espacio-temporal concreta. En particular, H.264/SVC define dos tipos de codificación escalable de tipo SNR: la escalabilidad *coarse grain scalability* (CGS), y la escalabilidad *medium grain scalability* (MGS). La codificación escalable CGS es un caso especial de escalabilidad espacial donde los tamaños de imagen son idénticos entre capas consecutivas. Estas capas espaciales/CGS son también denominadas capas de dependencia. La codificación escalable MGS emplea una configuración también basada en capas dentro de una capa de dependencia para proporcionar un mayor abanico de tasas objetivo dentro del *bit stream* escalable.

Estos tipos de escalabilidad pueden combinarse en una misma codificación, por lo que el número de capas de un codificador H.264/SVC debe ser adecuadamente con-

figurado en función de la cantidad de resoluciones espacio-temporales y/o calidades de reconstrucción objetivo impuestas por la aplicación.

Como ya se ha indicado, la tasa de bits objetivo está estrechamente ligada a la QoS, de manera que la tasa de salida de un codificador de vídeo es la magnitud clave a controlar con el fin de satisfacer los requerimientos de la aplicación. Por ello se emplaza en el codificador un algoritmo de control de tasa (*rate control*, RC), el cual opera en dos pasos. En primer lugar, un presupesto de bits objetivo es asignado a un segmento de vídeo a partir de las características espaciales y temporales del vídeo, de la tasa objetivo y de las restricciones del *buffer* de transmisión impuestas por el llamado descodificador hipotético de referencia (*hypothetical reference decoder*, HRD) [Ribas-Corbera et al., 2003]. Sin embargo, en una aplicación de almacenamiento de vídeo digital esas restricciones vienen marcadas principalmente por la capacidad máxima de almacenamiento del dispositivo multimedia. Una vez obtenido el presupuesto de bits para el segmento de vídeo, el segundo paso de operación asigna un valor para el parámetro de cuantificación (*quantization parameter*, QP) tal que se satisfagan las restricciones de *buffer* y/o de almacenamiento y, además, se maximice en la medida de lo posible la calidad del vídeo reconstruido. En el caso de SVC, un esquema de RC en realidad se compone de varios módulos de RC, cada uno situado en cada capa espacial o SNR, para que el *bit stream* escalable completo pueda recoger las diferentes QoS encaminadas a los diversos receptores objetivo.

Según el tipo de aplicación, dos métodos de RC pueden distinguirse: controladores de tasa de bit constante (*constant bit rate*, CBR) y controladores de tasa de bit variable (*variable bit rate*, VBR). En los algoritmos de CBR, comunmente usados para videoconferencia, se require un ajuste muy a corto plazo de la tasa objetivo con el fin de asegurar un retardo bajo asociado al *buffer*. Sin embargo, en los algoritmos de VBR, típicamente usados para *streaming* o almacenamiento de vídeo digital, una adaptación más a largo plazo de la tasa media respecto de la objetivo es factible para asegurar una mejor calidad visual, aunque a costa de un mayor retardo de *buffer*.

Muchos esquemas de RC se han propuesto en la literatura para dar soluciones

fiables, no sólo a las singularidades de cada estándar de codificación, sino también a los requerimientos de cada aplicación. Especialmente, en estos últimos años, con la llegada del estándar H.264/AVC, el número de propuestas ha aumentado significativemente en buena parte debido al creciente uso de las redes RTP/IP que han permitido la aparición de nuevas aplicaciones multimedia. Aunque la mayoría de estos algoritmos se han diseñado para aplicaciones de CBR, no han sido pocas las soluciones propuestas para aplicaciones que requieren una codificación de tipo VBR. No obstante, pese a que estas últimas aplicaciones son bastante populares en la sociedad actual, todavía existe una falta de soluciones prácticas de VBR para H.264/SVC, limitando así las posibilidades que una codificación de tipo escalable puede ofrecer. Concretamente, la restricción del HRD requerida para la transmisión de *bit streams* escalables aún no ha sido considerada adecuadamente, ya que, hasta donde sabemos, no hay una solución completa capaz de controlar el *buffer* de transmisión asociado a cada capa de dependencia.

Además, los algoritmos de RC para SVC propuestos hasta la fecha no aprovechan del todo las herramientas escalables, concretamente la escalabilidad temporal, ya que el cumplimiento de las restricciones del HRD únicamente es asegurado en el *sub-stream* correspondiente a la resolución temporal más alta de cada capa de dependencia. En consecuencia, el resto de *sub-streams* con otras resoluciones temporales correría un riesgo enorme de *overflows* o *underflows* en el llenado del *buffer*, lo cual se traduciría, en caso de producirse, en un empobrecimiento de la calidad visual durante la descodificación. Para poder entregar entonces varios *sub-streams* que cumplan con las restricciones del HRD, habría que incrementar el número de capas de dependencia. A modo ejemplo, si asumimos un servicio de transmisión de vídeo que ofrece la misma QoS a dos descodificadores objetivo con idéntica resolución espacial pero diferentes resoluciones temporales, dos capas CGS (base y realzada), una por frecuencia de cuadro de salida, tendrían que ser configuradas en el codificador si uno pretendiera utilizar alguno de los algoritmos de RC descritos en la literatura. Aunque en efecto esta configuración proporciona los dos *sub-streams* esperados, la

escalabilidad temporal está infrautilizada porque, de hecho, cada una de las frecuencias de cuadro de salida también incluye la frecuencia de cuadro menor, pudiéndo reducirse la configuración del codificador SVC a una sola capa de dependencia. En definitiva, una configuración típica para codificación SVC puede incurrir en capas de dependencia redundantes y, por lo tanto, en un incremento innecesario de la tasa y de la complejidad del codificador.

Objetivos

Esta tesis se centra en una solución de control de tasa de tipo VBR con restricciones de *buffer* para aplicaciones de codificación de vídeo H.264/SVC en tiempo real. El algoritmo de RC propuesto trata de proporcionar un rendimiento de codificación que represente el estado actual del arte en este contexto específico. Concretamente, el principal objetivo es que los *bit streams* escalables resultantes obedezcan a dos requisitos fundamentales en todos los niveles de escalabilidad: 1) cumplimiento con las restricciones del HRD para una descodificación limpia en recepción; 2) buena consistencia visual del video codificado. Este objetivo general para una codificación SVC controlada en tasa puede dividirse en dos objetivos particulares:

1. El algoritmo de VBR asume que planos consecutivos dentro de una misma escena a menudo exhiben grados de complejidad similares y, en consecuencia, deberían ser codificados usando valores de QP similares por el bien de la consistencia en la calidad visual. A diferencia de los algoritmos de CBR, los esquemas de RC diseñados para entornos de VBR normalmente modifican la QP sólo cuando es necesario, es decir, cuando la complejidad del vídeo varía de forma notable. Para prevenir entonces fluctuaciones de QP innecesarias, el controlador de VBR que proponemos tratará de estimar aquella variación incremental de QP (ΔQP) necesaria con respecto a un valor de referencia, centrándonos particularmente en un método efectivo para la estimación de ΔQP en lugar del valor absoluto de QP. En este contexto, el primer objetivo

de esta tesis es el diseño de un algoritmo novedoso para la estimación de ΔQP en H.264/SVC capaz de proporcionar un buen compromiso entre rendimiento de codificación y complejidad computacional.

2. Los algoritmos de RC para transmisión de vídeo escalable propuestos en la literatura son de tipo *in-layer single-buffer* (IL-SB), es decir, tienen en cuenta las restricciones del HRD únicamente en aquellos *sub-streams* de cada capa de dependencia correspondientes a la mayor resolución temporal. Esto significa que el número de capas de un codificador SVC está impuesto por el número de resoluciones espacio-temporales o calidades de reconstrucción objetivo. En cambio, el algoritmo de RC que describiremos en esta tesis será de tipo *in-layer multi-buffer* (IL-MB) para poder entregar, además, *sub-streams* que cumplan con las restricciones del HRD en más de una resolución temporal dentro de una capa de dependencia. Para ello, en lugar de emplear una configuración típica para el codificador H.264/SVC consistente en una capa de dependencia por resolución espacio-temporal [Wien and Schwarz, 2005], la idea básica que plantearemos en esta tesis es usar una configuración más compacta que conste de una capa de dependencia por resolución espacial, con el objetivo de que el algoritmo de RC controle simultáneamente varios *buffers* virtuales (salvo el que recoge el *bit stream* completo puede ser real) emplazados en una capa de dependencia (cada uno asociado a una resolución temporal distinta), mientras se maximiza la consistencia visual de los *sub-streams* correspondientes. En definitiva, el segundo objetivo de esta tesis es que el algoritmo de RC propuesto sea capaz de controlar, según las necesidades de la aplicación, uno o varios *buffers* de forma simultánea dentro de una capa de dependencia.

Organización

Esta tesis está compuesta por siete capítulos y dos apéndices, los cuales resumimos a continuación:

- El Capítulo 1 sirve para fijar el marco de trabajo específico en el que se desarrolla esta tesis, concretamente la problemática del RC para H.264/SVC. Contiene además la motivación que nos ha empujado a desarrollar este trabajo de investigación, así como sus objetivos finales. El capítulo concluye con una visión general del resto de capítulos que componen la tesis.

- En el Capítulo 2 describiremos las líneas generales del estándar de codificación H.264/SVC y, en particular, nos centraremos con más detalle en los tres tipos de escalabilidad: espacial, temporal y SNR. Posteriormente trataremeos ciertos aspectos de diseño de alto nivel del estándar H.264/SVC, tales como la posibilidad de introducir en un *bit stream* diferentes tipos de escalabilidad, la manera para conmutar de un *sub-stream* a otro, y la sintaxis empleada por el estándar para la lectura de las propiedades escalables incorporadas en el *bit stream*. Más adelante, hablaremos del algoritmo empleado por un descodificador H.264/SVC para extraer, partiendo de la QoS impuesta, el *sub-stream* más adecuado. Además, describiremos algunos de los escenarios de aplicación más comunes en SVC. Finalmente, comentaremos los perfiles soportados por el estándar H.264/SVC.

- El Capítulo 3 trata sobre la problemática del control de tasa en codificación de vídeo tanto *single-layer* como escalable. Posteriormente, explicaremos el esquema general de un algoritmo típico de RC y describiremos los pasos de operación empleados para la selección de QP. Asismismo, proporcionaremos un repaso extenso del estado del arte en RC tanto para aplicaciones de CBR como de VBR.

- El Capítulo 4 se centra en el algoritmo de VBR con restricciones de *buffer* propuesto para aplicaciones de codificación de vídeo H.264/SVC en tiempo real. Describiremos con todo detalle el esquema propuesto cuando un único *buffer* por capa de dependencia es controlado, como habitualmente sucede en los algoritmos de RC para SVC. En especial hablaremos del modelo propuesto para

la predicción de ΔQP con respecto a un valor de QP de referencia, el cual utiliza una regresión basada en procesos Gaussianos (GPs). Además, propondremos una manera para reducir, si cabe más, la complejidad computacional de nuestro algoritmo manteniendo similares prestaciones en terminos de calidad visual, control del *buffer*, y ajuste a la tasa objetivo. En el apartado de resultados, dos escenarios de aplicación en tiempo real se usarán para comparar las prestaciones del algoritmo propuesto con respecto a dos métodos de RC bien conocidos.

- El Capítulo 5 describe el nuevo enfoque de RC para el control IL-MB. Nos apoyaremos en el algoritmo de VBR explicado en el capítulo anterior para proponer las extensiones necesarias que garanticen un control simultáneo de varios *buffers* ubicados una capa específica de dependencia. En el apartado de resultados discutiremos las ventajas e inconvenientes del enfoque propuesto con respecto al tradicional método de RC de tipo IL-SB.

- En el Capítulo 6 describiremos la metodología utilizada para el diseño de GPs en un escenario como el que proponemos en la tesis, que consiste en la predicción de ΔQP. En primer lugar, explicaremos las razones que motivaron el uso del espacio de características de entrada a los GPs, así como también el uso de GPs para un problema de regresión. Finalmente, describiremos detalladamente las fases de generación del conjunto de muestras de entrenamiento, entrenamiento, y validación para el diseño de modelos de estimación de ΔQP basados en GPs.

- En el Capítulo 7 resumiremos las contribuciones de la tesis y proporcionaremos referencias a los papers asociados. Posteriormente, describiremos las principales conclusiones. Y por último, el capítulo concluye con una descripción breve de posibles líneas futuras de investigación.

- Para concluir la tesis, se incluyen dos apéndices con objeto de proporcionar información complementaria que, aunque no sea de vital importancia para la compresión global de la tesis, es necesaria para que los resultados de los experi-

mentos sean reproducibles. El primer apéndice explica en detalle el algoritmo de asignación de bits objetivo utilizado por el controlador de VBR descrito en el Capítulo 4. El segundo apéndice proporciona los valores de los párametros que constituyen los GPs finalmente seleccionados para estimar ΔQPs.

Contribuciones

A continuación se indican las principales contribuciones de esta tesis:

- A pesar de la relativa juventud de la extensión escalable del estándar de codificación de vídeo H.264/AVC, varios algoritmos de RC se han propuesto durante estos últimos años. Sin embargo, ninguno de ellos proporciona una solución flexible para aplicaciones de tipo VBR en tiempo real, como *streaming* de vídeo a través de redes inalámbricas o emisión de TV digital mediante redes IP. En el Capítulo 2 se propone un algoritmo de VBR para H.264/SVC con restricciones de *buffer*, el cual representa el estado del arte en RC para entornos de VBR por varios motivos:

  a) Buena consistencia visual y un control seguro del *buffer* de transmisión en todas las resoluciones espacio-temporales o calidades de reconstrucción objetivo.

  b) Configuración flexible del predictor de ΔQP para proporcionar una solución efectiva en un amplio espectro de aplicaciones de tipo VBR en tiempo-real.

  c) Bajo coste computacional.

- Pese a que el método de RC basado en la predicción de ΔQP con respecto a un valor de referencia no es novedoso, hemos aprovechado esta técnica para generar cuidadosamente un conjunto representativo de pares de *estado de codificación–ΔQP deseado* a ser modelado mediante algún interpolador fijo. En particular,

el interpolador utilizado para nuestros propósitos es el GP, el cual proporciona un excelente funcionamiento en tareas de regresión, cuya metodología de diseño es abordada en el Capítulo 6. Además, la función de coste que hemos diseñado para etiquetar las muestras de entrenamiento es suficientemente flexible para que pueda ser adaptada a otros requisitos y restricciones impuestos por la aplicación. De este modo, un nuevo etiquetado de datos de entrenamiento conlleva una nueva aplicación objetivo.

- Como ya se ha indicado con anterioridad, los esquemas de RC para SVC propuestos en la literatura tienen en consideración las restricciones del HRD solamente en el *sub-stream* correspondiente a la mayor resolución temporal de cada capa de dependencia. Dicho de otro modo, estos algoritmos no aprovechan del todo la escalabilidad temporal para proporcionar en una misma capa de dependencia *sub-streams* válidos con distintas resoluciones temporales. El enfoque general descrito en el Capítulo 5 basado en el control simultáneo de varios *buffers* dentro de una capa de dependencia supone un nuevo concepto de RC para SVC; y además un nuevo reto para futuras investigaciones, ya que el valor de QP debe elegirse adecuadamente para asegurar un buen control, no de un *buffer* sino de varios, mientras se garantiza una buena consistencia visual en los *sub-streams* implicados. Para un mismo número de resoluciones espacio-temporales o calidades de reconstrucción objetivo, este marco de operación permite reducir el número de capas de dependencia en comparación con las configuraciones de codificación SVC tradicionales, proporcionando así dos ventajas:

  a) Menor complejidad de codificación.

  b) Mayor calidad del vídeo codificado, ya que, para una misma tasa de bit objetivo, menos capas se codifican.

  Un último aspecto a destacar es que las ideas propuestas para el control IL-MB podría aplicarse a otros algoritmos de RC para SVC.

Líneas futuras

Finalmente, se plantean una serie de líneas de investigación futuras que permiten completar el trabajo iniciado en esta tesis:

- Con objeto de evaluar las prestaciones del algoritmo de VBR propuesto, se ha utilizado como QP inicial el valor obtenido por una codificación a QP constante para alcanzar una cierta tasa de bits objetivo. Pero no debemos olvidar que cualquier algoritmo de RC que opere en tiempo real necesita estimar ese valor inicial de QP al comienzo de la codificación, o bien después de un cambio de escena. Aunque algunas soluciones concevidas para H.264/SVC se han propuesto en la literatura, ninguna de ellas es lo bastante robusta para aplicaciones con restricciones de *buffer*. Una línea futura de investigación basada en la estimación del valor inicial de QP bajo restricciones de *buffer* ya ha sido iniciada [Sanz-Rodríguez and Díaz-de-María, 2011b], pero aún existe un margen de mejora para proporcionar predicciones de QP inicial más fiables.

- Un algoritmo de RC que adopte el enfoque para el control IL-MB necesita fijar una tasa objetivo para cada *sub-stream* implicado dentro de una capa de dependencia. Sin embargo, esas tasas dependen de las propiedades del vídeo. Por ejemplo, la tasa objetivo total para una sequencia con mucho detalle espacial y poco movimiento debería distribuirse entre las capas temporales para que los recursos de bit se asignen principalmente a los planos intra, los cuales pertenecen a la capa temporal más baja. Por el contrario, para una secuencia con detalle espacial moderado y mucho movimiento, el reparto de bits debería efectuarse de un modo más equitativo. En definitiva, para una aplicación en tiempo real necesitamos conocer a priori las propiedades del vídeo para distribuir adecuadamente la tasa objetivo total entre las capas temporales. Un método efectivo para estimar esas tasas objetivo es otra de las líneas futuras de trabajo planteadas.

- Probablemente el *streaming* de vídeo en dos pasadas sería un escenario de

aplicación interesante para un enfoque de codificación IL-MB, ya que podríamos aprovechar la primera pasada para encontrar esa distribución óptima de la tasa objetivo total entre las capas temporales.

- Asimismo, en esta tesis se asume, por simplicidad, que todos los receptores H.264/SVC objetivo comparten el mismo llenado de *buffer* objetivo y tamaño de *buffer* (en segundos), pero esta suposición no siempre es cierta debido al carácter heterogéneo de los dispositivos multimedia. El comportamiento del algoritmo de VBR cuando se usan diferentes parámetros del RC para cada resolución espacio-temporal podría ser también estudiado.

Agradecimientos

A lo largo de toda mi carrera académica he ido alcanzando objetivos que no imaginaba, hace quince años cuando comencé mis estudios universitarios, que podría lograr. En 2001 finalicé los estudios de Ingeniería Técnica de Telecomunicación, especialidad Sonido e Imagen. En 2005 obtuve el título de Ingeniero de Telecomunicación. Ahora, en 2011, esta tesis doctoral supone el cierre de una etapa más, no sólo como ingeniero, sino también como investigador novel. Además, este trabajo ha sido especialmente importante para mí por dos razones. El primero porque me ha motivado a continuar, si cabe más, ligado a la Universidad. El segundo porque esta tesis ha significado (espero) un pequeño, aunque minúsculo, avance en un área específica de investigación. Sin embargo, nuevos retos aún están por alcanzar y, en consecuencia, muchas alegrías y no pocas decepciones. Por ello, todas aquellas personas que han estado siempre a mi lado en esos momentos y que continuarán estándolo tienen reservado un gran pedazo de mi corazón y, sin duda alguna, se han ganado el cielo.

Antes de nada, este trabajo no habría sido posible sin la supervisión y el apoyo moral de mi Director de tesis, el Dr. Fernando Díaz de María. También agradezco a él la confianza depositada en mí durante estos años, tanto en proyectos de investigación como en responsabilidades docentes.

Quisiera proseguir mis agradecimientos recordando a todos mis compañeros del Departamento de Teoría de la Señal y Comunicaciones y, en particular, a los integrantes del Grupo de Procesado Multimedia incluyendo a los que en su día formaron también parte. Me gustaría mencionar en estas líneas especialmente a Chelus Molinero, Eduardo Martínez, Iván González, Luis Azpicueta, Manuel de Frutos, Mario de Prado, Rocío Arroyo, Rosa Mª Barrio (un beso enorme a tí para que no te disgustes tanto conmigo), Rubén Solera y Sara Pino, ya que con ellos he pasado unos ratos bastante agradables. A ti, Rubén, porque el destino ha querido que compartiéramos unas semanas de agobios, aunque también de buen rollo, antes de depositar la tesis. Y recuerda la frase que ha marcado esos momentos: "¡Que no llegamos!" . También

quisiera agradecer a Vanessa Gómez y Miguel Lázaro la inestimable ayuda que me han ofrecido durante ciertas fases de la tesis. A ti, Miguel, porque una parte de mis publicaciones también te pertenece.

I would also like to thank Prof. Moncef Gabbouj for welcoming me at the Nokia Research Center during my four-month stay at Tampere University of Technology and giving me the opportunity of collaborating with Dr. Mehdi Rezaei and other researchers. I am really grateful to Dr. Mehdi Rezaei since likely this thesis would have never started without his help and knowledge.

Fuera del ámbito laboral, a muchas personas quiero agradecer los ánimos que me han insuflado a lo largo de este tiempo. A mis amigos de Alcobendas y San Sebastian de los Reyes. A mis amigos de la Agencia Efe. A mis amigos de Herencia. A mis amigos de Tampere. A mi amigo Rubén López. También a Pepa Carracedo por haberme aguantado tanto y por decirme lo muy orgullosa que está de mí. Y sobre todo, con todo el cariño, doy las gracias a mi familia por estar siempre a mi lado.

Me siento tan afortunado, a veces inmerecidamente, de estar rodeado de gente tan valiosa que, de hecho, sois vosotros el mayor logro de mi vida.

A mis padres, hermanos y amigos

Contents

List of Figures xxix

List of Tables xxxiii

List of Abbreviations xxxv

List of Symbols xxxvii

1 Introduction 1

1.1 Motivation 1

1.2 Objectives 4

1.3 Overview of the Rest of the Thesis 5

2 Scalable Video Coding (SVC) 9

2.1 Single-Layer Video Coding Standards 10

2.2 Overview of the H.264/SVC Standard 11

2.3 Types of Scalability 12

2.3.1 Temporal Scalability 13

2.3.2 Spatial Scalability 14

2.3.3 Quality Scalability 17

2.4 SVC High-Level Design 21

2.4.1 Combined Scalability 22

2.4.2 Bit Stream Switching 22

2.4.3 System Interface 23
2.5 Bit Stream Extraction 24
2.6 Application Areas 25
2.7 Profiles 26

3 Rate Control (RC) for Video Coding 29
3.1 Requirements and Constraints 30
3.2 Optimal RC Solutions 31
3.3 Model-Based RC Solutions 33
3.4 Operation Steps in a RC Algorithm 37
3.4.1 Bit Allocation 37
3.4.2 Quantization Parameter (QP) Estimation 38
3.5 Rate-Distortion (R-D) Modeling for Video Compression 39
3.5.1 Operational R-D Functions 39
3.5.2 Model-Based R-D Functions 41
3.6 Constant Bit Rate (CBR) Coding and Variable Bit Rate (VBR) Coding 45
3.6.1 CBR Control Algorithms 46
3.6.2 VBR Control Algorithms 49

4 VBR Controller for H.264/SVC 51
4.1 System Overview 52
4.2 RC Stages 57
4.2.1 Parameter Updating 57
4.2.2 Gaussian Process (GP)-Based QP Increment Estimation . . . 60
4.3 Implementation Considerations 64
4.4 Experiments and Results 64
4.4.1 Description of the Application Scenarios 65
4.4.2 Experimental Results and Discussion 68
4.5 Summary and Conclusions 76

5 In-Layer Multi-Buffer Framework for SVC 79
5.1 System Overview 80
5.2 RC Stages 83
5.2.1 Parameter Updating 83
5.2.2 Buffer Modeling 84
5.2.3 GP-Based QP Increment Estimation 87
5.3 Experiments and Results 88
5.3.1 Description of the SVC Encoder and RC Configurations . . . 89
5.3.2 Experimental Results and Discussion 91
5.4 Summary and Conclusions 101

6 GP-Based QP Increment Estimation Design 103
6.1 Input Vector Selection 104
6.2 Why GPs for Regression? 105
6.3 Generation of the Training Data Set 106
6.3.1 Getting Initial Average Complexities 108
6.3.2 Generating Training Pairs 108
6.4 Training 111
6.5 Validation 113
6.5.1 GP Parameter Selection 113
6.5.2 Post-Processing Stage Configuration 117
6.6 Summary and Conclusions 119

7 Conclusions and Further Work 123
7.1 A Summary of Contributions 123
7.2 Conclusions 125
7.3 Future Research Lines 126

A Estimation of the Access Unit Target Bits 129

B GP Parameters 133

Bibliography 137

List of Figures

2.1 Video coding with hierarchical B pictures. The numbers at the bottom indicate the picture coding order, while $t = k$ specifies the temporal layer with k representing the corresponding temporal layer identifier. 13

2.2 Video coding using a hierarchical prediction structure with a zero structural encoding/decoding delay. The numbers at the bottom indicate the picture coding order, while $t = k$ specifies the temporal layer with k representing the corresponding temporal layer identifier. . . . 14

2.3 Layered coding approach with inter-layer prediction for enabling spatial scalable coding. The shadowed area indicates an AU and the number inside it specifies the coding order of the corresponding pictures. 15

2.4 Generic block diagram of the CGS coding process with three dependency layers. The coded video SNR is increased from one dependency layer to the next. 18

2.5 Generic block diagram of the MGS coding process with three quality refinements. The coded video SNR is increased for one quality layer to the next. 19

2.6 Key picture concept for hierarchical prediction structures. Between two K pictures, the enhancement representation is used for prediction. However, for the K pictures, only the base representation is used. . . 20

2.7 Example of DCT coefficient partition for MGS coding. 21

2.8 Example of an H.264/SVC encoder structure with two dependency layers. For the lowest spatial resolution, two quality refinements are shown, while for the highest spatial resolution, three quality refinements are included. 23

2.9 H.264/SVC video transmission through wireless broadcast networks. . 26

2.10 Unequal erasure protection and H.264/SVC quality layers. 27

3.1 Block diagram of an RC algorithm for single-layer video coding. . . . 34

3.2 Block diagram of an RC algorithm for SVC. A particular case with two dependency layers is shown. 36

3.3 Illustration of an ORD curve and its associated model-based R-D curve. 40

3.4 Operating regions in the PSNR-rate space for CBR coding (left) and VBR coding (right). 46

4.1 Block diagram of the proposed H.264/SVC RC scheme for two dependency layers ($D = 2$). 53

4.2 Block diagram of the rate controller $RC^{(d)}$ for a specific dependency layer d. 55

4.3 Output of the (a) K and (b) NK GPs for $nTF = 0.5$ and $BD = 3$. . . 62

4.4 Sample outputs of the (a) K and (b) NK GPs for $nTF = 0.5$ and several values of BD; and sample outputs of the (c) K and (d) NK GPs for $BD = 3$ and several values of nTF. For the sake of clarity, only a cut of the three-dimensional surface for $nAU^{(d)} = 1$ is drawn. . 63

4.5 Encoder buffer level, PSNR and QP time evolutions corresponding to (a) the spatial base layer ($d = 0$) and (b) the third enhancement layer ($d = 3$) from *The Lord of the Rings*. High-quality plots corresponding to every spatial/CGS layer are available on-line in [Sanz-Rodríguez, 2011]. 73

4.6 Encoder buffer level, PSNR and QP time evolutions corresponding to (a) the spatial base layer ($d = 0$) and (b) the third enhancement layer ($d = 3$) from *Stockholm*. High-quality plots corresponding to every spatial/CGS layer are available on-line in [Sanz-Rodríguez, 2011]. . . 74

5.1 Block diagram of the proposed H.264/SVC RC scheme for IL-MB control. Only the spatial base layer is depicted for the sake of clarity. 81

5.2 Block diagram of the MB-based rate controller module $RC^{(d)}$ for a specific dependency layer d. 84

5.3 Output of the (a) K-MB and (b) NK-MB GPs for $nTF = 0.5$ and $BD = 3$. 88

5.4 Encoder buffer level, PSNR and QP time evolutions corresponding to the spatio-temporal resolutions (a) QCIF@6.25 Hz and (b) QCIF@12.5 Hz for *Bus*. High-quality plots are available on-line in [Sanz-Rodríguez, 2011]. 98

5.5 Encoder buffer level, PSNR and QP time evolutions corresponding to the spatio-temporal resolutions (a) QCIF@6.25 Hz and (b) QCIF@12.5 Hz for *The Lord of the Rings*. High-quality plots are available on-line in [Sanz-Rodríguez, 2011]. 98

6.1 SB training data distributions for K pictures (black) and NK pictures (gray), with $nTF = 0.5$ and $BD = 3$. (a) Weight vector $\boldsymbol{\lambda}_a$. (b) Weight vector $\boldsymbol{\lambda}_b$. High-quality plots are available on-line in [Sanz-Rodríguez, 2011]. 112

6.2 SB training data distributions for the dependency base layer (black) and the dependency enhancement layers (gray), with $nTF = 0.5$ and $BD = 3$. (a) K pictures and weight vector $\boldsymbol{\lambda}_a$. (b) NK pictures and weight vector $\boldsymbol{\lambda}_b$. High-quality plots are available on-line in [Sanz-Rodríguez, 2011]. 112

6.3 Encoder buffer occupancy, PSNR, and QP time evolutions with post-processing (black) and without post-processing (gray) for the sequence *Container*. (a) Spatio-temporal resolution: QCIF@12.5 Hz ($d = 0, t = 2$). (b) Spatio-temporal resolution: QCIF@25 Hz ($d = 0, t = 3$). High-quality plots are available on-line in [Sanz-Rodríguez, 2011]. 119

6.4 Encoder buffer occupancy, PSNR, and QP time evolutions with post-processing (black) and without post-processing (gray) for the sequence *Ice Age*. (a) Spatio-temporal resolution: QCIF@12.5 Hz ($d = 0, t = 2$). (b) Spatio-temporal resolution: QCIF@25 Hz ($d = 0, t = 3$). High-quality plots are available on-line in [Sanz-Rodríguez, 2011]. . . 120

List of Tables

4.1 Average results achieved by both the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for the mobile live streaming scenario. Incremental results are given with respect to CQP encoding. 69

4.2 Average results achieved by both the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for the IPTV broadcast scenario. Incremental results are given with respect to CQP encoding. 70

4.3 Performance comparison between the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for a specific non-stationary complexity video sequence, *The Lord of the Rings*. The results achieved by CQP encoding have also been included for reference. 71

4.4 Performance comparison between the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for a specific stationary complexity video sequence, *Stockholm*. The results achieved by CQP encoding have also been included for reference. 72

4.5 CPU time comparison between the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for the mobile live streaming scenario using an Intel Core2 Duo CPU E8400@3.0 GHz. 76

4.6 CPU time comparison between the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for the IPTV broadcast scenario using an Intel Core2 Duo CPU E8400@3.0 GHz. 76

5.1 Target bit rates assigned to each spatio-temporal layer of the compared RC-SVC encoders. 92
5.2 Average results achieved by the SB-BC, the SB-CC, and the proposed MB-CC VBR controllers. Incremental results are given with respect to CQP-CC encoding. 93
5.3 Performance comparison among the SB-BC, the SB-CC, and the proposed MB-CC VBR controllers, for a specific stationary complexity video sequence, *Bus*. The results achieved by CQP-CC encoding have also been included for reference. 94
5.4 Performance comparison among the SB-BC, the SB-CC, and the proposed MB-CC VBR controllers, for a specific non-stationary complexity video sequence, *The Lord of the Rings*. The results achieved by CQP-CC encoding have also been included for reference. 96
5.5 Average results achieved by the proposed MB-CC VBR controller for different target bit rate deviations at layers $(0, 1)$ and $(2, 2)$. Incremental results are given with respect to those achieved by CQP-CC encoding. 100

6.1 Weight vectors, denoted as $\boldsymbol{\lambda}_a$, $\boldsymbol{\lambda}_b$, $\boldsymbol{\lambda}_c$, for the cost function in Equation (6.2) used for training data labeling. 111
6.2 Summary of the training and validation processes for K-SB and NK-SB GP parameter selection. The best GP pair is highlighted in gray. 114

List of Abbreviations

AU	Access Unit
AVC	Advanced Video Coding
CBR	Constant Bit Rate
CGS	Coarse Grain Scalability
CIF	Common Intermediate Format
GoP	Group of Pictures
GP	Gaussian Process
HDTV	High-Definition Television
HRD	Hypothetical Reference Decoder
IL-MB	In-Layer Multi-Buffer
IL-SB	In-Layer Single-Buffer
K	Key
MGS	Medium Grain Scalability
NK	Non Key
QCIF	Quarter Common Intermediate Format

QoS	Quality of Service
QP	Quantization Parameter
RC	Rate Control
R-D	Rate-Distortion
SDTV	Standard-Definition Television
SNR	Signal-to-Noise Ratio
SVC	Scalable Video Coding
VBR	Variable Bit Rate
4CIF	4 times Common Intermediate Format

List of Symbols

$\Delta QP^{(d)}$	QP increment for the layer d
$\mathbf{nAU}^{(d)}$	Set of normalized AU output bits of the layers $(d, t_{min}^{(d)})$ to $(d, t_{max}^{(d)})$
$\mathbf{nV}^{(d)}$	Set of normalized buffer fullness of the layers $(d, t_{min}^{(d)})$ to $(d, t_{max}^{(d)})$
$\mathbf{QP}^{(d)}$	Set of previous QPs of the layers $(d, t_{min}^{(d)})$ to $(d, t_{max}^{(d)})$
$\mathbf{V}^{(0,t)}$	Set of involved buffer fullness after encoding a picture of the layer t
$\mathbf{w}, w_0, \{\sigma, \mathbf{bC}\}$	Weights, bias and hyperparameters of the GP
$\mathbf{X}^{(d)}$	Input vector to the GP
$\overline{C}_{MOT}^{(d,t)}$	Average motion complexity of the layer (d, t)
$\overline{C}_{TEX}^{(d,t)}$	Average texture complexity of the layer (d, t)
$AU^{(d,t)}$	AU output bits of the layer (d, t)
BD	Buffer size in seconds
$BS^{(d,k)}$	Buffer size in bits associated with the sub-stream (d, k)
D	Number of dependency layers
d	Dependency layer identifier
$f_{out}^{(d,k)}$	Output frame rate of the sub-stream (d, k)

$G^{(d,t,k)}$	AU target bits of the layer (d,t) to satisfy $R^{(d,k)}$
H	Gaussian-type function
k	Index that takes values from $t_{min}^{(d)}$ to $t_{max}^{(d)}$
M	Number of Gaussian-type functions of the GP
$nAU^{(d)}$	A normalized version of the AU output bits of the layer d
nTF	Normalized target buffer fullness with respect to the buffer size
$nV^{(d)}$	A normalized version of the buffer fullness of the layer d
q	MGS layer identifier
$Q^{(d)}$	Number of MGS refinements in the layer d
$QP_j^{(d)}$	QP value for the jth picture of the layer d
$R^{(d,k)}$	Target bit rate for the sub-stream (d,k)
$RC^{(d)}$	Rate control module of the layer d
t	Temporal layer identifier
$T^{(d)}$	Number of temporal layers in the layer d
$t_{max}^{(d)}$	Maximum temporal layer identifier of the layer d
$t_{min}^{(d)}$	Minimum involved temporal layer identifier of the layer d
$V^{(d,k)}$	Buffer fullness associated with the sub-stream (d,k)

Chapter 1

Introduction

This thesis aims to provide rate control (RC) solutions for scalable video coding (SVC). In this chapter we describe the motivation, and the objectives of the thesis. Finally, an overview of the rest of the document is given.

1.1 Motivation

Nowadays, modern real-time transport protocol/Internet protocol (RTP/IP)-based transmission systems such as Internet and wireless networks are becoming more and more popular for video communications. For this kind of channels, SVC is able to provide proper adaptation to varying channel conditions as well as to receiving devices with heterogeneous display resolutions and computational capabilities. SVC allows for extracting, from a high-quality bit stream, either one or a subset of bit streams with lower spatio-temporal resolutions or reduced reconstruction qualities that can be decoded by any target receiver.

Almost every video coding standard, such as MPEG-2, H.263, MPEG-4 Visual, and H.264/advanced video coding (AVC), have incorporated tools to support the most common scalable coding modes: *spatial*, *temporal* and *quality* (or *signal-to-noise ratio*, SNR) scalability. Spatial scalability and temporal scalability provide

sub-streams that represent the video source content with either a reduced picture size (spatial resolution) or frame rate (temporal resolution), respectively. Regarding quality scalability, the sub-stream provides the same spatio-temporal resolution as that of the complete bit stream, but with lower reconstruction fidelity.

The scalable extension of H.264/AVC, named H.264/SVC, has recently been standardized [Schwarz et al., 2007, Wien et al., 2007b]. It is able to provide both coding efficiency and decoding complexity similar to those achieved using a non-scalable coding. In H.264/SVC spatial scalability is achieved by means of a layered coding approach in which each layer is used to encode a different picture size of the video source. The base layer generates an H.264/AVC compatible bit stream for the lowest spatial resolution, while larger picture sizes are encoded by the enhancement layers.

Each spatial layer is capable of supporting temporal scalability by using hierarchical temporal prediction structures. The temporal base layer provides an encoded version with the lowest temporal resolution. The next temporal layer adds the required pictures to double the output frame rate, and so on.

For SNR scalability, a layered coding approach is also used to encode different reconstruction quality levels with the same spatio-temporal resolution. In particular, the H.264/SVC standard defines two types of SNR scalable coding: *coarse grain scalability* (CGS) and *medium grain scalability* (MGS). The first is a special case of spatial scalability with identical picture sizes. It is worth mentioning that a spatial/CGS layer is also referred to as *dependency layer*. The second uses the multi-layer coding approach within a dependency layer in order to provide a finer bit rate granularity in the rate-distortion (R-D) space.

The different types of scalability can also be combined so that a multitude of representations with different spatio-temporal resolutions and quality levels can be supported within a single scalable bit stream.

For a proper multimedia content delivery in video communications, the RC algorithm is actually a key subsystem in both scalable (multi-layer) and non-scalable (single-layer) video coding systems. Typically, an RC scheme works in two steps:

1. A bit budget is allocated to a video segment according to the video content, the target bit rate, and either the buffer constraints for transmission or the budget constraint for digital storage.

2. A quantization parameter (QP) value is assigned to the video segment in order to satisfy the aforementioned buffer and/or budget constraints, while minimizing the reconstructed video distortion. To this end, an R-D model of encoded video source is typically used.

The buffer constraints are imposed by the *hypothetical reference decoder* (HRD) [Ribas-Corbera et al., 2003] that is conceptually connected to the output of an encoder and consists of a decoder buffer, a decoder, and a display unit. The decoder buffer model is characterized by three values: target bit rate, buffer size and initial decoder buffer fullness. Under these constraints, both the encoder and the RC algorithm should create a bit stream so that the decoder buffer does not incur in overflow or underflow. Additionally, the bit allocation method for digital storage must be aware of the maximum allowed storage capacity.

In the case of SVC, it is also worth noticing that the RC algorithm actually consists of a set of rate controllers, each one located at each spatial or SNR layer, to provide a set of HRD and/or budget-compliant scalable sub-streams, each one for a certain target bit rate suitable for a target decoding terminal managing a particular spatio-temporal resolution or computational capability.

The RC problem has been extensively studied for both single-layer video coding and SVC. According to the target application, two kinds of RC methods have been proposed: constant bit rate (CBR) and variable bit rate (VBR) control algorithms. On the one hand, the CBR controllers, commonly used for real-time video conference, pursue a short-term target bit rate adjustment to guarantee a low buffer delay. On the other hand, the VBR controllers, typically used for video streaming or digital storage, manage a long-term target bit rate adaptation at the expense of a longer buffer delay to maintain a high visual quality consistency [Lakshman et al., 1998, Ortega, 2000].

Numerous CBR and VBR control algorithms have been proposed for both scalable and non-scalable video coding. In the case of the H.264/SVC standard, a lot of contributions to improve the solution to the RC problem have been recently reported. In most of them, the ideas already developed for H.264/AVC have been adapted to the corresponding scalable extension, especially regarding the CBR control problem. Nevertheless, there is still a lack of practical solutions for VBR environments. Specifically, the HRD compliance required to properly deliver the scalable video content has not been properly considered yet, since, as far as we know, there is no any complete solution capable of managing the buffer control at every dependency layer, while providing good visual quality consistency for the corresponding sub-streams. Furthermore, temporal scalability is not totally exploited by the RC algorithms for SVC because the HRD requirement is only satisfied for the highest frame rate sub-stream of every dependency layer, so the correct delivery of lower temporal resolutions is not guaranteed.

1.2 Objectives

This thesis focuses on a VBR control solution for real-time H.264/SVC applications with buffer constraints. The proposed RC algorithm aims to provide state-of-the-art performance in this specific context. In particular, the main objective is that the resulting scalable bit streams satisfy two essential requirements at all the considered scalability levels: HRD compliance and quality consistency. This general objective for rate-controlled SVC encoding can be divided into two particular objectives:

1. In VBR applications, the RC algorithm relies on the fact that consecutive pictures within the same scene often exhibit similar degrees of complexity and, consequently, should be encoded using similar QP values for the sake of quality consistency. In order to prevent unnecessary QP fluctuations, the main objective of the VBR controller is to allow for just an incremental variation of QP with respect to a reference value, focusing on the design of an effective method

for the estimation of this QP variation and not the QP value itself. In this context, the first objective of this thesis is to design a novel method for the QP increment estimation in H.264/SVC that provides a good trade-off between coding performance and computational complexity.

2. The RC algorithms proposed in the literature for SVC only take into account the HRD compliance for the highest frame rate sub-stream at every dependency layer; therefore, the number of layers is the same as the number of target spatio-temporal resolutions and quality levels. However, the VBR controller could be also capable of delivering HRD-compliant sub-streams with different temporal resolutions within a particular dependency layer. Thus, the second objective of this thesis is to design a compact RC configuration that, instead of using a typical SVC encoder configuration consisting of a dependency layer per spatio-temporal resolution [Wien and Schwarz, 2005], is able to manage more than one HRD-compliant temporal resolution per dependency layer. In other words, the aim of the RC algorithm at a given dependency layer is to properly and simultaneously control several buffers, one per target temporal resolution, while maximizing the quality consistency of the corresponding sub-streams.

1.3 Overview of the Rest of the Thesis

The thesis is divided into seven chapters and two appendix. Once the motivation and the objectives have been described, the following two chapters aim to give an overview of SVC, in particular the scalable extension of the H.264/AVC standard, and a detailed description of the RC problem in video coding, respectively. Chapter 4 describes in detail the first contribution of this thesis, while Chapter 5 focuses on the second contribution. As described in Chapters 4 and 5, the proposed solution for the QP increment estimation relies on Gaussian processes (GP)-based regressors, and Chapter 6 has been devoted to explain the general methodology proposed for designing this type of regression for the problem at hand. Chapter 7 summarizes the

contributions and the conclusions, and outlines the future lines of research. A more detailed overview of the remaining chapters follows.

- Chapter 2 describes the basic concepts of the H.264/SVC standard for scalable video content delivery. In particular, the three supported scalable coding modes are described. Then, some aspects of SVC high-level design are summarized; specifically: the possibility of including different types of scalable coding modes in the bit stream, the mechanism used for bit stream switching, and the system interface. The basic ideas about the bit stream extraction mechanism used to find the desired sub-stream are also given. Moreover, we discuss some representative applications that can benefit from SVC. Finally, profiles supported by H.264/SVC are outlined.

- Chapter 3 discusses the RC problem in both single-layer video coding and SVC. Subsequently, the general approach to the RC problem is described, as well as the basic operating steps followed to compute a suitable QP value. The R-D models typically used in RC for QP computation are also described. Finally, the state of the art in RC concerning CBR and VBR control methods is also summarized, emphasizing the advantages and disadvantages of the most relevant proposals for H.264/SVC.

- Chapter 4 focuses on the proposed VBR control algorithm for real-time H.264/SVC with buffer constraints. A brief overview of the proposed method is given first. Then, each stage of the rate controller located at a specific dependency layer is described in detail, especially the GP regression method used to predict the QP increment with respect to the reference value. Some issues regarding the complexity of the algorithm are also discussed. The chapter finishes with some experimental results that demonstrate the excellent performance of the proposed algorithm in terms of visual quality, buffer control, and computational complexity.

- Chapter 5 describes the novel framework that aims to deliver HRD-compliant,

consistent-quality sub-streams with different temporal resolutions within a specific dependency layer. To this end, we will lean on the VBR controller described in the previous chapter to explain those extensions of that algorithm required for a simultaneous control of several buffers within a dependency layer. Some experiments are conducted to show the advantages and disadvantages of the proposed approach with respect to a traditional RC operation in SVC.

- Chapter 6 provides the details of the GP design process for QP increment estimation. Specifically, we discuss the motivation for the use of certain parameters as inputs to the GP regression model, as well as some properties of GPs that makes them suitable for regression. Finally, training data set generation, training, and validation processes are described.

- In Chapter 7 the contributions of the thesis are reviewed, including references to the associated papers. Then, the main conclusions are summarized and discussed. And finally, some interesting research lines are briefly described.

- To conclude, two appendices are included in order to provide some complementary information that, although it is not essential for understanding the contributions of the thesis, it is necessary for the sake of reproducibility of the results. The first appendix explains in detail the bit allocation algorithm used by the VBR controller described in Chapter 4, and the second provides the parameters for every GP regression model used in the thesis.

Chapter 2

Scalable Video Coding (SVC)

During the last years video applications have grown in popularity because of the increasing advances in video coding technology and standardization, network infrastructure, data storage, and memory and computational resources of multimedia devices. The most popular video application areas include multimedia messaging, video conference, Internet video streaming, standard-definition television (SDTV) and high-definition television (HDTV) broadcast, as well as record/playback in optical storage media such as Digital Versatile Disk (DVD) and Blu-ray Disk (BD).

For enabling these applications, either traditional or modern video transmission systems can be employed. In traditional video transmission systems, typically used for broadcast services (over satellite, cable and terrestrial channels) or DVD/BD storage, the video signal is characterized by a fixed spatio-temporal format (SDTV@25 Hz, HDTV@50 Hz, etc.). Therefore, the video source is not available at any other different spatio-temporal resolutions. These channels can be in one of two possible connection states: it works or it does not [Schwarz et al., 2007].

However, modern real-time RTP/IP-based transmission systems such as Internet and wireless networks are characterized by a wide range of quality of services (QoS), which are derived from the varying network bandwidth capabilities and the variety of decoding terminals, ranging from mobile phones with limited display resolu-

tion and/or computational capability to personal computers (PCs) or HDTV set-top boxes. It is also worth mentioning that the QoS associated with a spatio-temporal resolution is measured in terms of target bit rate (or quality) and some networking parameters such as the absolute delay [Lakshman et al., 1998].

Within this technological framework, SVC provides an appealing solution that matches to some extent the inherent characteristics of these RTP/IP-based transmission systems. Specifically, SVC allows for QoS adaptation to variable network conditions or needs or preferences of end user, as well as video content delivery to a variety of decoding terminals with heterogeneous display resolutions and computational capabilities, by means of a set of scalability features.

A bit stream is called *scalable* when parts of it can be ignored while the resulting sub-stream is also a valid bit stream, either for some target decoder or simply at a reduced target bit rate. That sub-stream represents a source content with a reconstruction quality lower than that of the complete scalable bit stream, but higher when considering any other sub-stream inside it. Bit streams that are not scalable are referred to as *non-scalable* or *single-layer* bit streams.

The rest of the chapter is organized as follows. In Section 2.1 a brief overview of the single-layer video coding standards is given, emphasizing the H.264/AVC standard. Sections 2.2 and 2.3 focus on the basic concepts of the scalable extension of H.264/AVC and its supported scalability modes. Section 2.4 describes some key aspects of SVC high-level design. In Section 2.5 a generic algorithm for scalable bit stream extraction is described. Section 2.6 briefly describes some application scenarios of SVC. Finally, Section 2.7 outlines the profiles supported by H.264/SVC.

2.1 Single-Layer Video Coding Standards

Single-layer video compression standards have played a paramount role in the success of digital video applications. MPEG-2 [ISO/IEC, 1994], H.263 [ITU-T, 1995], MPEG-4 Visual [ISO/IEC, 1999] and H.264/AVC [JVT, 2003] are some of the most

widely used video coding standards. Most of them are based on the classic *block-based hybrid* scheme that consists of a motion estimation and compensation stage for temporal redundancy reduction, a discrete cosine transform (DCT) stage for decorrelation and compaction of the prediction error, and a quantization stage followed by an entropy coding stage for statistical redundancy reduction of quantified DCT coefficients. A picture can be *intra* (I) coded, *predictive* (P) coded or *bipredictive* (B) coded, where the basic coding unit is the *macroblock* (MBk), defined as a fixed-size block of 16×16 pixels. Additionally, between a picture and a MBk, an intermediate coding unit so-called *slice* is defined as a specific set of connected MBks that is encoded without taking into account any other parts (slices) in the same picture. In I pictures (or slices), each MBk is encoded without referring to other reconstructed pictures in the video sequence. In P pictures, a MBk can be either intra or inter coded; in the last case by means of a *block-matching-based* algorithm for motion estimation between the original MBk and some reconstructed MBk belonging to an already coded picture. Finally, in B pictures, a MBk can be predicted from two already coded pictures.

Although all of the mentioned video coding standards provide similar coding tools, H.264/AVC achieves the highest coding efficiency. Its main improvements include I-frame prediction modes for spatial redundancy reduction, variable block-size motion compensation with small block sizes, quarter-pixel motion vectors, multiple reference pictures for motion compensation, in-loop deblocking filter, R-D optimization for mode decision, and context-adaptive binary arithmetic coding (CABAC) that achieves bit rate reductions between 5% and 15% when compared to context-adaptive variable-length coding (CAVLC) [Wiegand et al., 2003].

2.2 Overview of the H.264/SVC Standard

Several video coding standards prior to H.264/AVC, such as MPEG-2, H.263 and MPEG-4 Visual, have incorporated scalable profiles for supporting the most impor-

tant scalable coding modes, i.e., temporal, spatial and quality scalability. Nevertheless, most of these extensions have been hardly ever used in real applications. The following factors have influenced that limited deployment: on the one hand, the unsuitability of traditional video transmission systems and the lack of an actual diversity of decoding devices; on the other hand, the loss in coding efficiency and the increase in decoding complexity when compared to non-scalable video coding systems. Thus, alternative methods such as *simulcast* and *transcoding* have been preferred to scalable profiles for transmission and management of video content. The simulcast method incorporates analogous functionalities as those of SVC, though at the expense of a noticeable increase in bit rate. The transcoding method is typically used for bit stream adaptation to certain spatio-temporal format or reduced quality version, but each target spatio-temporal or quality format requires a specific transcoding.

Fortunately, the recently standardized scalable extension of the H.264/AVC standard named H.264/SVC [Schwarz et al., 2007, Wien et al., 2007b], is able to provide both coding efficiency and decoding complexity not far from those achieved by single-layer H.264/AVC coding in similar conditions. Consequently, H.264/SVC have attracted the interest of video coding researchers for the last years.

2.3 Types of Scalability

As already pointed out, the common SVC modes are temporal, spatial and quality scalability. The two firsts provide subsets of the complete bit stream that represent the source content at a reduced frame rate for temporal scalability, or a reduced picture size for spatial scalability. With respect to quality scalability, the sub-stream provides the same spatio-temporal resolution as that of the complete bit stream but lower reconstruction fidelity or SNR. The different types of scalability can also be combined providing sets of sub-streams with different spatio-temporal resolutions and SNR versions (or bit rates) within the complete scalable bit stream.

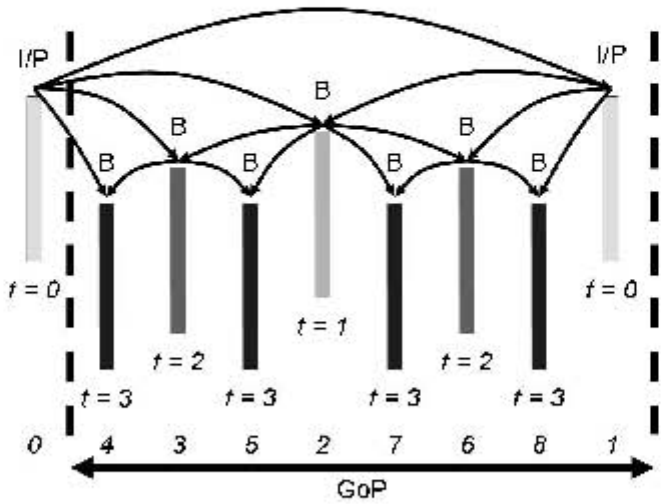


Figure 2.1: Video coding with hierarchical B pictures. The numbers at the bottom indicate the picture coding order, while $t = k$ specifies the temporal layer with k representing the corresponding temporal layer identifier.

These scalability modes are described in more detail in the following subsections.

2.3.1 Temporal Scalability

A bit stream provides temporal scalability when the encoded pictures can be divided into a temporal base layer and one or more enhancement layers. Each layer is identified by a temporal layer identifier t, which is equal to 0 for the base layer, 1 for the next temporal layer, and so on until reaching the highest enhancement layer. For a given identifier $t = k$, a sub-stream with reduced temporal resolution is extracted by discarding all pictures belonging to higher temporal layers ($t > k$).

Temporal scalability is supported by using hierarchical prediction structures, which go from these efficient ones using hierarchical B pictures (see Figure 2.1) to those with zero structural delay (see Figure 2.2) but lower coding efficiency. The pictures of the temporal base layer can only use previous pictures of the same layer as references. The pictures of a temporal enhancement layer can be bidirectionally

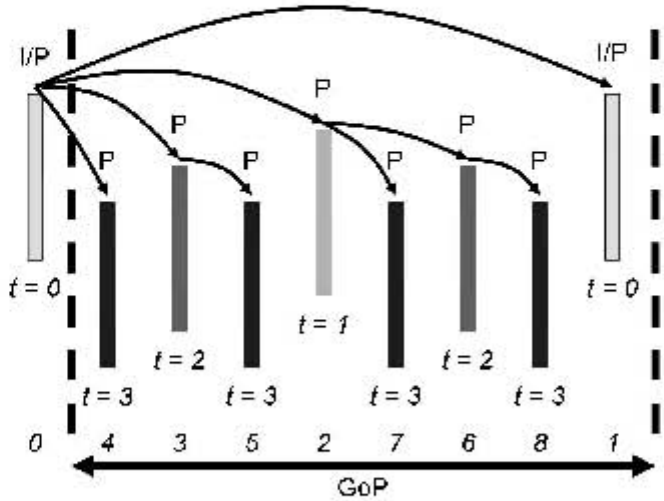


Figure 2.2: Video coding using a hierarchical prediction structure with a zero structural encoding/decoding delay. The numbers at the bottom indicate the picture coding order, while $t = k$ specifies the temporal layer with k representing the corresponding temporal layer identifier.

predicted by pictures of a lower layer. The number T of temporal layers in a sequence is determined by $T = \log_2 S_G + 1$, where S_G is the group of pictures (GoP) size defined in H.264/SVC as the distance between two consecutive pictures belonging to the temporal base layer, typically I or P pictures.

2.3.2 Spatial Scalability

Regarding spatial scalability, a layered coding approach is used to encode different picture sizes of an input video source. As illustrated in Figure 2.3, each layer corresponds to a target spatial resolution and is referred to as a spatial layer or dependency layer, which is denoted by a layer identifier d. For the spatial base layer, which provides an H.264/AVC compatible bit stream for the lowest spatial resolution, the layer identifier is $d = 0$, and it is increased by 1 from one spatial layer to the next until reaching the highest dependency layer. Furthermore, a spatial layer may

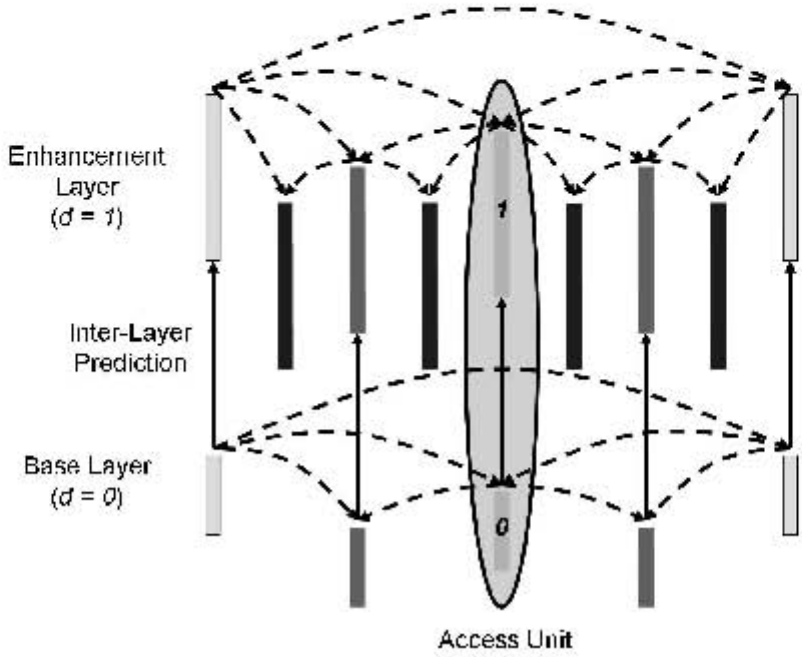


Figure 2.3: Layered coding approach with inter-layer prediction for enabling spatial scalable coding. The shadowed area indicates an AU and the number inside it specifies the coding order of the corresponding pictures.

contain several temporal layers according to the output frame rate targeted for each one. In particular, the H.264/SVC standard specifies a maximum of 8 supported dependency layers.

To limit the memory requirements and decoding complexity derived from this multi-layer coding approach, the same coding order for all supported spatial layers is used. Specifically, the coding order is on an *access unit* (AU) basis, where an AU is defined as the union of all the representations with different spatial resolutions for a given time instant. Inside an AU, the encoded pictures are transmitted in increasing order of their corresponding dependency identifiers, as illustrated in Figure 2.3.

In each spatial layer, the traditional motion-compensated and intra-prediction modes are supported as for non-scalable video coding. In addition to these ba-

sic coding tools of H.264/AVC, the H.264/SVC standard also provides *inter-layer* prediction tools, which exploit the redundancies between consecutive spatial layers inside an AU for improving the coding efficiency of the spatial enhancement layers (see Figure 2.3). Specifically, the H.264/SVC standard supports the following three inter-layer prediction modes:

- *Inter-Layer Intra-Prediction*: The enhancement layer MBk is predicted from an upsampled version of the reconstructed co-located intra-coded 8×8 block[1] of the layer used as reference. In order to prevent noisy signal components in the prediction signal, the H.264/AVC deblocking filter is applied to the reconstructed block of the spatial reference layer before the upsampling process.

- *Inter-Layer MBk Mode and Motion Prediction*: Only the residual signal of the enhancement layer MBk (i.e., neither intra-prediction modes nor motion information) is transmitted. If the reference signal is an intra-coded 8×8 block, the enhancement layer MBk is predicted by inter-layer intra-prediction; otherwise it is inter-coded using the associated side information of the co-located 8×8 block in the reference layer.

- *Inter-Layer Residual Prediction*: The residual signal of the enhancement layer MBk is predicted from an upsampled version of the residual signal corresponding to the co-located 8×8 block in the reference layer. Then, the resulting difference signal of the enhancement layer is encoded.

For further details about these inter-layer prediction modes, the reader is referred to [Schwarz et al., 2007]. Notice that the inter-layer information does not have to be always the most suitable for removing redundancies. For instance, in sequences with high spatial detail and slow motion, the signal used as reference for motion-

[1]For a dyadic spatial scalability consisting of doubling the picture width and height from one layer to the next, a MBk in a spatial enhancement layer corresponds to an 8×8 sub-MBk in its reference layer.

compensated prediction usually provides a better approximation of the original signal than that of the reference layer for inter-layer prediction.

Similar to priors SVC standards, the scalable extension of H.264/AVC supports spatial scalability with arbitrary resolution ratios. Furthermore, neither horizontal nor vertical resolution can diminish from one dependency layer to the next. Furthermore, two additional spatial scalability modes are specified: an enhancement layer picture may contain specific parts of its reference layer picture that presents the same or higher spatial resolution; and an enhancement layer picture may also contain other parts beyond the limits of the reference layer picture. The cropped parts of the reference picture may be altered on a frame basis. Moreover, the H.264/SVC design even provides tools for spatial scalable coding of interlaced video sources.

2.3.3 Quality Scalability

The H.264/SVC standard defines two types of quality scalable coding: CGS and MGS.

Coarse Grain Scalable Coding

CGS coding is a special case of spatial scalability with identical picture sizes between two consecutive dependency layers, thus not requiring the upsampling process and the inter-layer deblocking filter for the reference layer MBks that are intra coded. When using the inter-layer prediction tools for CGS coding, the texture information SNR is typically improved by decreasing the quantization step size from one dependency layer to the next. Figure 2.4 shows a generic block diagram of the CGS coding process that emphasizes the need for gradually improving the reconstruction fidelity when increasing the dependency layer identifier.

Nevertheless, the multi-layer approach for CGS coding only allows for supporting a few selected rate points in the scalable bit stream. Hence, the number of supported bit rates (or qualities) matches up with the number of dependency layers. In addition, switching between CGS layers is only possible at predefined points in the bit stream

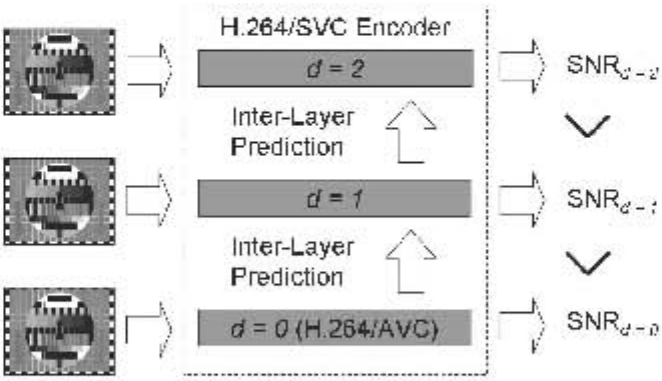


Figure 2.4: Generic block diagram of the CGS coding process with three dependency layers. The coded video SNR is increased from one dependency layer to the next.

corresponding to instantaneous decoding refresh (IDR) pictures[2]. Furthermore, the CGS coding efficiency decreases as the relative bit rate difference between consecutive dependency layers becomes smaller.

Medium Grain Scalable Coding

In order to provide a larger range of rate points for a more flexible bit stream adaptation and error robustness, as well as for coding efficiency improvement, MGS coding can be used. This type of quality scalability can be viewed as a high-level signaling variation of CGS coding, which allows for switching between different MGS layers in any AU, so two consecutive AUs can be decoded with a different quality level representations.

As it is shown in Figure 2.5, instead of using the spatial identifier d to define the number of quality refinements, a new signaling element q is included in the syntax to identify different quality representations within a dependency layer. For the quality

[2]Given a dependency layer, when a picture is signaled as IDR picture, the current and all following pictures can be decoded without decoding any previous pictures of the same layer. Moreover, in H.264/SVC the location of IDR pictures in a dependency layer can be different to that of any other layer.

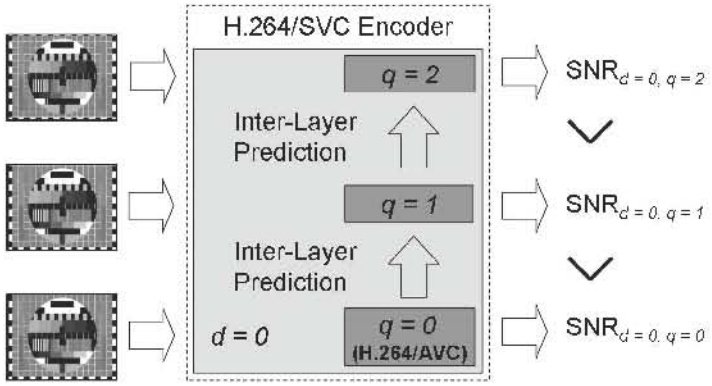


Figure 2.5: Generic block diagram of the MGS coding process with three quality refinements. The coded video SNR is increased for one quality layer to the next.

base layer representing the lowest reconstruction fidelity, the MGS layer identifier is $q = 0$, while for the SNR enhancement layers the identifier q is increased by 1 as the SNR becomes higher. Thus, given a dependency layer d, those coded picture packets[3] corresponding to a certain quality refinement layer q can be removed at any arbitrary point, so that the decoder cannot detect a deliberate quality refinement packet loss.

Nevertheless, this packet-based quality scalable coding results in mismatch or lack of synchronization between encoder and decoder since the uneven extraction of quality refinement packets between AUs implies that some reference pictures for motion compensation are different at both transmission sides. The mismatch disappears if motion compensation is only performed in the quality base layer (that is, the enhancement layer pictures are encoded using either intra-prediction or inter-layer prediction), but this approach significantly decreases the coding efficiency of enhancement layers. Alternatively, the enhancement layer with the highest available

[3]The coded source content is encapsulated in packets so-called *Network Abstraction Layer* (NAL) units, which are suitable for transmission over RTP/IP-based networks or use in packet oriented multiplex environments.

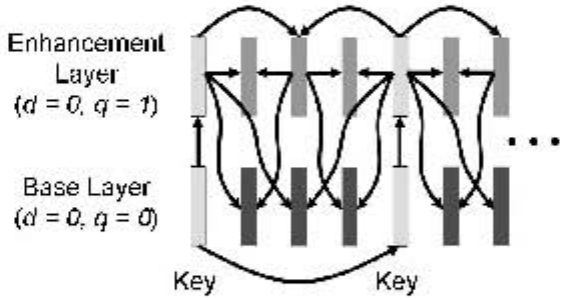


Figure 2.6: Key picture concept for hierarchical prediction structures. Between two K pictures, the enhancement representation is used for prediction. However, for the K pictures, only the base representation is used.

quality can be always used as reference for motion-compensated prediction. This solution ensures low complexity and high coding efficiency, but important mismatches when some quality refinement packets are discarded. In order to minimize this drawback, the *key* (K) picture concept combined with hierarchical prediction structures is introduced in H.264/SVC.

All pictures belonging to the temporal base layer ($t = 0$) are transmitted as K pictures (I/P pictures), and only for them the quality base layer representations ($q = 0$) are used for motion-compensated prediction, as illustrated in Figure 2.6. Thus, no mismatch is introduced in the motion compensation loop of such a temporal layer, thus avoiding the drift propagation between GoPs. The K pictures are actually used as resynchronization points between both transmission sides. Regarding the pictures of the temporal enhancement layers, the highest available quality of the reference pictures is used for motion-compensated prediction, so that a high coding efficiency is provided. The use of this K-picture-based MGS coding approach allows for restricting the mismatch propagation to the inside of a GoP.

It is also worth mentioning that in MGS coding the transform coefficients of an original slice can be optionally distributed among several slices, so that different quality refinements are provided. To make the explanation of this tool easier, let

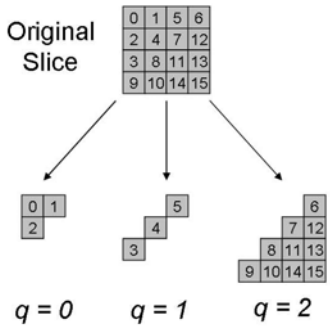


Figure 2.7: Example of DCT coefficient partition for MGS coding.

us consider an example as that illustrated in Figure 2.7, in which the original slice is going to be partitioned into three quality refinement slices. As can be observed, the slice with layer identifier $q = 0$ includes only those DCT coefficient values whose scan indexes are inside the range from 0 to 2, thus representing the lowest quality version of the original slice; the slice of the first enhancement layer ($q = 1$) contains the following three transform coefficient values; and the slice with layer identifier $q = 2$ provides a high-frequency quality refinement by including the remaining DCT coefficient values. The first and the last scan index for the transform coefficients of each quality refinement slice are signaled in the corresponding slice header.

Specifically, H.264/SVC allows for a total of 15 MGS refinements within a dependency layer, thus providing a finer bit rate granularity in the R-D space when compared to CGS coding.

2.4 SVC High-Level Design

As previously stated, in the scalable extension of the H.264/AVC standard, the three types of scalability described can be combined. However, an H.264/SVC encoder does not need to be configured to support all types of scalability. According to the application requirements, which determine the set of target spatio-temporal resolutions or reconstruction qualities as well as their corresponding QoS, a proper H.264/SVC

encoder configuration should provide only the required scalability levels so that the coding complexity is not unnecessarily increased.

2.4.1 Combined Scalability

Temporal scalability is achieved on an AU basis by ignoring the higher temporal layer AUs for frame rate reduction. Inside an AU, the coding structure is organized in dependency layers (spatial scalability or CGS) as illustrated in Figure 2.8, which depicts an example of H.264/SVC encoder structure with two dependency layers and more than one quality refinement per layer. A spatial layer usually represents a specific picture size, though two identical spatial resolutions can be located at consecutive dependency layers for CGS coding. Spatial/CGS layers are associated with a dependency identifier d. Additionally, each dependency layer may contain one or more MGS layers representing different degrees of fidelity. All quality layers within a dependency layer are identified by a quality identifier q and must correspond to the same spatio-temporal resolution. For MGS layers with identifier $q > 0$, always the immediately lower refinement layer $q-1$ is used for inter-layer prediction. However, for $q=0$, any present quality layer of a lower dependency layer $d-1$ can be selected as reference for inter-layer prediction.

2.4.2 Bit Stream Switching

One paramount difference between the concept of dependency layers and MGS refinement layers is that switching between different dependency layers is only supported at IDR pictures in certain AUs, whereas switching between different quality refinements is virtually possible at any AU. A quality refinement can be transmitted via dependency layer (CGS coding: different d) or by means of additional quality refinements inside a dependency layer (MGS: same d, different q). This H.264/SVC high-level design does not change the basic decoding process. Nevertheless, MGS coding offers a larger number of refinements (until 15 quality layers), as well as a finer granularity

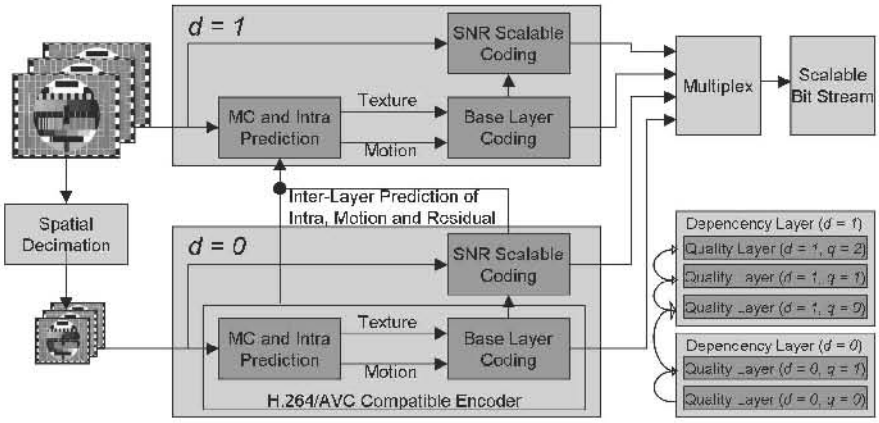


Figure 2.8: Example of an H.264/SVC encoder structure with two dependency layers. For the lowest spatial resolution, two quality refinements are shown, while for the highest spatial resolution, three quality refinements are included.

for bit rate adaptation since the coded picture NAL units of enhancement layers can be removed at any time instant, whenever K pictures are inserted in the temporal base layer in order to enable efficient packet-based quality scalable coding.

It should be emphasized that in the H.264/SVC context the IDR points of a dependency layer can be different to those of any other dependency layer. In other words, all pictures within an AU do not have to be cataloged as IDR. Thus, although a picture signaled as IDR picture for an enhancement layer cannot be encoded via motion-compensated prediction, any other picture inside the AU can be efficiently encoded using all supported prediction tools.

2.4.3 System Interface

In order to make the bit stream manipulations easier, the 1-byte header of H.264/AVC NAL units is extended by 3 additional bytes so that SVC is supported. This extended header of H.264/SVC NAL units contains the layer identifiers (d,q,t), and additional information regarding bit stream adaptations. One of those additional

syntax elements is a priority identifier p that indicates which NAL units are more important for the bit stream extraction process (see Section 2.5 for more details).

Each scalable bit stream includes a sub-stream that conforms to a non-scalable profile of H.264/AVC, but standard H.264/AVC NAL units do not contain the extended H.264/SVC NAL unit header. These additional bytes are not only useful for bit stream adaptations, but are required to decode any scalable bit stream. In order for standard H.264/AVC NAL units to be adapted to the SVC conditions, so-called prefix NAL units containing the H.264/SVC NAL unit header extension are also included in the bit stream.

The H.264/SVC standard also specifies additional *Supplemental Enhancement Information* (SEI) messages, which contain basic information such as the scalability features of the bit stream and the bit rate of each supported layer. This additional information is useful for the decoding and related processes like the sub-stream extraction or display. More detailed information on the system interface of H.264/SVC is provided in [Wang et al., 2007]. Information on the RTP payload format and the file format for H.264/SVC are given in [Wenger et al., 2006] and [Amon et al., 2007], respectively.

2.5 Bit Stream Extraction

This subsection focuses on basic ideas about the bit stream extraction mechanism to find the desired sub-stream according to the designated extraction option, typically a given target bit rate at a given spatio-temporal resolution (notice that this extraction option comes from the QoS for a specific decoding terminal). The extraction of a specific spatio-temporal resolution (d_k, t_k) can be done by the following steps [Wien et al., 2007b]:

1. From the SEI messages, find the dependency identifier d_k and the temporal identifier t_k associated with the spatio-temporal resolution to be found, as well as the bit rate information of each layer.

2. Discard all AUs with a temporal identifier greater than t_k.

3. Discard all NAL units with a dependency identifier grater than d_k.

The extraction of a particular bit rate for the spatio-temporal target resolution requires the comparison of the target bit rate, denoted as R_k, to the average bit rate generated by each quality refinement layer inside the layer (d_k, t_k). $R_0^{(d_k,t_k)}$ specifies the minimum extractable bit rate, i.e., that associated with the quality base layer $q=0$.

4. If $R_0^{(d_k,t_k)}$ is greater than R_k, then the requested spatio-temporal target resolution cannot be extracted.

5. Otherwise, process all quality refinement layers until that with identifier q_k whose bit rate (extracted from the SEI messages) is less than or equal to R_k. In order to meet the target bit rate, some NAL units of the immediately higher refinement layer q_k+1 can be discarded. If CGS coding is instead provided for quality scalability, this truncation of quality layers is not allowed and only full dependency layers are included in the extracted sub-streams.

In the case that NAL units contain a priority identifier p, an optimized bit stream extraction discards them in increasing order of priority until the target bit rate is reached. A more detailed information about the concept of optimized bit stream extraction is provided in [Amonou et al., 2007].

2.6 Application Areas

Several industries and application areas, such as video conference [Eleftheriadis et al., 2006], streaming [Wien et al., 2007a], surveillance [Schaefer et al., 2005] and IP television (IPTV) broadcast [Wiegand et al., 2009], have benefited from the SVC features for multimedia information delivery. A representative application example of H.264/SVC is a wireless transmission service with heterogeneous clients [Schaefer

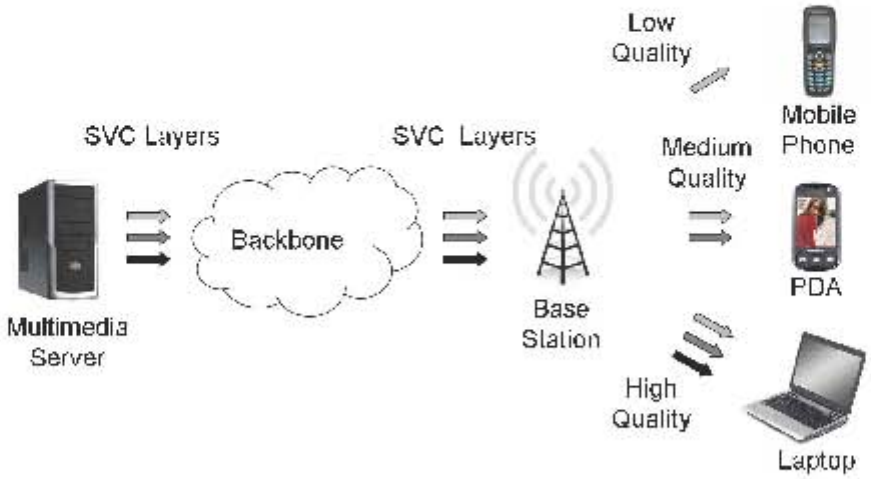


Figure 2.9: H.264/SVC video transmission through wireless broadcast networks.

et al., 2005]. As can be observed in Figure 2.9, the scalable bit stream is transmitted from the multimedia content server to the base station of a wireless broadcast network, and then several sub-streams (for instance, low, medium, and high quality) are generated according to the capabilities of the target decoding terminals.

Another interesting benefit of the SVC features for a video transmission service is that a minimum QoS can be guaranteed by using unequal error protection (UEP) or unequal erasure protection (UXP) techniques [Liebl et al., 2004] to ensure an error free transmission of more important sub-streams, such as the base layer of an H.264/SVC bit stream, as illustrated in Figure 2.10. UEP/UXP can be used on top of the already existing channel forward error correction.

2.7 Profiles

As any video coding standard, H.264/SVC must provide suitable solutions for a variety of applications. Nevertheless, in spite of the fact that H.264/SVC contains a wide range of coding tools, certain applications may only require some of them. For instance, in low-delay applications such as video conference the use of B pictures is

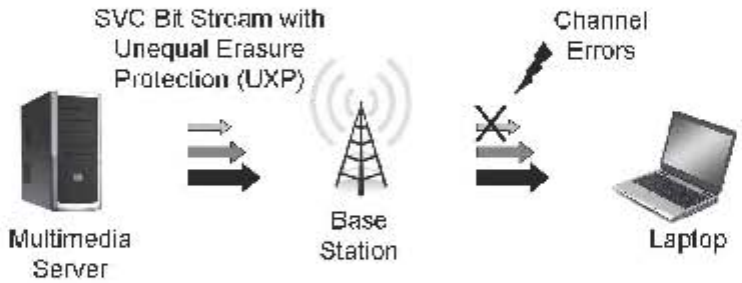


Figure 2.10: Unequal erasure protection and H.264/SVC quality layers.

not required, so the coding complexity may be reduced. In order to provide flexibility in the coding complexity according to the application requirements, the concepts of *profile* and *level* are introduced.

A **profile** defines a subset of coding options that can be used for generating a bit stream, and hence it determines the complexity degree of the implemented encoder. Thus, all decoders complying with a particular profile must support all included coding options. On the other hand, a **level** specifies limitations on certain key parameters of the bit stream, such as spatial resolution, frame rate, bit rate, etc.

Particularly, the scalable extension of the H.264/AVC standard supports three profiles [Wiegand et al., 2007]: *Scalable Baseline*, *Scalable High*, and *Scalable High Intra*.

Scalable Baseline Profile

- Mobile broadcast, conversational, and surveillance applications.
- Resolution ratios of 1.5 and 2 between successive spatial layers in both horizontal and vertical direction and MBk-aligned cropping.
- Progressive sources.
- Enhancement layers: B slices, weighted prediction, CABAC, 8×8 luma transform.

- Base layer conforms to the H.264/AVC Baseline Profile.

Scalable High Profile

- Broadcast, streaming, and storage applications.
- Restrictions of Scalable Baseline Profile are removed.
- Base layer conforms to the H.264/AVC High Profile.

Scalable High Intra Profile

- Professional applications.
- Only IDR pictures (for all layers).
- Scalable High Profile is supported.

Chapter 3

Rate Control (RC) for Video Coding

Video information is inherently variable in both the spatial and temporal domains. The motion activity (i.e., high or low) and the motion nature (i.e., chaotic or predictable) of a video sequence determine its temporal complexity, whereas the amount of spatial detail per picture determines its spatial complexity. It is well-known that the spatio-temporal complexity of two consecutive pictures can be different, so the output bit rate per encoded picture will also vary. The higher video complexity, the higher bandwidth required to transmit or store the encoded bit stream with reasonable distortion.

The output bit rate of a video encoder is a key parameter to be controlled in order to meet certain application requirements, such as the R-D trade-off, the buffer size or the maximum allowed storage capacity. For this purpose, an RC algorithm is embedded in the video encoder. The aim of the RC algorithm is to adjust certain encoder parameters affecting the bit rate, so that the average bit rate of the encoded video meets a specific target bit rate without exceeding some practical constraints, while maximizing the reconstructed video quality.

Some of the encoder parameters that an RC algorithm can adjust include the picture size, the picture type (I, P or B), the frame rate, and the QP, though this last is the most widely used encoder element to control the bit rate. From now on,

let us consider the QP as the unique controlling mechanism to explain, along the following sections, the basic concepts of RC, as well as some relevant RC algorithms proposed in the literature.

In Section 3.1 the fundamental issues to be considered when designing an RC algorithm are explained. Section 3.2 presents a general framework for optimal RC. In Sections 3.3 and 3.4 a typical RC scheme is described for both single-layer video coding and SVC. Section 3.5 describes the R-D concept from the RC point of view, emphasizing those R-D models commonly used in practical video coding applications. Finally, in Section 3.6, the differences between the two main RC types according to the target application (CBR and VBR) are discussed, and some of the most relevant CBR and VBR approaches for both single-layer video coding and SVC are briefly described.

3.1 Requirements and Constraints

The RC algorithm is an informative part of video coding standards whose design has become one of the major research areas for codec designers and network administrators. In order to obtain the desired bit rate, the RC algorithm should take into account the following requirements and constraints:

- *Rate and distortion trade-off*: There is an inherent trade-off between the distortion D and the bit rate R generated by a lossy source encoder. Since the distortion is a decreasing function of the output bit rate [Shannon, 1948], and the information loss is mainly introduced at the quantizer, a small QP value involves high bit rate and low distortion, and vice versa. The QP value determines the quantization step size Q for the coding unit. For instance, in H.264/AVC and its scalable extension, QP and Q are related by $Q = 2^{(QP-4)/6}$, and a discrete set of 52 QP candidates from 0 to 51 is allowed for each coding unit.

  In short, the aim of the RC algorithm is to achieve the maximum quality (or

minimum distortion) while meeting a given bit rate constraint. Typically, in video coding the quality is measured by means of the peak SNR (PSNR):

$$PSNR = 10 \log \frac{255^2}{MSE},$$

where MSE denotes the *mean squared error* between the original and the distorted signal.

- *Complexity*: The complexity requirement for an RC algorithm mainly comes from the target application. For instance, for real-time video coding applications, computationally complex RC algorithms should be avoided, while for off-line applications such as video storage, multiple-pass RC methods are feasible and lead to significant improvements in R-D performance. Thus, a good trade-off between an optimal and an efficient RC solution should be found according to the application.

- *Transmission Constraints*: For video transmission systems, some buffer constraints are introduced in order to guarantee that encoder and decoder buffers do not incur in overflow or underflow and, hence, a continuous play out is performed at the decoder side. For end-to-end real-time video communications, a delay constraint should be met to avoid temporal artifacts such as jitter and jerkiness [Huynh-Thu and Ghanbari, 2008].

- *Storage Constraints*: For video storage applications, a bit budget constraint is additionally imposed owing to the maximum allowed storage capacity.

3.2 Optimal RC Solutions

An optimal solution to the RC problem involves minimizing the distortion subject to a bit rate constraint, i.e., finding the quantizer that achieves minimum distortion while meeting a given bit rate constraint. To this end, several strategies can be employed: minimizing the average distortion (MINAVE) [Ramchandran et al., 1994],

minimizing the maximum distortion (MINMAX) [Schuster et al., 1999], or minimizing the distortion variation (MINVAR) [Lin and Ortega, 1998]. Following [Chen and Ngan, 2007a], these criteria are outlined bellow.

The hybrid video coding paradigm exhibit a strong temporal component (inter-frame redundancy); consequently, the bit rate R_j and the distortion D_j generated by an encoded picture at the time instant j not only depend on the quantizer Q_j used for this picture, but also on those of its neighboring pictures.

In particular, the unconstrained MINAVE criterion can be formulated as follows:

$$\mathbf{Q}^* = \left(Q_0^*, \ldots, Q_{J-1}^*\right) = \underset{(Q_0,\ldots,Q_{J-1})}{\operatorname{argmin}} \sum_{j=0}^{J-1} D_j\left(Q_{j-a}, \ldots, Q_{j+b}\right), \tag{3.1}$$

where a past and b future frames are considered, and J denotes the number of pictures involved in the optimization problem. The MINMAX criterion can be expressed as:

$$\mathbf{Q}^* = \left(Q_0^*, \ldots, Q_{J-1}^*\right) = \underset{(Q_0,\ldots,Q_{J-1})}{\operatorname{argmin}} \left(\max_{j\in\{0,\ldots,J-1\}} \left\{D_j\left(Q_{j-a}, \ldots, Q_{j+b}\right)\right\} \right); \tag{3.2}$$

and, finally, the MINVAR criterion can be formulated as follows:

$$\mathbf{Q}^* = \left(Q_0^*, \ldots, Q_{J-1}^*\right) = \underset{(Q_0,\ldots,Q_{J-1})}{\operatorname{argmin}} \sum_{j=0}^{J-1} \left|D_j\left(Q_{j-a}, \ldots, Q_{j+b}\right) - D_{j-1}\left(Q_{j-a-1}, \ldots, Q_{j+b-1}\right)\right|. \tag{3.3}$$

As previously stated in Section 3.1, according to the target application, some constraints can be applied to these unconstrained problems. For instance, for a video storage application where there is a maximum storage capacity R_{max}, Equations (3.1), (3.2) and (3.3) are subject to

$$\sum_{j=0}^{J-1} R_j\left(Q_{j-a}, \ldots, Q_{j+b}\right) \leq R_{max}. \tag{3.4}$$

On the other hand, when considering a real-time transmission application, the distortion criteria are subject to

$$0 \leq V_j \leq BS, \tag{3.5}$$

where V_j is the current buffer fullness and BS is the buffer size in bits.

In order to solve these constrained RC problems, two well-known approaches have been typically employed: Lagrange relaxation method [Shoham and Gersho, 1988] and dynamic programing [Zhai and Katsaggelos, 2007]. The former converts the "hard" constrained problem into an "easy" unconstrained problem parametrized by the Lagrange multiplier [Schuster et al., 1999]. The latter attempts to solve the multivariable problem by solving a series of single variable problems.

The main drawback of the optimal RC solutions is their high computational complexity, which makes them unfeasible for many video coding applications. Taking into account the balance between complexity and performance, the so-called model-based RC algorithms are used instead in most video encoders. Although these RC algorithms do not guarantee an optimal solution for the RC problem, their results are competitive in terms of R-D performance with acceptable complexity.

A more detailed description of the model-based RC algorithms is given in the following two sections.

3.3 Model-Based RC Solutions

Figure 3.1 shows a block diagram of a model-based RC algorithm for video coding [Rezaei, 2008]. As can be observed, a video encoder compresses an original video sequence and generates an output bit stream. The encoded bit stream can be either transmitted through a channel (CBR or VBR) or stored in a multimedia storage device. Before transmitting the compressed video sequence, an encoder buffering process is required to guarantee a continuous play out at the decoding size. At a given time instant j, the encoder buffer is filled with the amount of bits b_j yield by the encoding of a picture, and then emptied at the transmission channel rate R_T (in bits per seconds –bps–). The opposite process happens at the decoder buffer, which receives bits from the channel at R_T bps and then extracts the b_j bits for decoding. In order to avoid overflow and underflow in both transmission sides, the

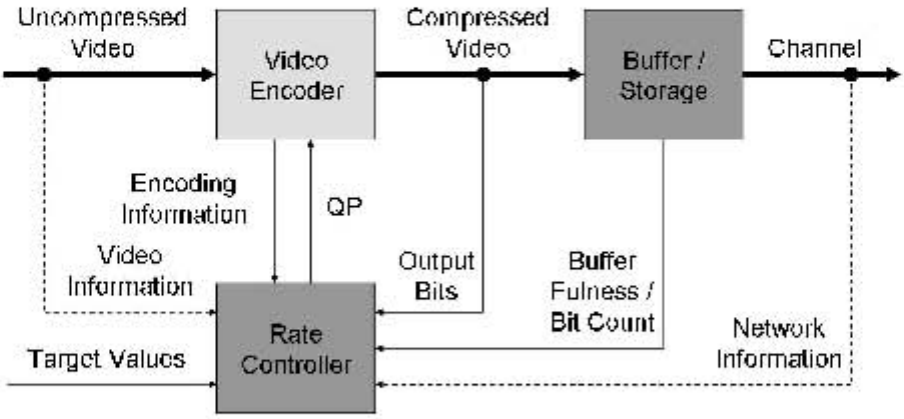


Figure 3.1: Block diagram of an RC algorithm for single-layer video coding.

HRD constraints[1] [Ribas-Corbera et al., 2003] should be taken into account. To this purpose, the bit budget per picture is limited by imposing a lower and upper bounds on the encoder buffer fullness.

The buffer can be either real (for real-time applications) or virtual (for off-line applications). A real buffer is located physically between the video encoder and the channel (see Figure 3.1), whereas a virtual buffer aims to simulate a real buffer. The current buffer fullness is used as a feedback signal by the RC algorithm in order to prevent the overflow and underflow risks. In the case of real-time applications, some feedback information from the channel or the decoder can also be used for a better bit rate regulation. If a multimedia storage device is used instead, the current amount of bits stored in the device is used as feedback signal by the rate controller in order to check the difference between the current bit count and the maximum storage capacity. Additionally, the RC algorithm can employ as feedback signal

[1]In video coding standards, a compliant bit stream must be decoded by a hypothetical decoder that is conceptually connected to the output of the encoder and consists of a decoder buffer, a decoder, and a display unit. This virtual decoder is known as the HRD in H.26X and the *video buffering verifier* in MPEG. The encoder must create a bit stream such that the hypothetical decoder buffer does not incur in overflow or underflow.

some information regarding the encoding process, such as a distortion measurement of the compressed video or the picture type.

Besides the aforementioned feedback information, some feed-forward information regarding pre-analysis of the uncompressed video source can be useful for the RC process. A common pre-analysis operation is the scene change detector attempting to find those pictures in which the video complexity abruptly vary and, therefore, the R-D relation of compressed video source [Yu et al., 1998, Sanz-Rodríguez et al., 2007b]. A perceptual analysis can also be performed to introduce elements from the human visual system behavior such as salient regions of visual attention [Itti, 2004, Tang et al., 2006] or areas with fine details [Minoo and Nguyen, 2005], so that more resources (target bits) can be assigned to them. For off-line applications, an analysis window collecting several past and future pictures or even the whole sequence in multi-pass encoding are used to extract some relevant video characteristics [Westerink et al., 1999, Yu et al., 2001].

In order for the RC process to work properly, the RC algorithm must comply with some given target values. The most commonly used target values are the following, some of them have already been introduced:

- *Target bit rate*: It is defined as the average bit rate to be reached after encoding the whole video sequence, which matches up with the channel bit rate in a transmission application. It is actually the most essential parameter owing to its direct relation to the QoS.

- *Maximum instantaneous bit rate*: In transmission environments, this parameter is related to the maximum buffer size. In storage applications, it could be identified with the access speed limit of the storage device.

- *Maximum exceeded bit count*: In transmission applications, it can be seen as a long-term restriction for the produced average bit rate when the target bit rate is exceeded. In storage applications, it is directly the percentage that the average bit rate can be exceeded after encoding the sequence.

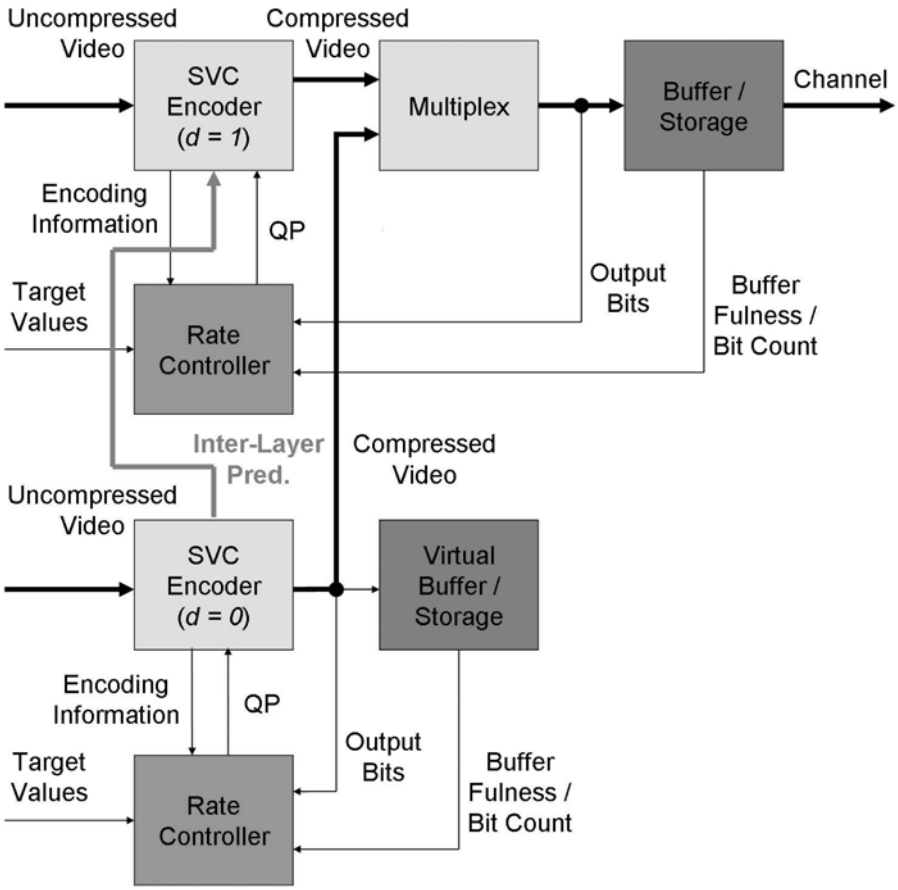


Figure 3.2: Block diagram of an RC algorithm for SVC. A particular case with two dependency layers is shown.

- *Buffer/storage parameters*: If a buffer is used, these parameters are related to the HRD: initial buffer fullness (related to the read time of the first picture from the decoder buffer), target buffer occupancy (typically set to the initial buffer fullness), and buffer size. If a storage device is used, its maximum storage capacity is considered.

In the case of SVC, the typical block diagram of an RC algorithm is depicted in Figure 3.2 for the particular case of two dependency layers. The optional feedback and feed-forward signals have been removed for clarity reasons. As can be observed,

a rate controller and the corresponding buffer or storage device are located at each dependency layer. If quality layers were included within a dependency layer, an RC block and the associated buffer or storage device per quality layer would also be located. For transmission applications, the encoder buffering process of those sub-streams to be decoded by the target decoding terminals is performed in virtual buffers. However, for the buffering process of the complete scalable bit steam, the buffer can be either real or virtual. For storage applications, the buffers are replaced by multimedia storage devices, in which the corresponding recording processes are used as feedback signals.

3.4 Operation Steps in a RC Algorithm

Generally, the RC algorithms follow two operation steps for QP selection: 1) bit allocation; and 2) QP estimation. These two processes are described in the sequel.

3.4.1 Bit Allocation

The bit allocation process aims to assign a bit budget to a video segment such as a GoP, a picture, a *basic unit* (BU)[2] or a MBk, according to the input information available to the RC algorithm. This bit assignment for a video segment usually follows a strategy based on levels. For instance, assuming a BU as video segment, three bit allocation levels are typically considered [Ma et al., 2003]: 1) GoP level; 2) picture level; and 3) BU level. The GoP-level bit allocation estimates the target bits for those pictures in the GoP that haven't been encoded yet. To this end, parameters such as the target bit rate, the total number of pictures in the GoP, the buffer occupancy and the amount of bits generated by the already encoded pictures are used. The picture-level bit allocation computes the target bit rate for a picture

[2]The BU concept was introduced in RC for H.264/AVC to define a group of successive MBs in a picture that share the same QP value. The set of possible BU sizes (generally an entire fraction of the total number of MBks in a picture) goes from one MBk to an entire frame.

considering the current buffer state, the frame rate, the picture type (I, P or B), the HRD constraints, and the total number of bits for the remaining pictures in the GoP (taken from the upper level). Finally, the BU-level bit allocation estimates the bit budget for the current BU according to the amount of bits consumed by the previously encoded BUs and its spatial complexity relative to those of the remaining BUs.

Nevertheless, the bit allocation algorithm does not necessarily have to be composed of the levels described above, but other configurations are also possible according to the codec properties, application requirements, and preferences of the RC designers. In some approaches, such as [ISO/IEC, 1993] for MPEG-2 and [Ribas-Corbera and Lei, 1999] for H.263, GoP-level bit allocation is not considered. In MPEG-4 an object-based bit allocation scheme can be used [Ronda et al., 1999, Lee et al., 2000]. In hierarchical video coding, as in SVC, a bit allocation stage for each hierarchical level can also be incorporated [Seo et al., 2010]. Other schemes propose higher-level bit control schemes for a better visual quality consistency. For instance, in [de-Frutos-López et al., 2010] a sliding-window (at least one GoP length) is included as high level bit allocation for target GoP bit budget.

3.4.2 Quantization Parameter (QP) Estimation

In this stage, a proper QP value for the video segment is computed according to its coding complexity and the allocated bit budget. To this end, an R-D model is normally used for QP estimation. A model-based R-D function can be obtained analytically or empirically. In analytical modeling, the R-D function typically derives from the modeling of the quantized transform coefficient distribution. Once the video segment is encoded, the model parameters are updated for the next QP estimation process. In empirical modeling, the R-D function for a given video segment is obtained by interpolating a set of R-D points coming from encodings of the current and/or previous video segments.

More details about R-D modeling for QP estimation are given next.

3.5 Rate-Distortion (R-D) Modeling for Video Compression

As stated in Section 3.1, there is a relation between the rate and the distortion generated by a lossy source encoder. This R-D modeling is based on the R-D theory [Berger, 1971], which is an important branch of information theory created by Claude E. Shannon [Shannon, 1948,Shannon, 1959] and has been widely used in lossy image and video compression [Ortega and Ramchandran, 1998]. Particularly, a lossy video coding scheme focuses on finding a good trade-off between the distortion and the output bit rate for a given input video sequence. In the R-D theory, an R-D function is formulated to provide a lower bound for the rate at a given distortion level. For an RC algorithm, the knowledge of such an R-D function is of paramount importance to find the QP value that minimizes the distortion of encoded video subject to a bit rate constraint.

Several approaches can be employed, from high complexity operational R-D (ORD) functions to simpler model-based R-D functions, which are summarized in the following subsections.

3.5.1 Operational R-D Functions

In practical lossy video coding applications, the R-D theory is used to allocate resources to a specific video segment given a particular encoder, but usually only a finite number of coding modes (e.g., intra or inter coding, quantization step size, etc.) and, consequently, a discrete set of R-D pairs, are considered. Additionally, it is practically impossible to find closed-form expressions for the $R(D)$ or $D(R)$ functions for general sources. For these reasons, ORD theory [Zhai and Katsaggelos, 2007] is used instead to generate an R-D function.

Let $\mathcal{Q}$ be the number of supported coding modes or parameters defined by a specific lossy video encoder. Each of the parameter choices will lead to a pair of rate and distortion values. Such a pair of operational points is indicated by a circle

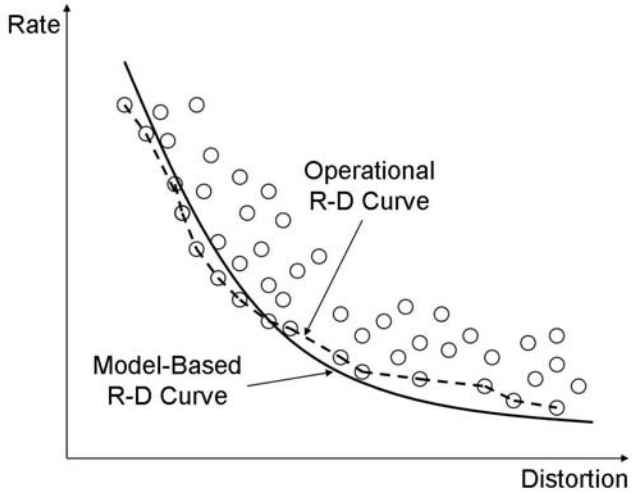


Figure 3.3: Illustration of an ORD curve and its associated model-based R-D curve.

in Figure 3.3 [Chen and Ngan, 2007a]. The lower bound of all these R-D pairs is referred to as the so-called ORD function, which is shown by a dashed line in the same figure. The set of source coding parameters that results in the ORD function can be formally defined as:

$$\mathcal{Q}_{ORDF} = \{q\colon q \in \mathcal{Q}, R(q) \geq R(p) \Rightarrow D(q) < D(p), \forall p \in \mathcal{Q}\},$$

where $R(q)$ and $D(q)$ are the rate and distortion generated by a particular coding mode q, respectively.

The ORD function defines the best achievable performance for a given source and compression framework, thus providing useful information so that any of the optimal solutions for rate control described in Section 3.2 can be attained. However, the generation of the R-D pairs results in a high computational complexity and, therefore, intolerable delay for many practical video coding applications. For this reason, simpler methods, so-called model-based R-D functions, are commonly adopted in spite of the fact that they may suffer from relatively large estimation errors, as shown in Figure 3.3 where the solid curve represents an approximation of the ORD curve by a model-based R-D function.

3.5.2 Model-Based R-D Functions

As already pointed out, there are two kinds of model-based R-D functions: analytical models and empirical models. The former are typically inferred from the statistical properties of the DCT coefficient distribution. The second estimate the R-D curve empirically by means of regression techniques. Some representative examples of these R-D estimation methods are described below.

Analytical R-D modeling

In this kind of modeling, it is assumed that an accurate estimation of the DCT coefficient distribution of an image or video source will lead to a more accurate estimation of the rate and distortion for a particular quantizer. Assuming that the DCT coefficients are uniformly quantized with a quantization step size Q, the rate model due to quantization $R(Q)$ can be estimated by the entropy of the quantized DCT coefficients $H(Q)$, that obeys the following expression:

$$H(Q) = -\sum_{i=-\infty}^{\infty} P(iQ) \log_2 \left(P(iQ)\right), \tag{3.6}$$

with

$$P(iQ) = \int_{(i-\frac{1}{2})Q}^{(i+\frac{1}{2})Q} f_X(x)\, dx, \tag{3.7}$$

where $P(iQ)$ is the probability that a coefficient is quantized as iQ, with $i = 0, \pm 1, \pm 2, \ldots$, and $f_X(x)$ is a probability density function (PDF) that fits the DCT coefficient distribution. On the other hand, the MSE-based distortion due to quantization $D(Q)$ is given by

$$D(Q) = \sum_{i=-\infty}^{\infty} \int_{(i-\frac{1}{2})Q}^{(i+\frac{1}{2})Q} |x - iQ|^2 f_X(x)\, dx. \tag{3.8}$$

According to these equations for uniform quantization, at high bit rates the R-D model for a quantized signal can be expressed as [Hang and Chen, 1997]:

$$R(Q) = \frac{1}{2} \log_2 \left(\frac{\epsilon^2 \beta \sigma^2}{Q^2} \right), \tag{3.9}$$

$$D(Q) = \frac{Q^2}{\beta}, \tag{3.10}$$

where ϵ^2 is a model coefficient that is set to 1 for uniform, 1.4 for Gaussian, and 1.2 for Laplacian distributions; β is a parameter that equals 12 for small quantization step sizes, but it needs to be empirically adjusted based on samples of the R-D curve in order to account for larger Q values; σ^2 denotes the variance of the random variable.

However, for a wider range of bit rates, other more accurate R-D models have been proposed. Considering a Laplacian PDF for DCT coefficient modeling, which has been suggested by several works such as [Smoot and Rowe, 1996] and [Lam and Goodman, 2000], the entropy function of the quantized signal given in Equations (3.6) and (3.7) was derived in [Moscheni et al., 1993]; but a simple approximation to that solution was later proposed by Ribas-Corbera *et al.* [Ribas-Corbera and Lei, 1999], which obeys the following expression:

$$H(Q) = \begin{cases} \frac{1}{2}\log_2\left(2e^2\frac{\sigma^2}{Q^2}\right), & \frac{\sigma^2}{Q^2} > 1/2e \\ \frac{e}{\ln 2}\frac{\sigma^2}{Q^2}, & \frac{\sigma^2}{Q^2} \leq 1/2e, \end{cases} \tag{3.11}$$

where $1/2e$ is a threshold that determines the operating point for high rate ($\sigma^2/Q^2 > 1/2e$) or low rate ($\sigma^2/Q^2 \leq 1/2e$). The picture distortion model was defined as:

$$D(Q) = \frac{1}{N}\sum_{i=1} N\frac{Q_i^2}{12}, \tag{3.12}$$

which was derived from the quantization error of a uniform random variable using a quantization step size Q_i for the ith MBk of a picture made up of N MBks.

Chiang *et al.* [Chiang and Zhang, 1997] also assumed that the source statistics have a Laplacian PDF and proposed the following quadratic model:

$$R(Q) = aQ^{-1} + bQ^{-2}, \tag{3.13}$$

where a and b are the model coefficients. MSE or *mean absolute difference* (MAD) between the original and predicted signals can be used instead of Q and the same formulation is still valid.

The model in Equation (3.13) was further improved by [Lee et al., 2000] as follows:

$$R(Q) = M\left(aQ^{-1} + bQ^{-2}\right) + H, \tag{3.14}$$

where M is a coding complexity measurement based on MAD, and H is the overhead information such as header and motion vector bits. For the particular case of the H.264/AVC standard, the current MAD cannot be computed until the RDO process for the picture has been finished. The RDO process consists of evaluating for each MBk of the picture every coding mode to find that achieving the best trade-off between distortion and rate for a specific Q. Particularly, it is performed by means of the minimization of the following Lagrangian functional:

$$J(K, M|Q) = D_{REC}(K, M|Q) + \lambda_{MODE} R(K, M|Q), \tag{3.15}$$

where K denotes a particular MBk, M denotes a specific coding mode for MBk (inter, intra, skip), $D_{REC}(K, M|Q)$ and $R_{REC}(K, M|Q)$ are, respectively, the distortion and the rate of the reconstructed MBk, and λ_{MODE} is a Lagrange parameter that depends on Q, specifically $\lambda_{MODE} = 0.85 \times 2^{(Q-12)/3}$. The MAD computation requires to have Q in advance so that the RDO process can be performed, but an RC algorithm using Equation (3.14) requires the MAD for QP estimation. This so-called *chicken and egg dilemma* is solved by predicting the current MAD using the following linear regression [Ma et al., 2003]:

$$\widehat{MAD} = a_1 MAD_{PREV} + a_2, \tag{3.16}$$

where a_1 and a_2 are the model coefficients, and MAD_{PREV} is the real MAD of the previous picture.

Furthermore, He *et al.* [He et al., 2001, He and Mitra, 2001, Kim et al., 2001] proposed an efficient R-D model based on the percentage of zeros in the quantized DCT coefficients, which was denoted as ρ. Considering also a Laplacian distribution for transform coefficients, the ρ-domain rate and distortion models obey the following expressions:

$$R(\rho) = \theta\left(1 - \rho\right), \tag{3.17}$$

and

$$D(\rho) = \sigma^2 \exp\left(-\alpha\left(1-\rho\right)\right), \tag{3.18}$$

where θ and α are the model parameters. The mapping between Q and ρ is then calculated based on the distribution of the DCT coefficients.

Dai *et al.* [Dai et al., 2006] assumed that the DCT coefficients followed a mixture Laplacian distribution and, based on the previous work [Dai et al., 2003], proposed the following square root (SQRT) R-D model:

$$PSNR(R) = AR + B\sqrt{R} + C, \tag{3.19}$$

where A and B are estimated from at least two R-D samples and $C = 10\log\left(255^2/\sigma^2\right)$.

Alternatively, Kamaci *et al.* [Kamaci et al., 2005] observed that the zero-mean Cauchy PDF was more accurate for estimating the distribution of the transform coefficients than the traditional Laplacian PDF and, starting from Equations (3.6), (3.7) and (3.8), proposed the following exponential rate and distortion models:

$$R(Q) = aQ^{-\alpha}, \tag{3.20}$$

$$D(Q) = bQ^{-\beta}, \tag{3.21}$$

where $\{a, \alpha\}$ and $\{b, \beta\}$ represent the rate and distortion model coefficients, respectively.

Empirical R-D modeling

Besides the aforementioned analytical R-D models, there are other RC approaches that use empirical methods to estimate the R-D characteristics from previously observed data. Lin *et al.* [Lin and Ortega, 1998] proposed an R-D estimation scheme based on constructing the whole R-D curve by means of cubic-spline interpolation from a set of R-D pairs computed by running several times the coding system. Zhao

et al. [Zhao et al., 2002] applied similar interpolation methods to MPEG-4 fine grain scalability (FGS) [Li, 2001]. In the work proposed by Ding *et al.* [Ding and Liu, 1996], the R-D curves are fitted by mathematical models with several control parameters, which are estimated from the observed R-D data of the current picture, so a feedback re-encoding method is also required. Furthermore, some non-linear regression methods were used for empirical R-D modeling [Vetro et al., 2003, Kim, 2003] as well.

3.6 Constant Bit Rate (CBR) Coding and Variable Bit Rate (VBR) Coding

According to the target application, two kinds of coding methods can be distinguished [Ortega, 2000]: CBR and VBR coding. In CBR coding, commonly used for low-delay transmission through CBR channels (for instance, real-time video conference), a short-term target bit rate adjustment is required to ensure low buffer delay. The typical operation range in the PSNR-rate space of an RC algorithm for CBR video is illustrated in Figure 3.4 (left). As can be observed, a fine control of the instantaneous bit rate (represented by a shadowed area) around the target bit rate R_T is pursued at the expense of a large quality variation due to the video content variability in natural scenes.

On the other hand, in VBR coding, typically used for video streaming, broadcast or digital storage, a long-term target bit rate adaptation and a major short-term bit rate variation are feasible so that a high visual quality consistency is provided. VBR coding can be open-loop or closed-loop. In open-loop or so-called *unconstrained* (U)-VBR coding [Lakshman et al., 1998], no bit rate constraints are imposed and the video sequence is encoded with almost constant QP to achieve a relative constant quality of encoded video. U-VBR coding is typically used for source and traffic modeling. Instead, in closed-loop VBR coding, the video sequence is encoded by using a VBR controller that shapes the output bit rate according to some constraints related

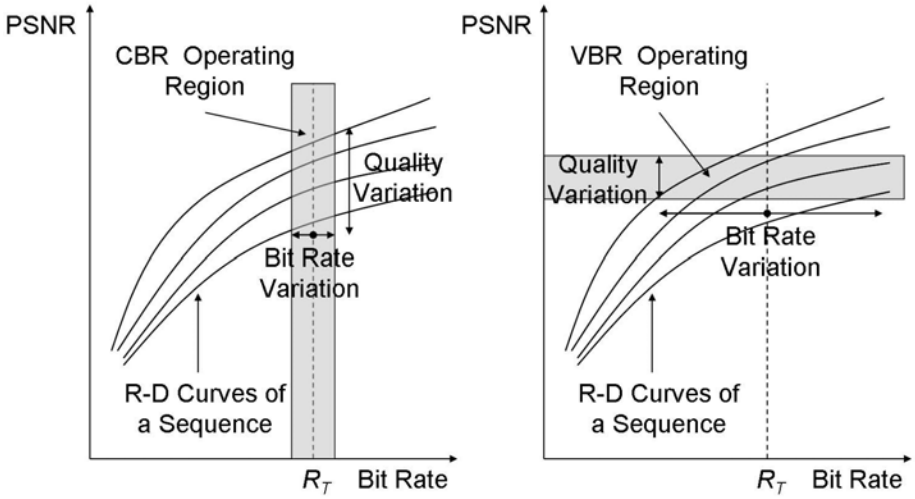


Figure 3.4: Operating regions in the PSNR-rate space for CBR coding (left) and VBR coding (right).

to the degree of variability allowed in the bit rate. This kind of VBR coding is more appropriate for real scenarios of transmission or storage that require high encoded video quality. As can be observed in Figure 3.4 (right), a VBR control algorithm operates in an almost constant quality region in the PSNR-rate space (represented by a shadowed area), despite potential large instantaneous bit rate variations. Nevertheless, in real-time transmission applications these bit rate fluctuations can be absorbed whenever the encoder buffer is large enough at the expense of a longer delay. It is also worth noting that a VBR bit stream can be transmitted through either a CBR channel or a VBR transmission networking infrastructure [Lakshman et al., 1998], as long as the application can tolerate the resulting delay.

3.6.1 CBR Control Algorithms

A large number of CBR controllers proposed in the literature have focused on modeling the DCT coefficients to provide analytical R-D functions for QP estimation. In single-layer video coding, several R-D functions have been presented:

- *Logarithmic model*: [Hang and Chen, 1997] (Equation (3.9)), [Ribas-Corbera and Lei, 1999] (Equation 3.11)) and [Tao et al., 2000].

- *Linear model*: [ISO/IEC, 1993, Ma et al., 2005].

- *Quadratic model*: [Chiang and Zhang, 1997] (Equation (3.13)), [Lee et al., 2000] (Equation (3.14)), [Ma et al., 2003, Xie and Zeng, 2006, Chen and Ngan, 2007b, Kwon et al., 2007]

- *ρ-domain model*: [He et al., 2001] (Equation (3.17)).

- *Exponential model*: [Kamaci et al., 2005, Sanz-Rodríguez et al., 2010] (Equation (3.20)).

All of these R-D models assume Laplacian PDF for DCT coefficient distribution, except that proposed in [Tao et al., 2000], which uses a Gaussian PDF, and the exponential models proposed in [Kamaci et al., 2005] and [Sanz-Rodríguez et al., 2010] for frame-layer and BU-layer RC, respectively, which use a Cauchy PDF. Some of these approaches employed separate analytical models according to the type of information contained in the video source. In particular, Chen *et al* [Chen and Ngan, 2007b] proposed separate rate models for luminance and chrominance transform coefficients, while Kwon *et al.* [Kwon et al., 2007] proposed separate rate models for source and header bits (typically the amount of header bits is estimated by averaging those of the previously encoded pictures).

Regarding CBR controllers for SVC, most of them have also employed some of the model-based R-D functions indicated above: logarithmic [Xu et al., 2007], linear [Liu et al., 2008], quadratic [Leontaris and Tourapis, 2007], ρ-domain [Pitrey et al., 2009, Liu et al., 2010b], and exponential [Cho et al., 2009, Liu et al., 2010a] models. Alternatively, a few CBR schemes proposed a strategy based on estimating the QP increment instead the QP value itself, for instance, [Sun et al., 2008] for H.264/AVC and [Anselmo and Alfonso, 2007] for H.264/SVC.

Although the RC algorithm is not a normative part of video coding standards, some of the aforementioned approaches form part of their reference implementations, such us the Test Model Version 5 for MPEG-2 [ISO/IEC, 1993], the Verification Model Version 8 for MPEG-4 [Chiang and Zhang, 1997], the Test Model Near-Term 8 for H.263 [Ribas-Corbera and Lei, 1999], and the Joint Model for H.264/AVC [Ma et al., 2003]. Nevertheless, these reference RC algorithms typically result in poor performance at specific stages that are critical in low-delay environments, especially at the bit allocation stage.

Several improved bit allocation algorithms have been proposed for CBR coding. Jiang and Ling [Jiang and Ling, 2006] proposed a novel bit allocation based on an improved frame complexity prediction. Sanz-Rodríguez *et al.* [Sanz-Rodríguez et al., 2007a] completed the bit allocation algorithm given in [Ma et al., 2003] by incorporating a saw-tooth-shaped model of the target buffer level when using stored-B pictures. Furthermore, Yu *et al.* [Yu et al., 2005] introduced perceptual considerations for a proper QP selection on a MBk basis.

Another fundamental problem in RC for low-delay applications is how to determine an appropriate QP value for the first picture of the sequence, so that the low-delay buffer fullness does not incur in overflow or underflow. Most of the initial QP selection methods for H.264/AVC [Jing and Chau, 2006, Wang and Kwong, 2008] and H.264/SVC [Yang et al., 2010, Sanz-Rodríguez and Díaz-de-María, 2011b] consisted of measuring first the spatial activity of the picture (gradient-based methods are typically used [Kim et al., 1999]), and then estimating the QP by means of model-based R-D modeling. Although other methods have also included the temporal activity for a more proper estimation of video complexity [Wu and Kim, 2009, Czuni et al., 2006], they required the buffering of some future pictures to compute the temporal differences between them, thus producing additional delay that could be intolerable in some low-delay transmissions. Finally, it is also worth mentioning that the HRD requirements have been taken into account in [Sanz-Rodríguez and Díaz-de-María, 2011b], so that the buffer overflow and underflow risks could be prevented at the

beginning of the encoding process.

3.6.2 VBR Control Algorithms

Several VBR control solutions for single-layer video coding have been proposed for a variety of applications, such as video streaming and broadcast [Lin and Ortega, 1998, Mohsenian et al., 1999, Rezaei et al., 2008], one-pass digital storage [Ding and Liu, 1996, Jagmohan and Ratakonda, 2003, de-Frutos-López et al., 2010], or two-pass digital storage [Westerink et al., 1999, Yu et al., 2001]. Other schemes, such as those in [Ding, 1997] and [Bai et al., 2002], take the advantage of networking infrastructures supporting VBR transport [Ortega, 2000] to improve the visual quality while reducing the buffer delay.

With respect to SVC, only a few algorithms have been designed for MPEG-4 FGS [Zhao et al., 2002, Zhang et al., 2003, Dai et al., 2006] and H.264/SVC [Lee et al., 2010], but this last approach allows for few SVC configurations and target applications, among other reasons, because the HRD compliance is not taken into account at the enhancement layers.

From the R-D modeling point of view, some of these methods rely on some well-known analytical R-D models [Dai et al., 2006, Mohsenian et al., 1999, Jagmohan and Ratakonda, 2003, Westerink et al., 1999, Zhang et al., 2003] and empirical R-D models [Lin and Ortega, 1998, Ding and Liu, 1996, Zhao et al., 2002] for QP estimation, while other RC schemes estimate the QP increment with respect to a reference QP [Ding, 1997, Bai et al., 2002, Rezaei et al., 2008, de-Frutos-López et al., 2010], in order to reduce its variation for the sake of visual quality consistency. Actually, this last approach based on QP increment estimations has grown in popularity during the last years. Moreover, specific R-D modeling for H.264/SVC MGS has been designed as well [Mansour et al., 2008].

The bit allocation problem has also been studied in VBR coding for both single-layer video coding (see Section 3.2) and SVC. In particular, R-D models for optimal bit allocation among spatial, quality, and temporal layers have been proposed in [Cho

et al., 2009] and [Liu et al., 2010a]. Likewise, the optimal distribution of the total target bit rate among dependency layers for visual quality maximization has been addressed in [Unterweger and Thoma, 2007] for real-time applications.

Chapter 4

VBR Controller for H.264/SVC

In comparison to CBR coding, VBR coding allows for better visual quality consistency at the expense of more resources in terms of (instantaneous) transmission bandwidth and delay [Lakshman et al., 1998, Zhang et al., 2003]. VBR coding is used in many popular video applications such as streaming, broadcast, one-pass and multiple-pass digital storage. These applications can benefit from the scalable features of SVC to provide a wider range of decoding possibilities and a better adaptation to RTP/IP-based network.

Although the CBR control problem for SVC has been studied during the last years, there is still a lack of practical solutions concerning VBR environments. Specifically, the HRD compliance required to properly deliver the scalable video content deserves, in our opinion, more attention, since, as far as we know, there is no any complete solution able to manage the buffer control at every dependency layer, while providing good visual quality consistency for the corresponding sub-streams.

This chapter focuses on a VBR control algorithm for real-time H.264/SVC applications with buffer constraints. In particular, the proposed VBR controller aims to provide scalable bit streams that satisfy two essential requirements at all the considered spatial and quality layers: HRD compliance and quality consistency. To this end, for the sake of quality consistency, the proposed VBR controller assumes that

consecutive pictures within the same scene often exhibit similar degrees of complexity and should be encoded using similar QP values. Consequently, the VBR controller aims to reduce unnecessary QP fluctuations by allowing for just limited variations of QP with respect to a reference value. In particular, a novel GP regression method has been proposed to estimate the proper QP variation (instead of the QP absolute value). Furthermore, as shown later on, the low computational cost is another valuable property of the proposed RC algorithm.

This chapter is organized as follows. In Section 4.1 a brief overview of the VBR control algorithm is given, outlining the different subsystems that make up the whole algorithm. In Section 4.2 every subsystem of the rate controller located at each specific spatial or quality layer is described in detail. In Section 4.3 some issues concerning the complexity of the algorithm are discussed. Section 4.4 describes the experimental setup and shows and discusses the experimental results. Finally, some conclusions in Section 4.5 close the chapter.

4.1 System Overview

The proposed RC scheme is illustrated in dark gray in Figure 4.1 for an H.264/SVC encoder consisting of two spatial/CGS layers. Let us denote as D the number of dependency layers, identified as $d = \{0, 1 \ldots D-1\}$, and let us denote as $T^{(d)}$ the number of temporal layers for a particular dependency layer, identified as $t = \left\{0, 1 \ldots T^{(d)} - 1\right\}$. Alternatively, for the sake of notation consistency with the proposed method, we will refer to the maximum temporal layer identifier $T^{(d)} - 1$ as $t_{max}^{(d)}$.

Each dependency layer d involves a rate controller $RC^{(d)}$ and a buffer. The buffer of the base layer is a virtual buffer, while that of the enhancement layer is a real buffer to assume a transmission scenario. The buffer at the layer d receives the contributions of the layers from $(0, 0)$ to $(d, t_{max}^{(d)})$ and simulates the encoder buffering process of the highest temporal resolution sub-stream. Thus, both the

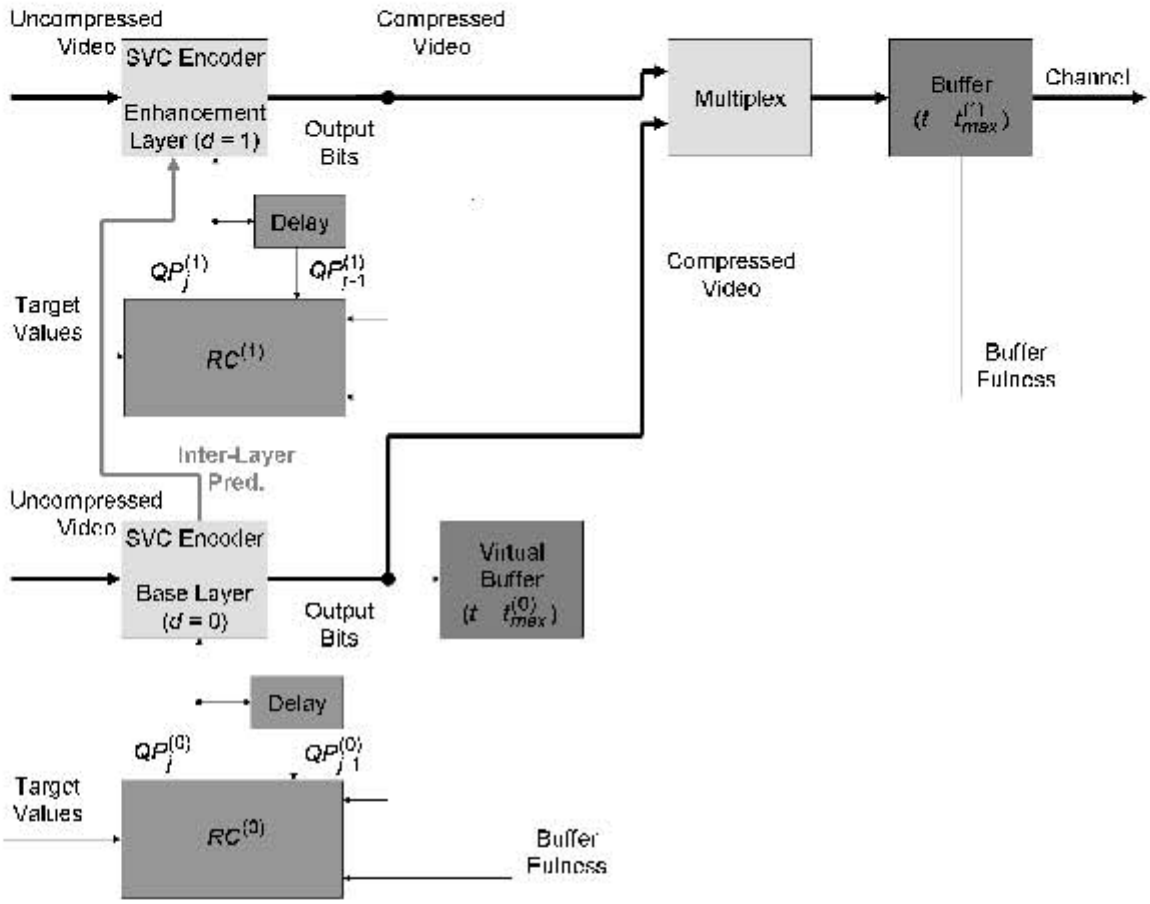


Figure 4.1: Block diagram of the proposed H.264/SVC RC scheme for two dependency layers ($D = 2$).

buffer and the corresponding sub-stream will be identified as $(d, t_{max}^{(d)})$ to indicate that the video packets with higher spatio-temporal identifiers will be discarded by the target decoder. The generation of each HRD-compliant sub-stream depends on two fundamental parameters: the target bit rate $R^{(d,t_{max}^{(d)})}$ to and the output frame rate $f_{out}^{(d,t_{max}^{(d)})}$. It should be noticed that $R^{(d,t_{max}^{(d)})}$ must be higher than those associated with lower layers, i.e.,

$$R^{(d-x,t_{max}^{(d)}-y)} \leq R^{(d,t_{max}^{(d)})} \qquad x = 0, 1 \ldots d, \qquad y = 0, 1 \ldots t_{max}^{(d)},$$

since those lower layers form part of the sub-stream $\left(d, t_{max}^{(d)}\right)$. Furthermore, $f_{out}^{(d,t_{max}^{(d)})}$

is obtained from the corresponding temporal layer as follows:

$$f_{out}^{(d,t_{max}^{(d)})} = f_{in}^{(d)} \times 2^{-(\log_2 S_G - t_{max}^{(d)})},$$

with $f_{in}^{(d)}$ being the input sequence frame rate, and S_G the GoP size.

In the case that a particular dependency layer contained additional $Q^{(d)}$ MGS refinements, denoted as $q = \left\{1 \ldots Q^{(d)} - 1\right\}$ (it should be noticed that $q = 0$ represents the quality base layer for a given dependency layer), a rate controller $RC^{(d,q)}$ and the corresponding virtual buffer would be located at each spatio-quality layer (d, q). However, in order to make the notation easier, hereafter we will only consider spatial/CGS and temporal scalability.

In order to encode the jth picture with layer identifier (d, t), the $RC^{(d)}$ module should provide an appropriate $QP_j^{(d)}$, on a frame basis, so that the QP fluctuation is minimized (to improve visual quality consistency), while the buffer fullness $V^{(d,t_{max}^{(d)})}$ is maintained at secure levels. To this end, the $RC^{(d)}$ module operation leans on three input parameters:

1) The fullness $V^{(d,t_{max}^{(d)})}$ of the corresponding buffer.

2) The amount of bits yield by the encoding of the spatial/CGS layers 0 to d for a given time instant. Henceforth, following the H.264/SVC nomenclature (see Chapter 2), we will refer to this amount of bits as AU output bits $AU^{(d,t)}$.

3) The QP value used to encode the previous picture of the same dependency layer $QP_{j-1}^{(d)}$.

A proper QP increment $\Delta QP^{(d)}$ is estimated from the two firsts, and $QP_{j-1}^{(d)}$ is employed as a reference value to obtain the final quantization parameter as follows:

$$QP_j^{(d)} = QP_{j-1}^{(d)} + \Delta QP^{(d)}. \tag{4.1}$$

This approach takes advantage of the fact that the VBR environments allow for a slow QP evolution in order to maintain a consistent visual quality. Thus, it assumes

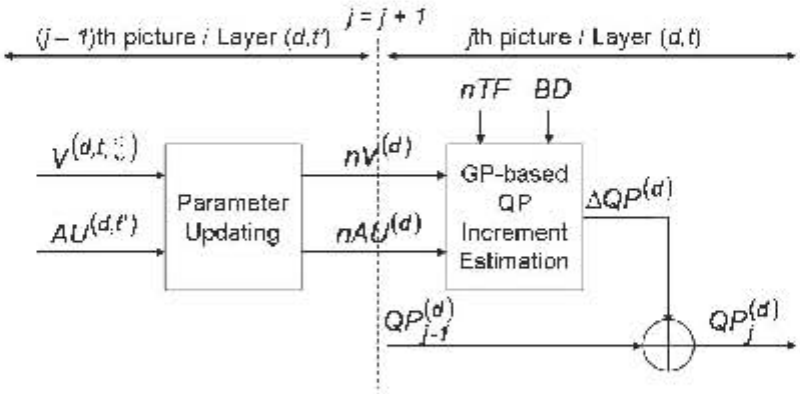


Figure 4.2: Block diagram of the rate controller $RC^{(d)}$ for a specific dependency layer d.

similarity between consecutive frames and aims to model only those QP changes required to compensate for large bit rate deviations owing to time-varying video complexity. Consequently, the method to predict $\Delta QP^{(d)}$ becomes the main focus of this VBR controller.

It is also worth noting that, in the case of CGS scalability, the QP obtained is lower bounded by the QP of the reference layer, so that a higher quality for the enhancement layer is ensured:

$$QP_j^{(d)} = \min\Big[QP_j^{(d-1)}, QP_j^{(d)}\Big]. \tag{4.2}$$

The VBR control algorithm for a specific spatial or CGS layer, i.e., the algorithm that obtains an appropriate incremental variation of QP for the jth picture with identifier (d, t) is illustrated in Figure 4.2. As shown in the figure, the $RC^{(d)}$ module is organized in two stages named *parameter updating* and *GP-based QP increment estimation*.

1) *Parameter updating stage*: after encoding the $(j-1)^{th}$ picture with layer identifier (d, t') (t' is used instead of t because the previous picture can belong to a different temporal layer), some parameters required to estimate $\Delta QP^{(d)}$ are updated. In particular, the following two parameters are updated.

a) A normalized version of the buffer fullness, denoted as $nV^{(d)}$.

b) A normalized version of the amount of bits generated by the AU, denoted as $nAU^{(d)}$.

The normalized versions of the buffer fullness and the AU output bits are defined as follows:

$$nV^{(d)} = \frac{V^{(d,t_{max}^{(d)})}}{BS^{(d,t_{max}^{(d)})}}, \tag{4.3}$$

$$nAU^{(d)} = \frac{AU^{(d,t')}}{G^{(d,t',t_{max}^{(d)})}}, \tag{4.4}$$

where $V^{(d,t_{max}^{(d)})}$ has already been defined as the current buffer fullness; $BS^{(d,t_{max}^{(d)})}$ denotes the buffer size in bits; $AU^{(d,t')}$ has already been defined as the AU output bits; and $G^{(d,t',t_{max}^{(d)})}$ denotes the target bits for the AU at the layer (d, t') in order for the sub-stream $(d, t_{max}^{(d)})$ to satisfy the target bit rate constraint $R^{(d,t_{max}^{(d)})}$.

2) *GP-based QP increment estimation stage*: before encoding the jth picture, a proper $\Delta QP^{(d)}$ value is estimated from four parameters (whose selection is discussed in Section 6.1): $nV^{(d)}$, $nAU^{(d)}$, and two additional constant parameters that are included so that the achieved solution is able to work in a variety of scenarios. The first constant parameter, denoted as nTF, is the normalized target buffer fullness with respect to the buffer size, and the second, denoted as BD, is the maximum buffering delay (or buffer size in seconds), which is related to that measured in bits as $BS^{(d,t_{max}^{(d)})} = BD \times R^{(d,t_{max}^{(d)})}$. Then the $\Delta QP^{(d)}$ value is added to $QP_{j-1}^{(d)}$ as indicated in Equation (4.1). In particular, a non-linear relation between the aforementioned input parameters and the desired $\Delta QP^{(d)}$ value has been obtained by means of a GP regression that is able to deal with a wide range of practical situations, as described in Section 4.2.2.

4.2 RC Stages

The stages of the rate controller module $RC^{(d)}$ are described in detail in the following subsections.

4.2.1 Parameter Updating

The aim of this subsection is to describe the updating procedure for parameters $nV^{(d)}$ and $nAU^{(d)}$. The updating equations for $nV^{(d)}$ and $nAU^{(d)}$ require the previous computation of both the buffer fullness and the AU target bits. In turn, the computation of the buffer fullness requires to obtain the AU output bits, and the estimation of AU target bits requires to estimate the average texture and motion complexities for each temporal layer. Therefore, the calculation of all of these quantities are explained first, to end up with the updating equations for $nV^{(d)}$ and $nAU^{(d)}$.

Computation of the AU Output Bits

Assuming that the picture coding order in H.264/SVC is established so that the AUs are sequentially encoded (the encoding of an AU starts when the previous has been completed), the number of bits generated by $AU_{j-1}^{(d,t')}$ obeys:

$$AU_{j-1}^{(d,t')} = \sum_{m=0}^{d} \left(b_{j-1}^{(m,t')} + h_{j-1}^{(m,t')} \right), \tag{4.5}$$

where $b_{j-1}^{(m,t')}$ and $h_{j-1}^{(m,t')}$ are, respectively, the amount of texture bits and header plus motion data bits generated by the $(j-1)^{th}$ picture, with spatio-temporal layer identifier (m, t').

Buffer Fullness Updating

Once the AU output bits have been obtained, the buffer fullness is updated as follows:

$$V_j^{(d,t_{max}^{(d)})} = V_{j-1}^{(d,t_{max}^{(d)})} + AU_{j-1}^{(d,t')} - \frac{R^{(d,t_{max}^{(d)})}}{f_{out}^{(d,t_{max}^{(d)})}}. \tag{4.6}$$

Estimation of the Average Texture and Motion Complexities of a Layer $(\mathrm{d}, \mathrm{t}')$

Let us define $\overline{C}_{TEX}^{(d,t')}$ as the average texture complexity of the encoded pictures at the dependency layers 0 to d belonging to the temporal layer t'. The following updating equation is proposed:

$$\overline{C}_{TEX}^{(d,t')} = \alpha \sum_{m=0}^{d} \left(Q_{j-1}^{(m)} b_{j-1}^{(m,t')} \right) + (1-\alpha)\overline{C}_{TEX}^{(d,t')}, \tag{4.7}$$

where α is a forgetting factor that is set to 0.5 in our experiments, and $Q_{j-1}^{(m)}$ is the quantization step value associated with $QP_{j-1}^{(m)}$. Likewise, the average motion complexity $\overline{C}_{MOT}^{(d,t')}$ is defined as:

$$\overline{C}_{MOT}^{(d,t')} = \beta \sum_{m=0}^{d} h_{j-1}^{(m,t')} + (1-\beta)\overline{C}_{MOT}^{(d,t')}, \tag{4.8}$$

where β is a forgetting factor that is also set to 0.5 in our experiments. It is also worth mentioning that for the lowest temporal layer, which can include I or P pictures, these average complexities are reset (that is, α and β are temporary set to 1) when the current type of picture is different from the previous one at the same temporal layer, so that potential complexity mismatches due intrinsic encoding differences between I and P pictures are prevented.

Estimation of the AU Target Bits

In order for the sub-stream with layer identifier $(d, t_{max}^{(d,t)})$ to satisfy the target bit rate constraint $R^{(d,t_{max}^{(d,t)})}$, the amount of AU output bits should be controlled according to a bit budget $G^{(d,t',t_{max}^{(d)})}$. The AU target bits obeys the following model:

$$G^{(d,t',t_{max}^{(d)})} = G_{NOM}^{(d,t_{max}^{(d)})} + \Delta G_{TEX}^{(d,t',t_{max}^{(d)})} + \Delta G_{MOT}^{(d,t',t_{max}^{(d)})}, \tag{4.9}$$

where $G_{NOM}^{(d,t_{max}^{(d,t)})}$ is the nominal bit budget:

$$G_{NOM}^{(d,t_{max}^{(d)})} = \frac{R^{(d,t_{max}^{(d)})}}{f_{out}^{(d,t_{max}^{(d)})}}, \tag{4.10}$$

and $\Delta G_{TEX}^{(d,t')}$ and $\Delta G_{MOT}^{(d,t')}$ represent the bit increments that depend on the relative texture and motion complexities among temporal layers, respectively, i.e.:

$$\Delta G_{TEX}^{(d,t',t_{max}^{(d)})} = \frac{R^{(d,t_{max}^{(d)})}}{f_{out}^{(d,t_{max}^{(d)})}} \left(\frac{\overline{C}_{TEX}^{(d,t')} \sum_{u=0}^{t_{max}^{(d)}} N^{(d,u)}}{\sum_{u=0}^{t_{max}^{(d)}} \left(\overline{C}_{TEX}^{(d,u)} N^{(d,u)} \right)} - 1 \right), \tag{4.11}$$

$$\Delta G_{MOT}^{(d,t',t_{max}^{(d)})} = \overline{C}_{MOT}^{(d,t')} - \frac{\overline{C}_{TEX}^{(d,t')} \sum_{u=0}^{t_{max}^{(d)}} \left(\overline{C}_{MOT}^{(d,u)} N^{(d,u)} \right)}{\sum_{u=0}^{t_{max}^{(d)}} \left(\overline{C}_{TEX}^{(d,u)} N^{(d,u)} \right)}, \tag{4.12}$$

with $N^{(d,u)}$ being the total number of pictures per GOP with layer identifier (d, u). A more detailed explanation of this bit allocation algorithm is given in Appendix A.

$\mathbf{nV^{(d)}}$ and $\mathbf{nAU^{(d)}}$ Updating Equations

After encoding the $(j-1)^{th}$ picture with layer identifier (d, t'), the parameters required to estimate the incremental variation of QP for the next picture are finally updated by means of the following expressions:

$$nV^{(d)} = \max \left[0, \min \left[\frac{V^{(d,t_{max}^{(d)})}}{BS^{(d,t_{max}^{(d)})}}, 1 \right] \right], \tag{4.13}$$

$$nAU^{(d)} = \max \left[\frac{1}{2}, \min \left[\frac{AU^{(d,t')}}{G^{(d,t',t_{max}^{(d)})}}, 2 \right] \right]. \tag{4.14}$$

Since these parameters bear the current state of the encoding process in terms of buffer occupancy and target bit rate mismatch, the most appropriate QP variation should be derived from them. For instance, if $nV^{(d)}$ were close to 1 (overflow risk) and $nAU^{(d)}$ were close to 2 (large bit rate mismatch), then $\Delta QP^{(d)}$ would be high in order to quickly correct such mismatches. On the other hand, if $nV^{(d)}$ were close to 1 but $nAU^{(d)}$ were also close to 1, then $\Delta QP^{(d)}$ would not be high, so that the visual quality is maintained. Nevertheless, it is not easy to infer practical decision-making rules from particular examples such as the previous ones. Instead, this task has been addressed through a carefully designed $\Delta QP^{(d)}$ estimation process that is described in the next section.

4.2.2 Gaussian Process (GP)-Based QP Increment Estimation

As previously stated, the proposed $\Delta QP^{(d)}$ estimation method operates on the following input vector:

$$\mathbf{X}^{(d)} = \left(nV^{(d)}, nAU^{(d)}, nTF, BD\right)^T, \tag{4.15}$$

implicitly assuming that all the buffers share the same nTF and BD values.

A carefully designed GP is used to estimate $\Delta QP^{(d)}$ from the input vector $\mathbf{X}^{(d)}$. The mean predictions of the GP for regression can be expressed as:

$$\Delta QP^{(d)} = \text{round}\left[w_0 + \sum_{i=1}^{M} w_i H_i\left(\mathbf{X}^{(d)}\right)\right], \tag{4.16}$$

where M is the number of basis functions $\left\{H_i\left(\mathbf{X}^{(d)}\right)\right\}_{i=1\ldots M}$, $\mathbf{w} = \{w_i\}_{i=1\ldots M}$ the output weights, and w_0 the bias. It should be noticed that the output of the GP is converted into an integer, given the discrete nature of the QP in H.264/SVC. The basis functions are anisotropic, unnormalized squared exponential functions[1], that is:

$$H_i\left(\mathbf{X}^{(d)}\right) = \sigma \exp\left(-\frac{1}{2}\sum_{j=1}^{4} b_j\left(X_j^{(d)} - C_{ij}\right)^2\right), \tag{4.17}$$

where σ is the size hyperparameter, $\mathbf{b} = \{b_j\}_{j=1\ldots 4}$ the length-scale hyperparameter vector; and $\mathbf{C} = \{C_{ij}\}_{i=1\ldots M, j=1\ldots 4}$, the center matrix. The squared exponential functions are probably the most widely-used ones within the kernel machines field [Rasmussen and Williams, 2006, Chapter 4] and, as shown later on, have provided good results in our experiments.

It is also worth noting that the resulting expression for the predictive mean in GP regression given in Equations (4.16) and (4.17) corresponds to that of an RBF network with one hidden layer, though the training procedure used in GP regression

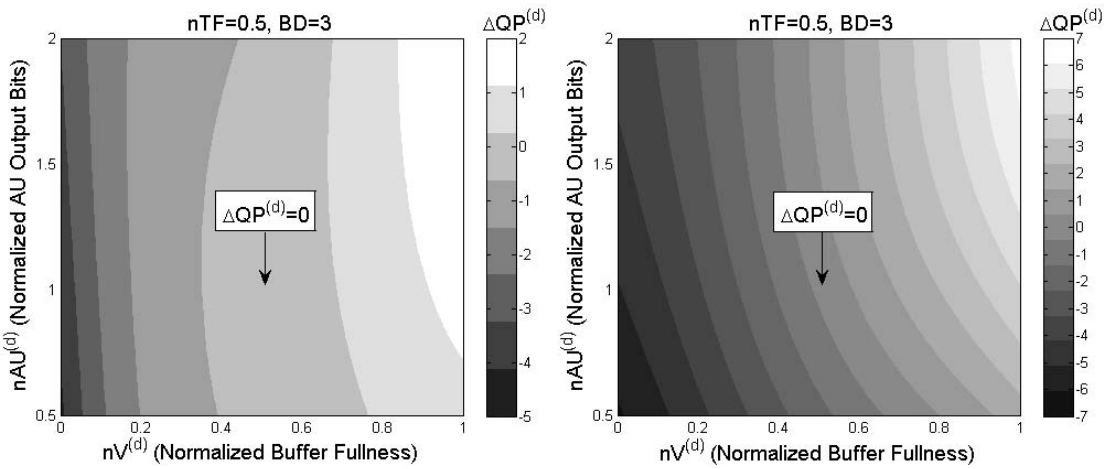
[1]Sometimes in the GP context the squared exponential function is called the Radial Basis Function (RBF) or simply Gaussian.

of course differs from the usual maximum likelihood procedures followed when training RBFs. Although other regression methods could have been employed for our purposes, GP was the preferred one because of its good performance in supervised learning.

As it will be explained in detail in Chapter 6, the training of the GP relies on a training data set containing pairs *input vector-desired output*, which have to be previously generated. Once these training data were generated, it was observed that the data distributions for the lowest temporal layer and the higher temporal layers were different enough to justify the design of two specific GPs. There were two alternatives for classifying the temporal layers into two subsets depending on in which subset the layer immediately higher than the lowest layer is considered. We decided to design one GP for K pictures (temporal base layer) and the other for non-K (NK) pictures given the notable influence of the K-picture quality on the global quality. Both QP increment models are named K and NK GPs to emphasize the dependence on the frame type.

After training the above GPs, some experiments were performed to properly dimension each regression scheme. The validation results led us to select seven basis functions in both K and NK GPs. It should be said that similar results were obtained for any higher number of basis functions.

Figure 4.3(a) shows the output of the K GP, and Figure 4.3(b) shows the output of NK GP, both for $nTF = 0.5$ and $BD = 3$. As can be observed, since the input parameters nTF and BD are set before starting the encoding process, the GP regression for $\Delta QP^{(d)}$ prediction can be seen as surfaces whose shapes depend on these constants. Several outputs are also depicted in Figures 4.4(a) and 4.4(c), for the K GP, and Figures 4.4(b) and 4.4(d), for the NK GP, for different target buffer levels and buffer sizes. Specifically, a cut of the three-dimensional surface for $nAU^{(d)} = 1$ is depicted for clarity reasons. We can see in these figures how the $\Delta QP^{(d)}$ prediction models modify their outputs according to the parameter values (nTF, BD), as should be expected to fix the VBR operating point.



(a) (b)

Figure 4.3: Output of the (a) K and (b) NK GPs for $nTF = 0.5$ and $BD = 3$.

Once the system was implemented, some unnecessary fluctuations of the QP value at NK pictures were observed, especially in cases of stationary video complexity when the buffer level approached the target buffer fullness. The problem was related to the estimation of $nAU^{(d)}$, which is normalized by a bit budget that is computed from estimated video complexities. The estimation errors in the complexities cause random short-term variations in $nAU^{(d)}$ that, in turn, produce short-term QP fluctuations in NK pictures since the output of the corresponding GP exhibits small step sizes $\Delta QP^{(d)}$ (see Figures 4.3(b), 4.4(b) and 4.4(d)). The proposed model for $\Delta QP^{(d)}$ estimation can not correct such fluctuations since the QP time evolution is not considered; in other words, the NK GP are not aware of the QP time evolution because the $\Delta QP^{(d)}$ value at the jth time instant is estimated just from the input vector at the previous time instant. In order to solve this drawback, three solutions were studied. The first of them consisted of enlarging the input vector to span a couple of time instants; however, the associated computational cost turned out to be unacceptable. The second solution consisted of filtering $nAU^{(d)}$ to smooth its noisy instantaneous fluctuations [Rezaei et al., 2008], but the encoding results were not satisfactory, especially at the scene changes. The final solution consisted of expand-

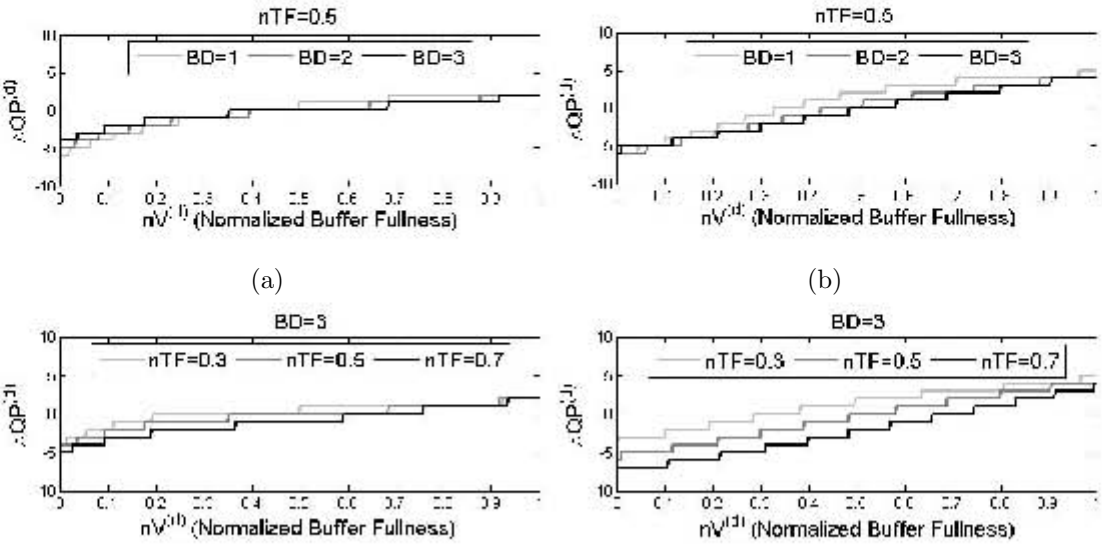


Figure 4.4: Sample outputs of the (a) K and (b) NK GPs for $nTF = 0.5$ and several values of BD; and sample outputs of the (c) K and (d) NK GPs for $BD = 3$ and several values of nTF. For the sake of clarity, only a cut of the three-dimensional surface for $nAU^{(d)} = 1$ is drawn.

ing the input region $\left(nV^{(d)}, nAU^{(d)}\right)$ for which the output is $\Delta QP^{(d)} = 0$. To this end, a simple post-processing stage of the output of the NK GP is proposed, that obeys:

$$\Delta QP^{(d)} = \begin{cases} -1 & \text{if } \Delta QP^{(d)} = -2 \\ 0 & \text{if } \Delta QP^{(d)} = -1 \\ 0 & \text{if } \Delta QP^{(d)} = 1 \\ 1 & \text{if } \Delta QP^{(d)} = 2. \end{cases} \tag{4.18}$$

This solution is used in every NK picture and provides a good trade-off between the performance in stationary video complexity and that achieved in time-varying situations. A detailed explanation about the procedure followed to design this post-processing stage is also given in Chapter 6.

4.3 Implementation Considerations

Although the complexity of the RC algorithm is negligible when compared to that of the encoding process as a whole, it deserves a brief comment. The GP-based $\Delta QP^{(d)}$ estimation can be seen as a parametric two-dimensional function, where the parameters are nTF and BD, and the inputs are $nV^{(d)}$ and $nAU^{(d)}$. Furthermore, since $\Delta QP^{(d)}$ is quantized, the output of this two-dimensional function is discrete. Therefore, if the two input variables are also quantized the function can be readily implemented as a look-up table. In summary, a look-up table can be used to implement the GP. A different look-up table should be used for each pair of parameter values (nTF, BD).

4.4 Experiments and Results

The Joint Scalable Video Model (JSVM) H.264/SVC reference software version JSVM 9.16 [Vieron et al., 2007] was used to implement the VBR controller. In order to assess its performance, it was compared to two methods: 1) constant QP (CQP) encoding[2], which can be seen as an unconstrained VBR controller [Lakshman et al., 1998], was used as a reference for nearly constant quality video; and 2) the frame level CBR control algorithm described in [Liu et al., 2008].

Following the recommendations for SVC testing conditions described in [Wien and Schwarz, 2005], both the H.264/SVC encoder and the proposed RC algorithm were configured to simulate on a PC two real-time application scenarios: mobile live streaming and IPTV broadcast. In the following subsections, both the H.264/SVC and RC configurations for each of the proposed testing scenarios are described, and then the experimental results are shown and discussed.

[2]CQP encoding means that every temporal layer within a spatial/CGS layer shares the same QP value, while the QP value of each spatial/CGS layer can be different in order to reach the pre-established target bit rate $R^{(d)}$.

4.4.1 Description of the Application Scenarios

Mobile Live Streaming

A brief description of the H.264/SVC encoder configuration for mobile live streaming is given in the following paragraphs. For a more detailed explanation of this application the reader is referred to [Schaefer et al., 2005].

A high-quality scalable bit stream that consists of a base layer and a set of enhancement layers is made available by a service provider. A mobile terminal, which can be a multimedia phone, personal digital assistant or laptop, accesses that scalable bit stream through a wireless network and decodes the sub-stream that complies with the arranged QoS. Particularly, starting out with the design suggested in [Wien and Schwarz, 2005] as reference, the following spatial/CGS encoding configuration was used:

a) Number of pictures: 900.

b) GoP size/Intra period: 8/32 pictures.

c) GoP structure: hierarchical B pictures.

d) Search range for motion estimation: 16×16 pixels.

e) Number of dependency layers: $D = 5$.

   i) $d = 0$: QCIF, $f_{out}^{(0,1)} = 6.25$ Hz $\left(T^{(0)} = 2\right)$.

   ii) $d = 1$: QCIF, $f_{out}^{(1,2)} = 12.5$ Hz $\left(T^{(1)} = 3\right)$.

   iii) $d = 2$: CIF, $f_{out}^{(2,2)} = 12.5$ Hz $\left(T^{(2)} = 3\right)$.

   iv) $d = 3$: CIF, $f_{out}^{(3,2)} = 12.5$ Hz $\left(T^{(3)} = 3\right)$.

   v) $d = 4$: CIF, $f_{out}^{(4,3)} = 25$ Hz $\left(T^{(4)} = 4\right)$.

f) Symbol mode: CAVLC at every dependency layer (as suggested in [Wiegand et al., 2009]).

The RC parameters for each dependency layer were set as follows: target buffer fullness $nTF = 50\%$, and buffer size $BD = 3$ s.

Two sets of video sequences at 25 Hz exhibiting a variety of complexities were used in our experiments. The first set consisted of four well-known test sequences [Xiph.org, 2011] recommended in [Wien and Schwarz, 2005] for streaming applications: *Bus*, *Football*, *Foreman* and *Mobile*. These sequences were concatenated to themselves several times to reach the aforementioned number of pictures. The second set consisted of three sequences displaying scene changes: *Soccer-Mobile-Foreman*, *Spiderman* (movie), and *The Lord of the Rings* (movie). *Soccer-Mobile-Foreman* was formed by concatenating 300 frames of each sequence. The other two were extracted from high-quality DVDs and downsampled to either common intermediate format (CIF) or quarter CIF (QCIF), and have been made available on-line in [Sanz-Rodríguez, 2011]. They show many scene cuts, so they are challenging from the RC point of view.

All the sequences were encoded using the set of CQP values that best approached some pre-established target bit rates. For the first group of sequences the target bit rates were those suggested in [Wien and Schwarz, 2005] for the spatial/CGS testing scenario. For the second group, the following medium-quality target bit rates were selected: 64 ($d = 0$), 96 ($d = 1$), 192 ($d = 2$), 384 ($d = 3$) and 512 kbps ($d = 4$). In all cases, the exact output bit rates obtained by CQP encoding were used as target bit rates $R^{(d,t_{max}^{(d)})}$ for both the RC algorithm in [Liu et al., 2008] and the proposed VBR controller.

IPTV Broadcast

TV broadcast through IP networks involving heterogeneous terminals (resolutions) is one of the natural fields of application for SVC [Wiegand et al., 2009]. According to both the IP network characteristics and the target IPTV set-top box definition, a wide variety of scenarios can be specified. Nevertheless, in order to define the IPTV broadcast scenario used in our experiments, we only took into consideration the dis-

play resolution and computational capabilities of the receiving devices, regardless the actual underlying type of IP network (fixed or mobile access, managed or unmanaged core). In particular, SDTV and HDTV were selected as target resolutions (emphasizing the difference with respect to those employed for the mobile live streaming scenario) for the following spatial/CGS encoding configuration:

a) Number of pictures: 500/600.

b) GoP size/Intra period: 16/16 pictures.

c) GoP structure: hierarchical B pictures.

d) Search range for motion estimation: 32×32 pixels.

e) Number of dependency layers: $D = 4$.

   i) $d = 0$: SDTV, $f_{out}^{(0,3)} = 25/30$ Hz $\left(T^{(0)} = 4\right)$.

   ii) $d = 1$: SDTV, $f_{out}^{(1,3)} = 25/30$ Hz $\left(T^{(1)} = 4\right)$.

   iii) $d = 2$: HDTV (720p), $f_{out}^{(2,4)} = 50/60$ Hz $\left(T^{(2)} = 5\right)$.

   iv) $d = 3$: HDTV (720p), $f_{out}^{(3,4)} = 50/60$ Hz $\left(T^{(3)} = 5\right)$.

f) Symbol mode: CABAC at every dependency layer.

The RC parameters for each dependency layer were set as follows: target buffer fullness $nTF = 40\%$, and buffer size $BD = 1.5$ s.

The following set of HDTV test video sequences of duration 10 s were used in our experiments [Xiph.org, 2011]: *Mobcal*_720*p*50, *Parkrun*_720*p*50, *Shields*_720*p*50 and *Stockholm*_720*p*60. They were downsampled to obtain the corresponding SDTV versions.

The criterion used to select the target bit rate for each dependency layer was that recommended in [Wien and Schwarz, 2005] for the testing scenario. The criterion suggests doubling the rate from the lowest rate point to the highest rate point for each spatial resolution, and increasing the minimum rate by a factor of four between

consecutive spatial resolutions. Thus, the following target bit rates were proposed to cover the medium-quality range: 1024 ($d = 0$), 2048 ($d = 1$), 4096 ($d = 2$) and 8192 kbps ($d = 3$).

Similarly to the mobile live streaming application, the set of CQP values that best approached the target bit rates was found, and the actual output bit rates were used as target bit rates for the two RC algorithms.

4.4.2 Experimental Results and Discussion

In order to assess the performance of the proposed VBR control algorithm from a quality point of view, the average luminance PSNR μ_{PSNR} was used. The Bjøntegaard recommendation [Bjøntegaard, 2001] was followed to compute PSNR differences with respect to CQP encoding. The average results over all the test video sequences in terms of PSNR increments $\Delta\mu_{PSNR}$ are summarized in Tables 4.1 and 4.2 for the mobile live streaming and IPTV broadcast scenarios, respectively. Two rows per spatial/CGS layer are shown, one for [Liu et al., 2008] and the other for the proposed method. As can be observed, the performance achieved by the VBR controller in terms of average PSNR was similar to that of CQP encoding, and notably superior to that of [Liu et al., 2008]. Furthermore, the good results achieved by the proposed method at the second and third enhancement layers in the IPTV broadcast scenario (see rows $d = \{2, 3\}$ in Table 4.2) deserve a special comment. These layers correspond to HDTV sequences and no samples of HDTV sequences were used for training. Therefore, these results prove that the GP regression models for $\Delta QP^{(d)}$ estimation generalize properly and are able to work well for any resolution.

Tables 4.3 and 4.4 show a detailed comparison of the three assessed algorithms for two representative video sequences. *The Lord of the Rings*, taken from the mobile live streaming scenario, is a good example of non-stationary video complexity. On the other hand, *Stockholm*, from the IPTV broadcast scenario, is an example of stationary video complexity. The experiments were conducted using the following target bit rates: 66.47 (d = 0), 97.32 (d = 1), 189.47 (d = 2), 388.07 (d = 3) and

Layer d	Algorithm	$\Delta\mu_{PSNR}$ (dB)	$\Delta\overline{\sigma}_{PSNR,j}$ (dB)	Bit Rate Error (%)	#O/#U	μ_V (%)
0	[Liu et al., 2008]	-0.19	0.41	1.87	8/0	57.42
	Proposed	-0,13	0.12	0.93	0/0	52.45
1	[Liu et al., 2008]	-0.43	0.75	1.35	9/0	57.29
	Proposed	-0,14	0.14	1.25	0/0	59.30
2	[Liu et al., 2008]	-0.33	0.35	0.68	6/0	54.91
	Proposed	-0,10	0.05	0.87	0/0	53.41
3	[Liu et al., 2008]	-0.20	0.36	0.44	0/0	52.81
	Proposed	-0,07	0.05	0.69	0/0	52.81
4	[Liu et al., 2008]	-0.46	0.51	0.30	0/0	53.45
	Proposed	-0,07	0.06	0.90	0/0	57.29

Table 4.1: Average results achieved by both the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for the mobile live streaming scenario. Incremental results are given with respect to CQP encoding.

500.56 kbps (d = 4) for *The Lord of the Rings*; and 975.92 (d = 0), 1885.90 (d = 1), 4209.83 (d = 2) and 7331.63 kbps (d = 3) for *Stockholm*. The analysis of these results allowed us to draw two main conclusions: 1) for non-stationary complexity sequences, the performance of the proposed method was remarkably good, exceeding even that of the nearly constant quality system at some dependency layers; and 2) for stationary complexity sequences, the performance of the proposed method was quite close to that of the nearly constant quality system.

Representative behaviors of the encoder buffer level, PSNR and QP time evolutions are depicted in Figures 4.5(a) and 4.5(b) for *The Lord of the Rings*, and in Figures 4.6(a) and 4.6(b) for *Stockholm*. Specifically, Figures 4.5(a) and 4.6(a) correspond to the spatial base layer ($d = 0$) and Figures 4.5(b) and 4.6(b) correspond to the third enhancement layer ($d = 3$). When compared to [Liu et al., 2008], the

Layer d	Algorithm	$\Delta\mu_{PSNR}$ (dB)	$\Delta\overline{\sigma}_{PSNR,j}$ (dB)	Bit Rate Error (%)	#O/#U	μ_V (%)
0	[Liu et al., 2008]	-0.07	0.70	0.57	0/0	49.77
	Proposed	-0.11	0.31	1.86	0/0	38.16
1	[Liu et al., 2008]	-0.52	0.45	0.41	0/0	46.55
	Proposed	-0.15	0.26	1.99	0/0	35.80
2	[Liu et al., 2008]	-0.74	0.25	0.31	0/0	45.42
	Proposed	0.06	0.16	1.77	0/0	37.43
3	[Liu et al., 2008]	-0.40	0.20	0.14	0/0	44.16
	Proposed	0.06	0.20	1.43	0/0	35.14

Table 4.2: Average results achieved by both the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for the IPTV broadcast scenario. Incremental results are given with respect to CQP encoding.

proposed VBR controller made better use of the buffer to provide PSNR and QP time evolutions closer to those of the nearly constant quality system. Furthermore, in the non-stationary scenario, the strong correlation among buffer occupancy, PSNR time evolution, and QP time evolution reveals that the proposed method made a proper use of the buffer to successfully allocate larger amounts of bits for more complex scenes, and vice versa. Consequently, the potential quality fluctuation of the compressed video was kept low, in particular at the scene changes (see, for example, the PSNR time evolution around pictures #260 and #703). It is also worth noting that the proposed method did an excellent work on minimizing unnecessary changes in QP time evolution, which is our main design goal; particularly, in the stationary scenario, it was able to provide a performance close to that of the nearly constant quality system. In terms of PSNR time evolution, the results were not so good for some sequences, such as *Stockholm* (Figures 4.6(a) and 4.6(b)). In these cases, the GoP-periodic PSNR leaps are due to large R-D differences between K and NK

Layer d	Algorithm	μ_{PSNR} (dB)	$\overline{\sigma}_{PSNR,j}$ (dB)	Bit Rate Error (%)	#O/#U	μ_V (%)
0	CQP	34.45	0.66	-	42/48	49.76
	[Liu et al., 2008]	33.14	1.10	3.82	55/0	78.04
	Proposed	34.35	0.90	1.57	0/0	53.46
1	CQP	34.39	0.67	-	100/107	46.90
	[Liu et al., 2008]	33.19	2.05	1.72	66/0	69.65
	Proposed	34.30	0.97	1.93	0/0	58.08
2	CQP	32.88	0.91	-	96/111	47.15
	[Liu et al., 2008]	32.26	1.51	0.30	40/0	63.69
	Proposed	32.80	1.09	1.93	0/0	52.19
3	CQP	35.24	0.82	-	92/114	45.22
	[Liu et al., 2008]	35.43	1.31	1.26	0/0	52.99
	Proposed	35.33	0.97	1.57	0/0	52.97
4	CQP	35.14	0.82	-	205/237	45.58
	[Liu et al., 2008]	34.86	1.57	1.00	0/0	53.82
	Proposed	35.23	0.98	2.43	0/0	63.35

Table 4.3: Performance comparison between the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for a specific non-stationary complexity video sequence, *The Lord of the Rings*. The results achieved by CQP encoding have also been included for reference.

pictures. As can be observed, this behavior also happens in CQP encoding whose performance we intend to meet.

In order to assess the performance of the VBR control algorithm from the quality consistency point of view, a time-local version of the PSNR standard deviation was computed. This local PSNR standard deviation attempts to measure the quality consistency within a scene by reducing the impact of the scene changes on the PSNR

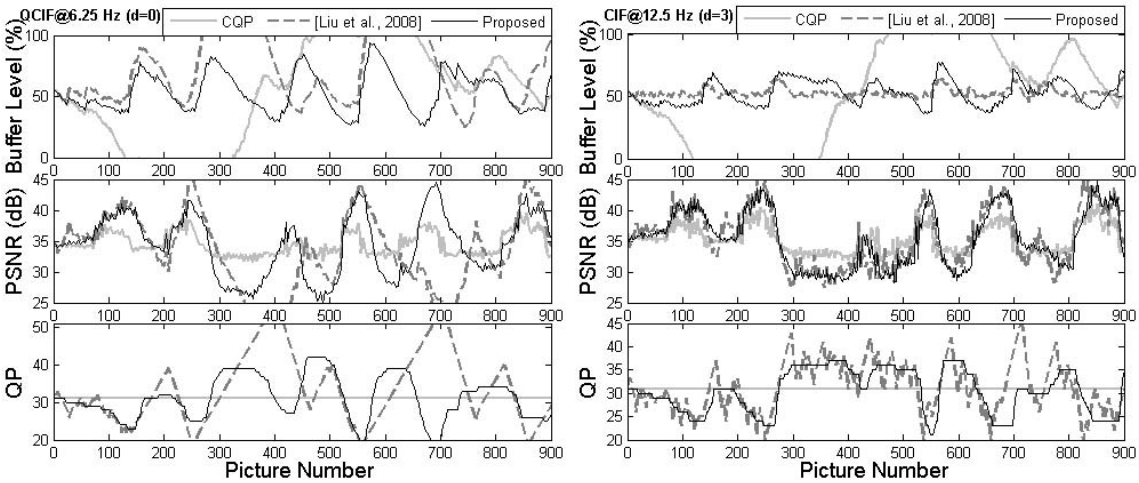
Layer d	Algorithm	$\mu_{\mathbf{PSNR}}$ (dB)	$\overline{\sigma}_{\mathbf{PSNR,j}}$ (dB)	Bit Rate Error (%)	#O/#U	$\mu_{\mathbf{V}}$ (%)
0	CQP	35.54	0.20	-	0/0	50.31
	[Liu et al., 2008]	35.47	0.91	0.80	0/0	49.48
	Proposed	35,53	0.34	-1.64	0/0	37.20
1	CQP	38.60	0.14	-	0/0	50.55
	[Liu et al., 2008]	37.94	0.54	0.21	0/0	46.14
	Proposed	38,58	0.26	-1.89	0/0	35.36
2	CQP	34.18	0.18	-	0/0	43.30
	[Liu et al., 2008]	33.60	0.34	0.29	0/0	45.10
	Proposed	34,27	0.23	-1.88	0/0	35.59
3	CQP	34.93	0.25	-	0/0	40.71
	[Liu et al., 2008]	34.53	0.32	0.15	0/0	43.91
	Proposed	34.98	0.32	-1.17	0/0	33.95

Table 4.4: Performance comparison between the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for a specific stationary complexity video sequence, *Stockholm*. The results achieved by CQP encoding have also been included for reference.

standard deviation. Thus, small local PSNR standard deviations indicate smooth short-term PSNR fluctuations and, therefore, high quality consistency. In particular, the local PSNR standard deviation was computed over a time-window as follows:

$$\sigma_{PSNR,j} = \sqrt{\frac{1}{W}\sum_{i=j-W/2}^{j+W/2-1}\left(PSNR_i - \mu_{PSNR,W}\right)^2}, \tag{4.19}$$

where W denotes the time-window size (in number of pictures), and $\mu_{PSNR,W}$ the average PSNR for a given window size. In particular, W was set to $2^{T^{(d)}}$ pictures in our experiments, which is a time interval short enough to minimize the influence of PSNR leaps at the scene changes. Finally, in order to summarize the results in an



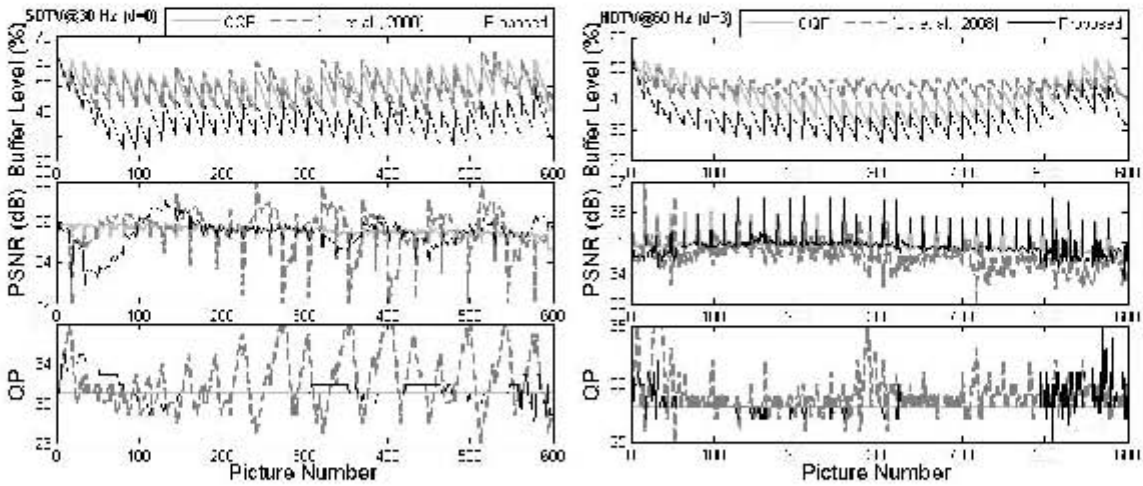
(a) (b)

Figure 4.5: Encoder buffer level, PSNR and QP time evolutions corresponding to (a) the spatial base layer ($d = 0$) and (b) the third enhancement layer ($d = 3$) from *The Lord of the Rings*. High-quality plots corresponding to every spatial/CGS layer are available on-line in [Sanz-Rodríguez, 2011].

unique measurement, the mean value of the local PSNR standard deviation, denoted as $\overline{\sigma}_{PSNR,j}$, was computed.

Additionally, it should be noticed that, since the local PSNR standard deviation does not take into account any buffer constraint, CQP encoding provided a smaller local PSNR standard deviation (see Figures 4.5(a) and 4.5(b)). Obviously, this smaller local PSNR standard deviation was in exchange for high instantaneous bit rate variations at the scene changes that are not allowed in a constrained buffer scenario. The results in terms of $\overline{\sigma}_{PSNR,j}$ increment with respect to CQP encoding, $\Delta\overline{\sigma}_{PSNR,j}$, are provided in Tables 4.1 and 4.2. As can be observed, the proposed VBR controller achieved better quality consistency than that of the RC algorithm in [Liu et al., 2008]. Furthermore, the results, especially at the higher dependency layers, were remarkably close to those of CQP encoding, in spite of the buffer constraint.

The proposed VBR controller was also assessed in terms of target bit rate ad-



(a) (b)

Figure 4.6: Encoder buffer level, PSNR and QP time evolutions corresponding to (a) the spatial base layer ($d = 0$) and (b) the third enhancement layer ($d = 3$) from *Stockholm*. High-quality plots corresponding to every spatial/CGS layer are available on-line in [Sanz-Rodríguez, 2011].

justment and mean buffer level. In particular, its performance was comparatively evaluated by computing the output bit rate error, the number of pictures in which either an overflow (#O) or an underflow (#U) occurred, and the mean buffer level, μ_V. As can be observed in Tables 4.1 – 4.4, both the RC scheme in [Liu et al., 2008] and the VBR control algorithm provided in most cases output bit rate differences below 2%, which is the maximum bit rate error recommended in [Wien and Schwarz, 2005] for the spatial/CGS testing scenario. The average results in terms of μ_V achieved by the proposed method were close to the target buffer fullness, thus proving a good long-term adaptation to the target bit rate at each dependency layer. Furthermore, the results in terms of #O and #U revealed that the VBR controller was able to significantly reduce both the overflow and underflow risks in sequences with scene changes, such as *The Lord of the Rings*. The poor performance of the RC algorithm in [Liu et al., 2008] at the scene changes was due to the lack of a specif

mechanism to deal with such events. The use of a scene change detector would be helpful to improve its performance in such cases.

Finally, from the complexity point of view, the central processing unit (CPU) time consumed by the proposed VBR controller and the RC scheme in [Liu et al., 2008] were measured by means of a high-resolution performance counter. In order to minimize the measurement error caused by occasional multi-task operations, each sequence was encoded five times and the minimum CPU time was selected for the complexity analysis (nevertheless, it is worth mentioning that the variance of the measured CPU times was very small). The complexity results using an Intel Core2 Duo CPU E8400@3.0 GHz are given in Table 4.5 for the mobile live streaming scenario and in Table 4.6 for the IPTV broadcast scenario. As can be observed, the RC algorithm in [Liu et al., 2008] consumed an average CPU time per AU of 239 μs for the mobile live streaming scenario and 2071 μs for the IPTV broadcast scenario, while our proposal only consumed 26 μs and 33 μs, respectively. These differences in terms of complexity between both algorithms are mainly due to the R-D model employed by the CBR controller in [Liu et al., 2008]. This RC algorithm, which follows the usual approach in H.264/AVC [Ma et al., 2003], first estimates the frame complexity and subsequently the desired QP value. The QP estimation model relies on a linear regression that is computationally heavier than the proposed GPs. Furthermore, the frame complexity measurement requires performing simple operations on the whole picture, what explains the significant CPU time increment that happens in the IPTV broadcast scenario (which operates on larger pictures).

Furthermore, as previously described in Section 4.3, the complexity of the GP-based $\Delta QP^{(d)}$ estimation model can be reduced even more by means of a look-up table-based implementation. In particular, preliminary experiments using 10×8 $\left(nV^{(d)} \times nAU^{(d)}\right)$ look-up tables for $\Delta QP^{(d)}$ estimation were conducted, achieving nearly equivalent results. Therefore, the proposed GPs can be successfully implemented using look-up tables.

Sequence	CPU Time (μs)	
	[Liu et al., 2008]	Proposed
Bus	211355	23658
Football	221029	22555
Foreman	220253	23793
Mobile	209543	23149
Average	215545	23289
Average per AU	239	26

Table 4.5: CPU time comparison between the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for the mobile live streaming scenario using an Intel Core2 Duo CPU E8400@3.0 GHz.

Sequence	CPU Time (μs)	
	[Liu et al., 2008]	Proposed
Mobcal	1065523	16349
Parkrun	1038447	17061
Shields	1049664	16758
Stockholm (first 500 pictures)	988212	16550
Average	1035462	16679
Average per AU	2071	33

Table 4.6: CPU time comparison between the RC algorithm in [Liu et al., 2008] and the proposed VBR controller for the IPTV broadcast scenario using an Intel Core2 Duo CPU E8400@3.0 GHz.

4.5 Summary and Conclusions

In this chapter a novel VBR controller for real-time H.264/SVC video coding applications has been described. The proposed VBR controller aims to improve the quality

consistency by preventing unnecessary QP fluctuations. The proper estimation of the incremental variation of QP at each dependency layer is computed by means of two GPs, one for K pictures and the other for NK pictures that have been specially designed for this purpose. This approach offers the additional advantage of not using any analytic R-D model for QP estimation, so the chicken and egg dilemma for frame complexity estimation is no longer a concern. Furthermore, the input vector to the GPs has been enlarged with two additional constant parameters to provide an effective solution for a wide range of both target buffer fullness and buffer size.

Two real-time application scenarios were simulated to assess the performance of the VBR controller, which was compared to both CQP encoding, as a reference for nearly constant quality, and a recently proposed CBR controller for H.264/SVC [Liu et al., 2008]. For stationary complexity sequences, the average quality achieved by the VBR controller was quite close to that of the nearly constant quality system (the time evolution of QP was maintained almost constant in time). For non-stationary complexity sequences, the average quality of the proposed algorithm was remarkably good, exceeding even that of the nearly constant quality system at some dependency layers, since it was able to allocate larger amounts of bits for more complex scenes, and vice versa.

In terms of quality consistency, the performance of the VBR controller was significantly better than that of the CBR controller in [Liu et al., 2008]. Furthermore, the experimental results, especially at the higher dependency layers, were remarkably close to those of CQP encoding, in spite of the buffer constraint. With respect to the overflow and underflow risks, again the results revealed that the VBR control algorithm was notably superior. From the complexity point of view, the proposed method notably outperformed the RC scheme in [Liu et al., 2008].

To sum up, the VBR controller described in this chapter achieved an excellent performance in terms of average quality, quality consistency, long-term adjustment to the target rate, and buffer overflow and underflow prevention at each dependency layer, with low complexity.

Chapter 5

In-Layer Multi-Buffer Framework for Rate-Controlled SVC

The RC algorithms proposed in the literature for SVC only guarantee the HRD requirement for the highest temporal layer at every dependency layer. Therefore, the temporal scalability is not fully exploited since, in order to deliver HRD-compliant sub-streams, it is necessary to increase the number of dependency layers. For instance, if a video transmission service offered the same QoS to two target decoders with identical spatial resolutions but different temporal resolutions, the SVC encoder would have to use two CGS layers, one per temporal layer. Although the two desired HRD-compliant sub-streams are provided, the temporal scalability is underused since each one of the highest temporal layers actually also contains the lower frame rate. In summary, the common SVC encoder configuration for rate-controlled video may incur in redundant dependency layers, producing an unnecessary increase of bit rate and coding complexity.

In this chapter we describe a novel RC approach for delivering more than one HRD-compliant temporal resolution within a particular dependency layer. Specifically, the proposed method uses a set of virtual buffers (one per HRD-compliant temporal layer) within a dependency layer, so that the buffer levels can be simul-

taneously controlled for overflow and underflow prevention, while minimizing the reconstructed video distortion of the corresponding sub-streams. This in-layer multi-buffer (IL-MB) framework has been built on top of the VBR controller described in the previous chapter, which, according to the new nomenclature, will be referred to as IL single-buffer (IL-SB) RC algorithm. Alternatively, in this chapter, we will also refer to this IL-SB RC algorithm as *baseline* RC algorithm.

The chapter is organized as follows. In Section 5.1 a general description of the RC scheme for IL-MB control is given. In Section 5.2 a detailed description the rate controller located at each dependency layer is provided, making special emphasis on the *buffer modeling* stage, which is used to properly manage the set of virtual buffers. Section 5.3 describes the experimental setup and reports and discusses the experimental results. Finally, in Section 5.4 the main conclusions are summarized.

5.1 System Overview

The proposed VBR control scheme is illustrated in Figure 5.1. For clarity reasons, only the dependency base layer ($d = 0$) of the SVC encoder is shown. The blocks depicted in dark gray are the extensions required by the baseline VBR controller shown in Figure 4.1 to become an IL-MB controller.

Each dependency layer d involves a rate controller $RC^{(d)}$ and a set of virtual buffers (except to that corresponding to the complete scalable bit stream that is a real buffer). Each of these virtual buffers simulates the encoder buffering process of the sub-stream corresponding to certain temporal resolution. In order to formulate properly the IL-MB controller, a parameter $t_{min}^{(d)}$ is introduced that indicates which of those temporal resolution sub-streams from $(d, 0)$ to $(d, t_{max}^{(d)})$ should comply with the HRD constraints. Specifically, when $t_{min}^{(d)} = t_{max}^{(d)}$, the proposed IL-MB RC scheme becomes the baseline VBR algorithm.

In order to make the explanation of the IL-MB model easier, let us follow the example illustrated in Figure 5.1. In particular, the input video is a QCIF video

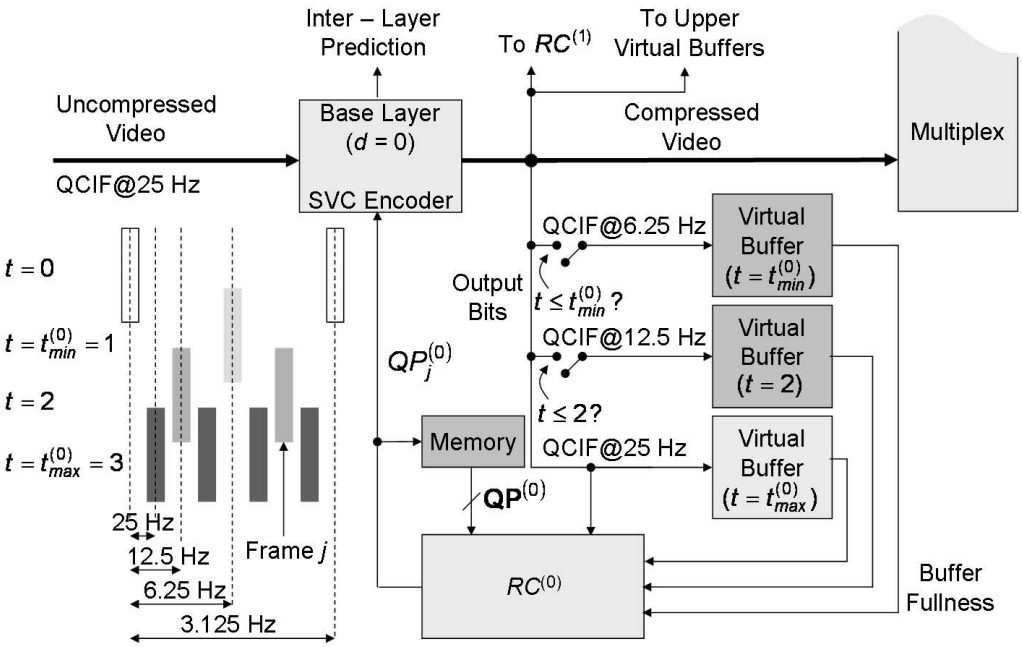


Figure 5.1: Block diagram of the proposed H.264/SVC RC scheme for IL-MB control. Only the spatial base layer is depicted for the sake of clarity.

sequence at 25 Hz using a GoP size of 8 pictures, so that encoded video from QCIF@3.125 Hz to QCIF@25 Hz can be provided. Setting $t_{min}^{(0)} = 1$ means that the three higher temporal resolution sub-streams $(0, 1)$, $(0, 2)$, and $(0, 3)$ should be HRD-compliant and, consequently, their corresponding virtual buffers should be controlled for proper video content delivery. For the lowest temporal resolution sub-stream, however, the HRD compliance would not be guaranteed.

Following with the example, when the jth picture with layer identifier $(0, 2)$ (see Figure 5.1) is going to be encoded, the goal of the the rate control module $RC^{(0)}$ is to provide an appropriate $QP_j^{(0)}$ value, so that the set of virtual buffers involved are maintained at secure levels. Specifically, the set of virtual buffers involved in the encoding of jth picture with layer identifier $(0, 2)$ are:

$$\mathbf{V}^{(0,2)} = \left\{ V^{(0,k)} \right\}_{k=\max\left[t_{min}^{(0)},2\right]\ldots t_{max}^{(0)}},$$

where $V^{(0,k)}$ denotes the buffer fullness associated with the sub-stream $(0, k)$, with

$k = \max\left[t_{min}^{(0)}, 2\right] \ldots t_{max}^{(0)}$. It should be noticed that, since $t_{min}^{(0)} = 1$, the lowest k value is 2 and, therefore, the two higher virtual buffers are updated. However, if the picture belonged to a temporal layer lower than or equal to $t_{min}^{(0)}$, the three virtual buffers would be updated. From now on, we will refer to the virtual buffers to be updated at the jth time instant as *involved buffers*.

It is also worth mentioning that all the involved buffers must be taken into account to estimate the current QP value, since a proper behavior is not guaranteed in all of them otherwise. Thus, the method for properly controlling any set $\mathbf{V}^{(d,t)}$ becomes the main focus of the proposed IL-MB VBR controller.

The rate controller $RC^{(d)}$, similarly to what was described for the baseline RC approach, obtains a reference QP, $QP_{REF}^{(d)}$, estimates a $\Delta QP^{(d)}$ value, and finally computes the desired $QP_j^{(d)}$ as follows:

$$QP_j^{(d)} = QP_{REF}^{(d)} + \Delta QP^{(d)}, \tag{5.1}$$

The reference QP is computed from those QPs used for the encoding of the last pictures belonging to the sub-streams $(d, t_{min}^{(d)})$ to $(d, t_{max}^{(d)})$ (see Subsection 5.2.2 for details). This set of previous QPs, defined as

$$\mathbf{QP}^{(d)} = \left\{QP^{(d,k)}\right\}_{k=t_{min}^{(d)} \ldots t_{max}^{(d)}},$$

is updated on a frame basis according to the involved buffers at the jth time instant, as described in Algorithm 1.

Algorithm 1 $\mathbf{QP}^{(d)}$ updating procedure.

1. **for** $k = \max\left[t_{min}^{(d)}, t\right]$ to $t_{max}^{(d)}$ **do** {involved buffers}
2. $\quad QP^{(d,k)} \leftarrow QP_j^{(d)}$
3. **end for**

It should be noticed that the storage of this set of QPs requires a memory block (see Figure 5.1) that was not necessary in the baseline approach (see Figure 4.1), where there was just a delay line to make previous QP value available.

The QP increment is selected to provide a slow QP variation so that the visual quality consistency is improved. Similarly to what was described for the baseline VBR control algorithm, the following input parameters are required to compute $\Delta QP^{(d)}$:

1) The current fullness of the virtual buffers $(d, t_{min}^{(d)})$ to $(d, t_{max}^{(d)})$.

2) The amount $AU^{(d,t)}$ of AU output bits.

In the following section, a detailed description of the RC module for IL-MB control at a specific dependency layer is given.

5.2 RC Stages

The MB-based rate controller $RC^{(d)}$ is illustrated in Figure 5.2. The estimation of $QP_j^{(d)}$ is performed in three stages, namely: *parameter updating*, *buffer modeling* and *GP-based QP increment estimation*, which are described in more detail through the next subsections.

5.2.1 Parameter Updating

After encoding the $(j-1)$th picture with layer identifier (d, t'), two parameter sets, required to estimate $\Delta QP^{(d)}$, should be updated: 1) the normalized versions of the buffer levels $(d, t_{min}^{(d)})$ to $(d, t_{max}^{(d)})$, denoted as $\mathbf{nV}^{(d)}$; and 2) the normalized versions of $AU^{(d,t')}$ for the sub-streams $(d, t_{min}^{(d)})$ to $(d, t_{max}^{(d)})$, denoted as $\mathbf{nAU}^{(d)}$. These parameter sets are defined as follows:

$$\mathbf{nV}^{(d)} = \left\{ \frac{V^{(d,k)}}{BS^{(d,k)}} \right\}_{k=t_{min}^{(d)} \ldots t_{max}^{(d)}},$$

$$\mathbf{nAU}^{(d)} = \left\{ \frac{AU^{(d,t')}}{G^{(d,t',k)}} \right\}_{k=t_{min}^{(d)} \ldots t_{max}^{(d)}},$$

where $BS^{(d,k)}$ is the buffer size in bits for the sub-stream (d, k), and $G^{(d,t',k)}$ the AU target bits at the layer (d, t') to satisfy the target bit rate $R^{(d,k)}$.

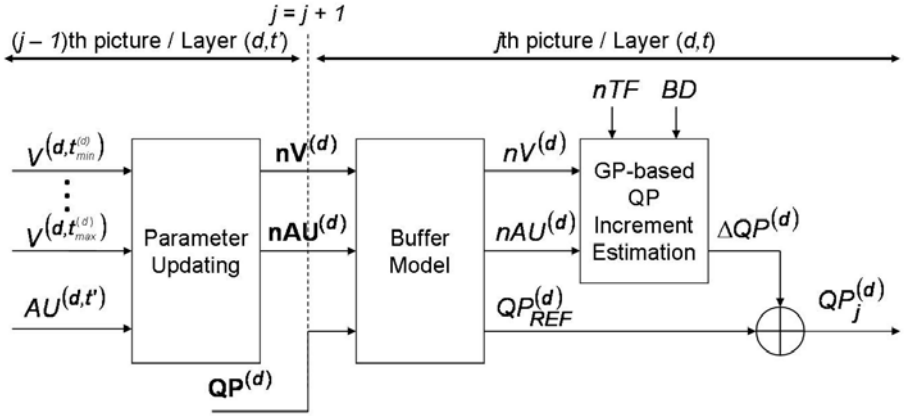


Figure 5.2: Block diagram of the MB-based rate controller module $RC^{(d)}$ for a specific dependency layer d.

These updating equations require the previous update of the involved buffers $\mathbf{V}^{(d,t')}$ and the estimation of the set of AU target bits $\{G^{(d,t',k)}\}$. In turn, the update of the set of buffers requires to obtain the AU output bits $AU^{(d,t')}$, and the estimation of the set of AU target bits requires the previous update of average texture and motion complexities for each temporal layer u from 0 to $t_{max}^{(d)}$, $\overline{C}_{TEX}^{(d,u)}$ and $\overline{C}_{MOT}^{(d,u)}$, respectively.

The virtual buffer levels, the AU output bits, the AU target bits, as well as the average texture and motion complexities are updated as in Subsection 4.2.1, but replacing $t_{max}^{(d)}$ by the index k, which takes values from $t_{min}^{(d)}$ to $t_{max}^{(d)}$. Algorithm 2 summarizes the complete updating procedure for $\mathbf{nV}^{(d)}$ and $\mathbf{nAU}^{(d)}$.

5.2.2 Buffer Modeling

In this stage three parameters required to estimate the QP value are computed. These parameters are representative values of the sets $\mathbf{QP}^{(d)}$ (Algorithm 1), $\mathbf{nV}^{(d)}$, and $\mathbf{nAU}^{(d)}$ (Algorithm 2), which are denoted as $QP_{REF}^{(d)}$, $nV^{(d)}$, and $nAU^{(d)}$, respectively. The first parameter is used as reference QP in Equation (5.1), while the last two are required for $\Delta QP^{(d)}$ estimation.

Algorithm 2 $\mathbf{nV}^{(d)}$ and $\mathbf{nAU}^{(d)}$ updating procedure

1. Compute $AU^{(d,t')}$ (4.5)
2. Update $\overline{C}_{TEX}^{(d,t')}$ (4.7)
3. Update $\overline{C}_{MOT}^{(d,t')}$ (4.8)
4. **for** $k = \max\left[t_{min}^{(d)}, t'\right]$ to $t_{max}^{(d)}$ **do** {involved buffers}
5. Update $V^{(d,k)}$ (4.6)
6. Compute $G^{(d,t',k)}$ (4.9), (4.10), (4.11), (4.12)
7. $nV^{(d,k)} \leftarrow \max\left[0, \min\left[\frac{V^{(d,k)}}{BS^{(d,k)}}, 1\right]\right]$
8. $nAU^{(d,k)} \leftarrow \max\left[\frac{1}{2}, \min\left[\frac{AU^{(d,t')}}{G^{(d,t',k)}}, 2\right]\right]$
9. **end for**

The buffer modeling algorithm suggested for the estimation of the aforementioned values is made up of several decision rules that are described next. If none of the involved buffer levels is close to overflow or underflow, then $nV^{(d)}$, $nAU^{(d)}$ and $QP_{REF}^{(d)}$ are computed as the arithmetic average of $\mathbf{nV}^{(d)}$, $\mathbf{nAU}^{(d)}$, and $\mathbf{QP}^{(d)}$, respectively. Otherwise, only the parameters coming from that temporal resolution showing the most critical buffer fullness is considered. Nevertheless, given that more than one involved buffer fullness could be considered as critical at a certain time instant, the following precedence rules have been established (relying on certain observations about the time evolution of the virtual buffers for a variety of video sequences):

1) Since the overflow risk is more likely than the underflow risk, especially when encoding I pictures, the overflow risk is given precedence in each involved buffer.

2) Since the buffer of the lowest temporal resolution usually exhibits the largest fluctuations and, therefore, the highest overflow and underflow risks (since its buffer size in bits is the smallest for a given buffer delay), the involved buffer levels are given precedence according to their temporal layer identifier.

The pseudocode given in Algorithm 3 summarizes the proposed buffer modeling process.

Algorithm 3 $nV^{(d)}$, $nAU^{(d)}$ and $QP_{REF}^{(d)}$ updating procedure

1. $nV^{(d)} = nAU^{(d)} = QP_{REF}^{(d)} = 0$
2. **for** $k = \max\left[t_{min}^{(d)}, t\right]$ to $t_{max}^{(d)}$ **do** {involved buffers}
3. **if** $nV^{(d,k)} \geq 0.8$ **then** {overflow risk}
4. $nV^{(d)} \leftarrow nV^{(d,k)}$
5. $nAU^{(d)} \leftarrow nAU^{(d,k)}$
6. $QP_{REF}^{(d)} \leftarrow QP^{(d,k)}$
7. **break for**
8. **else if** $nV^{(d,k)} \leq 0.2$ **then** {underflow risk}
9. $nV^{(d)} \leftarrow nV^{(d,k)}$
10. $nAU^{(d)} \leftarrow nAU^{(d,k)}$
11. $QP_{REF}^{(d)} \leftarrow QP^{(d,k)}$
12. **break for**
13. **else** {secure level}
14. $nV^{(d)} \leftarrow nV^{(d)} + nV^{(d,k)}$
15. $nAU^{(d)} \leftarrow nAU^{(d)} + nAU^{(d,k)}$
16. $QP_{REF}^{(d)} \leftarrow QP_{REF}^{(d)} + QP^{(d,k)}$
17. **if** $k = t_{max}^{(d)}$ **then** {all buffers at secure levels}
18. $nV^{(d)} \leftarrow \frac{nV^{(d)}}{t_{max}^{(d)} - \max\left[t_{min}^{(d)}, t\right] + 1}$
19. $nAU^{(d)} \leftarrow \frac{nAU^{(d)}}{t_{max}^{(d)} - \max\left[t_{min}^{(d)}, t\right] + 1}$
20. $QP_{REF}^{(d)} \leftarrow \text{round}\left[\frac{QP_{REF}^{(d)}}{t_{max}^{(d)} - \max\left[t_{min}^{(d)}, t\right] + 1}\right]$
21. **end if**
22. **end if**
23. **end for**

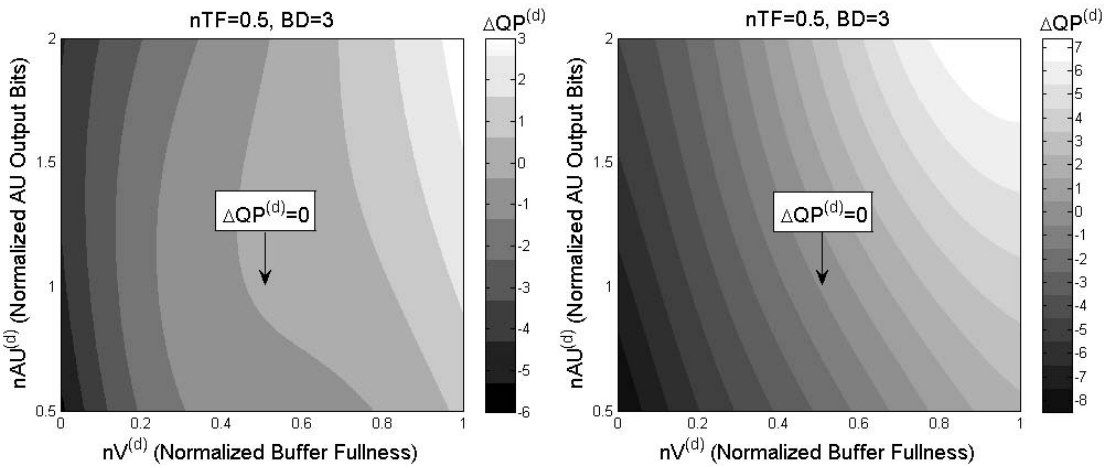
It is worth noticing that, although the given description of the buffer modeling stage is tied to the baseline RC algorithm formulation, the underlying ideas might be adapted to any other RC algorithm for SVC in order to obtain the proper values of the required parameters for QP estimation.

5.2.3 GP-Based QP Increment Estimation

As in the baseline RC scheme, the four-dimensional input vector given in Equation (4.15) is fed into GP to produce a $\Delta QP^{(d)}$ estimation. Actually, two different GPs are used, one for K pictures and the other for NK pictures. The architecture of each GP is the same than that given in Equations (4.16) and (4.17); however, the GP parameters must be specifically trained to cope with the proposed IL-MB model, where the buffer and distortion constraints for QP selection are tougher; in particular, the GP parameters should be chosen to properly deal with the fact that several buffers have to be simultaneously controlled within a dependency layer.

In order to find the most suitable GP parameters, a training data set was previously generated. Subsequently, the training and parameter selection processes were performed. To this end, the same methodology as that followed to design the GPs of the baseline RC algorithm for IL-SB control (see Chapter 6 for details) was used for the IL-MB case, but modifying the cost function for data labeling to provide a good trade-off between the control of the involved buffers and the quality consistency of the corresponding sub-streams.

The training and validation results led us to select ten Gaussian-type functions for both both K-picture MB (K-MB) and NK-picture MB (NK-MB) GPs. Figure 5.3(a) shows the output of K-MB GP, and Figure 5.3(b) shows the output of the NK-MB GP, both of them for $nTF = 0.5$ and $BD = 3$. When compared these GPs to those for SB rate control (see Figures 4.3(a) and 4.3(b)), the two MB GPs are more sensitive to the variation in $nAU^{(d)}$, especially the GP designed for NK pictures. It means that, for a given buffer fullness $nV^{(d)}$, a noticeable target bit rate mismatch would imply a larger $\Delta QP^{(d)}$ value with respect to that provided by the



(a) (b)

Figure 5.3: Output of the (a) K-MB and (b) NK-MB GPs for $nTF = 0.5$ and $BD = 3$.

SB GPs. Another characteristic of the QP increment models for MB rate control is related to the larger $\Delta QP^{(d)}$ ranges that are capable of achieving. In short, these differences between both types of GPs due to buffering process allow us to conclude that the MB GPs generate more conservative outputs because of a larger number of buffers to be controlled simultaneously.

Finally, the post-processing stage of the output of the NK-MB GP (see Equation (4.18)) is also performed in order to reduce unnecessary QP fluctuations.

5.3 Experiments and Results

The JSVM H.264/SVC reference software version JSVM 9.16 [Vieron et al., 2007] was also used to implement the proposed IL-MB VBR controller. Its performance was compared to other two methods: 1) CQP encoding, which is used as a reference for nearly constant quality video; and 2) our baseline VBR controller, which can be seen as a particular case of the proposed method when $t_{min}^{(d)} = t_{max}^{(d)}$ for every dependency layer.

In the following subsections, the SVC encoder and RC configurations employed for comparisons are described, the experimental results are given, and a discussion concerning these results is provided.

5.3.1 Description of the SVC Encoder and RC Configurations

According to the SVC testing conditions recommended in [Wien and Schwarz, 2005], the mobile live streaming scenario described in Subsection 4.4.1 was used to assess the performance of the aforementioned algorithms. In particular, the following five-dependency layer H.264/SVC encoder configuration was used for the baseline VBR controller:

a) Number of pictures: 900.

b) GoP size/Intra period: 8/32 pictures.

c) GoP structure: hierarchical B pictures.

d) Search range for motion estimation: 16×16 pixels.

e) Number of dependency layers: $D = 5$.

   i) $d = 0$: QCIF, $f_{out}^{(0,1)} = 6.25$ Hz $\left(T^{(0)} = 2\right)$.

   ii) $d = 1$: QCIF, $f_{out}^{(1,2)} = 12.5$ Hz $\left(T^{(1)} = 3\right)$.

   iii) $d = 2$: CIF, $f_{out}^{(2,2)} = 12.5$ Hz $\left(T^{(2)} = 3\right)$.

   iv) $d = 3$: CIF, $f_{out}^{(3,2)} = 12.5$ Hz $\left(T^{(3)} = 3\right)$.

   v) $d = 4$: CIF, $f_{out}^{(4,3)} = 25$ Hz $\left(T^{(4)} = 4\right)$.

f) Symbol mode: CAVLC.

The RC parameters for each dependency layer were set as follows: target buffer fullness $nTF = 50\%$, and buffer size $BD = 3$ s. Henceforth, we will refer to this

SVC configuration as *baseline configuration* (BC) and to the rate-controlled SVC (RC-SVC) as SB-BC.

For the proposed IL-MB VBR controller, the following three-dependency layer H.264/SVC encoder configuration was used:

a) Number of pictures: 900.

b) GoP size/Intra period: 8/32 pictures.

c) GoP structure: hierarchical B pictures.

d) Search range for motion estimation: 16×16 pixels.

e) Number of dependency layers: $D = 3$.

   i) $d = 0$: QCIF, $f_{out}^{(0,2)} = 12.5$ Hz $\left(T^{(0)} = 3\right)$.

   ii) $d = 1$: CIF, $f_{out}^{(1,2)} = 12.5$ Hz $\left(T^{(1)} = 3\right)$.

   iii) $d = 2$: CIF, $f_{out}^{(2,3)} = 25$ Hz $\left(T^{(2)} = 4\right)$.

f) Symbol mode: CAVLC.

We will refer to this SVC encoder configuration as *compact configuration* (CC) since it consists of only three layers in comparison with the BC, which is made of five layers. The RC parameters took the following values: $nTF = 50\%$ and $BD = 3$ s., the same as for SB-BC, and $t_{min}^{(0)} = 1$, $t_{min}^{(1)} = 2$, and $t_{min}^{(2)} = 2$. As can be observed, $t_{min}^{(0)}$ and $t_{min}^{(2)}$ were set such that HRD-compliant sub-streams for QCIF@6.25 Hz ($d = 0$) and high-quality (HQ) CIF@12.5 Hz ($d = 2$) were available, as for SB-BC. Henceforth, this RC-SVC encoder will be referred to as MB-CC.

Furthermore, in order to analyze the behavior of the proposed VBR controller if only one buffer per dependency layer was controlled (that corresponding to the highest temporal resolution), an additional H.264/SVC encoder and RC configuration with $t_{min}^{(d)} = t_{max}^{(d)}$ for every dependency layer was also studied. We will refer to it as SB-CC.

The set of video sequences used in these experiments was the same as that used to evaluate the performance of the baseline VBR controller in the mobile live streaming scenario, that is: *Bus*, *Football*, *Foreman*, *Mobile*, *Soccer-Mobile-Foreman*, *Spiderman*, and *The Lord of the Rings*.

All the video sequences were encoded using the set of constant QP values that best approached some pre-established target bit rates. We will refer to this RC-SVC encoder as CQP-CC. For the first group of sequences, the target bit rates for the highest temporal resolution of each layer d, i.e., QCIF@12.5 Hz $(0, 2)$, low-quality (LQ) CIF@12.5 Hz $(1, 2)$ and HQ CIF@25 Hz $(2, 3)$ were those suggested in [Wien and Schwarz, 2005] for the spatial/CGS testing scenario. For the second group, the following medium-quality target bit rates associated with the highest temporal resolution of each layer d were selected: 96 $(0, 2)$, 192 $(1, 2)$, and 512 kbps $(2, 3)$. The output bit rates $R_{out}^{(d,t)}$ generated by CQP-CC encoding for the five target spatio-temporal resolutions were used as target bit rates $R^{(d,t)}$ for the three assessed RC-SVC encoders, i.e.: SB-BC, SB-CC, and MB-CC. The same target bit rates were assigned to each involved spatio-temporal layer for all the RC-SVC encoders so that all the compared encoders operated under the same target bit rate constraints. The actual $R^{(d,t)}$ values are listed in Table 5.1. It should be noticed that the spatio-temporal resolutions QCIF@6.25 Hz and HQ CIF@12.5 Hz are not rate-controlled in SB-CC encoding.

5.3.2 Experimental Results and Discussion

The average results in terms of average PSNR μ_{PSNR} over all the test video sequences are summarized in Table 5.2. Specifically, the PSNR increments $\Delta\mu_{PSNR}$ with respect to CQP-CC encoding are given. Three rows per spatio-temporal layer are shown, one for each assessed RC-SVC encoder. As can be observed, the average PSNR achieved by SB-CC and MB-CC at every spatio-temporal layer were similar to that of CQP-CC and higher than that of SB-BC, which, for the same target bit rate $R^{(d,t)}$, is encoding more layers.

Layer (d,t)	**RC-SVC Encoder**	**Resolution**	**Assigned $\mathbf{R^{(d,t)}}$ from CQP-CC**
(0,1)	SB-BC	QCIF@6.25 Hz	$R_{out}^{(0,1)}$
-	-		
(0,1)	MB-CC		
(1,2)	SB-BC	QCIF@12.5 Hz	$R_{out}^{(0,2)}$
(0,2)	SB-CC		
(0,2)	MB-CC		
(2,2)	SB-BC	LQ CIF@12.5 Hz	$R_{out}^{(1,2)}$
(1,2)	SB-CC		
(1,2)	MB-CC		
(3,2)	SB-BC	HQ CIF@12.5 Hz	$R_{out}^{(2,2)}$
-	-		
(2,2)	MB-CC		
(4,3)	SB-BC	HQ CIF@25 Hz	$R_{out}^{(2,3)}$
(2,3)	SB-CC		
(2,3)	MB-CC		

Table 5.1: Target bit rates assigned to each spatio-temporal layer of the compared RC-SVC encoders.

A detailed comparison of the algorithms is shown in Tables 5.3 and 5.4. Table 5.3 shows the results achieved for *Bus*, a representative example of video sequence with stationary complexity, and Table 5.4 shows the results for *The Lord of the Rings*, a representative example of video sequence with scene changes. The results in terms of average PSNR indicate that, for non-stationary complexity sequences, the performance of either SB-CC or the proposed MB-CC improved that of the nearly constant quality system at most spatio-temporal layers. However, for stationary complexity sequences, the performance achieved by the three VBR controllers were

Layer (d,t)	RC-SVC encoder.	$\Delta\mu_{PSNR}$ (dB)	$\Delta\overline{\sigma}_{PSNR,j}$ (dB)	Bit Rate Error (%)	#O/#U	μ_V (%)
(0,1)	SB-BC	-0.12	0.09	1.00	0/0	52.34
(0,1)	SB-CC	0.05	0.22	2.68	5/0	59.90
(0,1)	MB-CC	0.05	0.19	1.48	0/0	55.72
(1,2)	SB-BC	-0.25	0.15	1.86	1/0	64.77
(0,2)	SB-CC	0.10	0.19	0.94	0/0	55.09
(0,2)	MB-CC	0.08	0.16	0.84	0/0	54.66
(2,2)	SB-BC	-0.16	0.11	0.94	0/0	52.98
(1,2)	SB-CC	0.00	0.09	0.93	0/0	55.26
(1,2)	MB-CC	0.00	0.09	1.02	0/0	55.19
(3,2)	SB-BC	-0.10	0.07	0.59	0/0	52.42
(2,2)	SB-CC	0.05	0.12	1.45	0/0	56.42
(2,2)	MB-CC	0.06	0.11	0.79	0/0	54.17
(4,3)	SB-BC	-0.21	0.06	1.57	0/0	64.82
(2,3)	SB-CC	0.09	0.11	0.48	0/0	54.02
(2,3)	MB-CC	0.08	0.10	0.51	0/0	53.43

Table 5.2: Average results achieved by the SB-BC, the SB-CC, and the proposed MB-CC VBR controllers. Incremental results are given with respect to CQP-CC encoding.

very close to that of the nearly constant quality system.

Representative behaviors of the encoder buffer occupancy, PSNR and QP time evolutions corresponding to the two lower spatio-temporal resolutions, QCIF@6.25 Hz and QCIF@12.5 Hz, are depicted in Figs. 5.4(a) and 5.4(b) for *Bus*, and Figs. 5.5(a) and 5.5(b) for *The Lord of the Rings*, where the QCP-CC plots have been removed for clarity reasons. High quality plots including those of CQP-CC encoding can be found in [Sanz-Rodríguez, 2011] for every spatio-temporal resolution. As can

Layer (d,t)	$\mathbf{R^{(d,t)}}$ **(kbps)**	**RC-SVC Scheme**	$\mu_{\mathbf{PSNR}}$ **(dB)**	$\overline{\sigma}_{\mathbf{PSNR,j}}$ **(dB)**	**Bit Rate Error (%)**	**#O/#U**	$\mu_{\mathbf{V}}$ **(%)**
(0,1)	73.89	CQP-CC	31.24	0.31	-	0/0	51.97
(0,1)		SB-BC	31.24	0.31	-0.03	0/0	51.89
(0,1)		SB-CC	31.23	0.39	0.95	0/0	56.07
(0,1)		MB-CC	31.22	0.40	0.96	0/0	55.53
(0,2)	101.61	CQP-CC	31.11	0.27	-	0/0	52.70
(1,2)		SB-BC	31.00	0.32	1.27	0/0	63.15
(0,2)		SB-CC	31.10	0.35	0.52	0/0	53.94
(0,2)		MB-CC	31.10	0.36	0.50	0/0	52.88
(1,2)	202.67	CQP-CC	26.94	0.16	-	0/0	51.91
(2,2)		SB-BC	26.86	0.19	-0.09	0/0	51.03
(1,2)		SB-CC	26.92	0.23	0.26	0/0	53.44
(1,2)		MB-CC	26.92	0.23	0.44	0/0	53.15
(2,2)	404.97	CQP-CC	30.01	0.19	-	0/0	52.01
(3,2)		SB-BC	29.99	0.19	-0.04	0/0	52.15
(2,2)		SB-CC	30.01	0.25	0.40	0/0	54.76
(2,2)		MB-CC	30.02	0.24	0.53	0/0	53.66
(2,3)	517.67	CQP-CC	30.05	0.17	-	0/0	52.20
(4,3)		SB-BC	29.91	0.18	1.5	0/0	65.57
(2,3)		SB-CC	30.05	0.22	0.31	0/0	54.43
(2,3)		MB-CC	30.06	0.21	0.46	0/0	53.41

Table 5.3: Performance comparison among the SB-BC, the SB-CC, and the proposed MB-CC VBR controllers, for a specific stationary complexity video sequence, *Bus*. The results achieved by CQP-CC encoding have also been included for reference.

be shown, in the stationary scenario the three assessed VBR controllers were able to keep the QP fluctuation low most of the time, thus providing a nearly constant PSNR time evolution. However, some high buffer levels and QP fluctuations were observed at certain time instants for SB-BC (see Figure 5.4(b)) because more layers were encoded for a given target bit rate. In the non-stationary scenario the three assessed algorithms made, with some exceptions that will be discussed, a proper use of the buffer fullness to provide PSNR and QP evolutions closer to those of the nearly constant quality system, as expected for VBR control algorithms, given that larger amount of bits were assigned to more complex scenes. The undesirable buffer levels observed in the SB-CC VBR controller at the layer $(0, 1)$ (see Figure 5.5(a)) were due to the fact that only the highest temporal resolution buffer associated with the layer $(0, 2)$ was considered for QP estimation. Furthermore, as in the stationary scenario, some undesirable buffer levels and QP fluctuations also happened at the highest temporal resolution sub-stream for SB-BC (see Figure 5.5(b)), again due to the fact that it is coding more layers.

From the quality consistency point of view, the performance of the VBR controllers was also assessed by means of a time-local version of the PSNR standard deviation $\overline{\sigma}_{PSNR,j}$, which, as already defined in Subsection 4.4.2, attempts to measure the quality consistency within a scene by reducing the impact of the scene changes on the PSNR standard deviation. The average results over all the test video sequences in terms of $\overline{\sigma}_{PSNR,j}$ increment with respect to CQP-CC encoding, $\Delta\overline{\sigma}_{PSNR,j}$, are provided in Table 5.2. As can be observed, the three VBR controllers achieved a quality consistency close to that of CQP-CC encoding. Furthermore, the $\overline{\sigma}_{PSNR,j}$ differences among them were not significant either in particular stationary (see Table 5.3) or non-stationary scenarios (see Table 5.4), as expected, since the VBR controllers were specially designed to provide consistent-quality scalable sub-streams.

The VBR controllers were also quantitatively compared in terms of target bit rate adjustment and buffer level behavior. To this end, the following metrics were employed: output bit rate error with respect to that of CQP-CC encoding, number

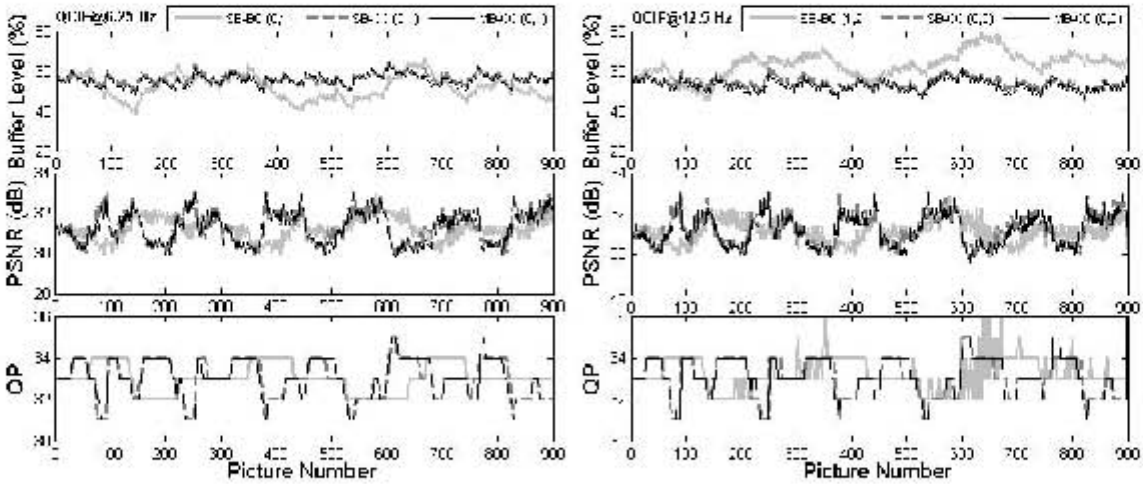
Layer (d,t)	$\mathbf{R^{(d,t)}}$ **(kbps)**	**RC-SVC Scheme**	$\mu_{\mathbf{PSNR}}$ **(dB)**	$\overline{\sigma}_{\mathbf{PSNR,j}}$ **(dB)**	**Bit Rate Error (%)**	**#O/#U**	$\mu_{\mathbf{V}}$ **(%)**
(0,1)	66.50	CQP-CC	34.45	0.66	-	42/48	49.70
(0,1)		SB-BC	34.40	0.91	2.31	0/0	53.89
(0,1)		SB-CC	34.75	0.96	5.53	36/0	70.17
(0,1)		MB-CC	34.77	0.94	1.70	0/0	62.76
(0,2)	93.99	CQP-CC	34.36	0.66	-	104/113	46.71
(1,2)		SB-BC	34.23	0.99	2.59	0/0	60.90
(0,2)		SB-CC	34.80	0.97	1.05	0/0	54.38
(0,2)		MB-CC	34.76	0.94	0.20	0/0	50.47
(1,2)	186.51	CQP-CC	32.87	0.90	-	98/113	47.26
(2,2)		SB-BC	32.77	1.09	1.90	0/0	52.02
(1,2)		SB-CC	33.15	1.08	1.00	0/0	55.21
(1,2)		MB-CC	33.12	1.07	1.18	0/0	55.88
(2,2)	385.34	CQP-CC	35.25	0.83	-	93/114	45.16
(3,2)		SB-BC	35.31	0.94	1.73	0/0	51.98
(2,2)		SB-CC	35.54	1.00	3.24	0/0	71.54
(2,2)		MB-CC	35.58	0.94	0.92	0/0	58.58
(2,3)	507.26	CQP-CC	35.29	0.81	-	217/241	45.11
(4,3)		SB-BC	35.27	0.95	2.26	0/0	58.81
(2,3)		SB-CC	35.69	0.99	0.42	0/0	53.58
(2,3)		MB-CC	35.67	0.94	0.04	0/0	49.95

Table 5.4: Performance comparison among the SB-BC, the SB-CC, and the proposed MB-CC VBR controllers, for a specific non-stationary complexity video sequence, *The Lord of the Rings*. The results achieved by CQP-CC encoding have also been included for reference.

#O of overflowed pictures, number #U of underflowed pictures, and mean buffer level μ_V. As can be seen in Table 5.2, the average output bit rate errors achieved by the three VBR controllers at every spatio-temporal layer were generally below 2%, that is the maximum bit rate error recommended in [Wien and Schwarz, 2005] for the spatial/CGS testing scenario. Nevertheless, in some sequences with time-varying complexity, such as *The Lord of the Rings*, higher bit rate errors occurred in some spatio-temporal layers for the SB-BC and SB-CC VBR controllers (see Table 5.4). Specifically, for the SB-BC VBR controller, such bit rate mismatches together with the large μ_V values observed in layers $(1, 2)$ and $(4, 3)$ indicate that the corresponding target bit rates were not high enough to encode all the spatio-temporal layers. For the SB-CC VBR controller, the results in terms of bit rate error, mean buffer level, and number of overflows shown in Table 5.4 for layers $(0, 1)$ (see also Figure 5.5(a)) and $(2, 2)$, proved the need of simultaneously controlling all the involved buffers within a dependency layer, as in the MB-CC VBR controller, which was able to prevent overflow and underflow in all the encodings (see Tables 5.2–5.4).

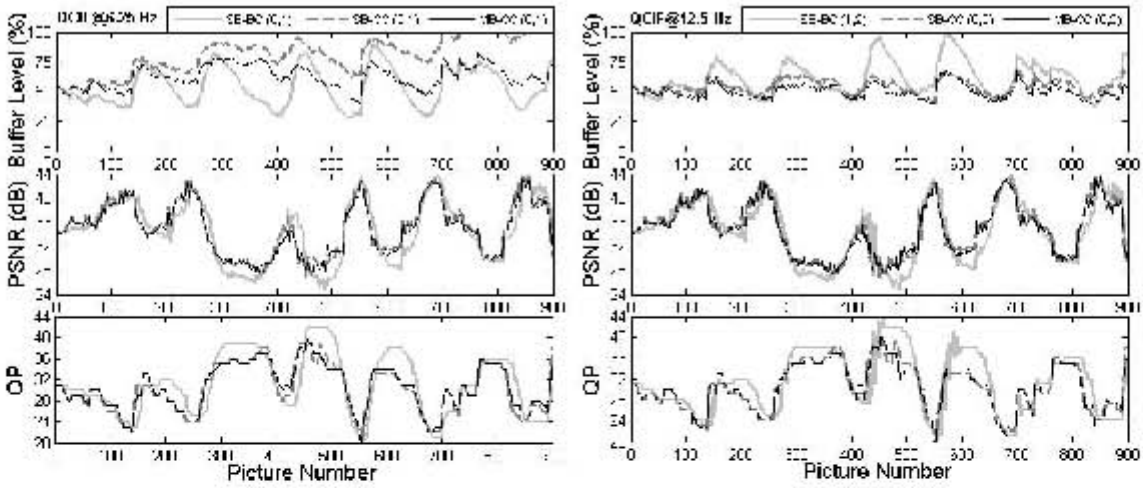
It should be noticed that the good performance achieved by the proposed MB-CC VBR controller, specifically at the lowest and the highest dependency layers, could be partly due to the fact that the total bit rate per dependency layer was optimally distributed among temporal layers since the corresponding $R^{(d,t)}$ values were previously obtained using CQP-CC encoding. In real-time video coding applications, the optimal distribution of the target bit rate among temporal layers is not known in advance because it depends on the video content. For instance, the target bit rate for a sequence with high spatial detail but low motion content should be shared out among temporal layers such that the bit resources are mainly allocated to K pictures. However, for a sequence with medium-low spatial detail but high motion content, a more balanced target bit rate distribution between K and NK pictures is desirable to encode the motion information better.

In order to explore the sensitivity of the proposed MB-CC VBR controller to target bit rate deviations with respect to those obtained by CQP-CC encoding, we



(a) (b)

Figure 5.4: Encoder buffer level, PSNR and QP time evolutions corresponding to the spatio-temporal resolutions (a) QCIF@6.25 Hz and (b) QCIF@12.5 Hz for *Bus*. High-quality plots are available on-line in [Sanz-Rodríguez, 2011].



(a) (b)

Figure 5.5: Encoder buffer level, PSNR and QP time evolutions corresponding to the spatio-temporal resolutions (a) QCIF@6.25 Hz and (b) QCIF@12.5 Hz for *The Lord of the Rings*. High-quality plots are available on-line in [Sanz-Rodríguez, 2011].

performed an *ad hoc* experiment. This experiment involved modifying the target bit rates of the low temporal resolutions of those layers encoded using the proposed IL-MB approach. In particular, assuming that the target bit rates for the highest temporal resolutions can be set in advance following, for instance, the recommendation in [Wien and Schwarz, 2005], the target bit rates for QCIF@6.25 Hz $(0, 1)$ and HQ CIF@12.5 Hz $(2, 2)$ were deviated $\pm 2\%$, $\pm 5\%$, and $\pm 10\%$ from their corresponding reference target bit rates. The average results over all the test video sequences in terms of $\Delta\mu_{PSNR}$, $\Delta\overline{\sigma}_{PSNR,j}$, and bit rate error with respect to the those achieved without $R^{(d,t)}$ deviations, as well as the number of overflows and underflows and mean buffer level, are summarized in Table 5.5. As can be observed, target bit rate deviations of 10% led to noticeable loss of qualty consistency (due to the increase of QP fluctuations caused by the sub-optimal target bit rates), bit rate errors above 2%, and mean buffer levels close to either overflow or underflow.

It is also interesting to notice how the sub-optimal distribution of the target bit rate affects to the buffer levels of the involved temporal layers. To this end, let us focus on the results from layers $(0, \{1, 2\})$ for a target bit rate deviation of $+10\%$. As can be observed, the corresponding μ_V took opposite values: the low temporal resolution buffer was close to underflow, while the high temporal resolution buffer was close to overflow. This mirror-like behavior of the buffers is due to the fact that the buffer modeling stage averages the current encoding states of the involved temporal resolutions at many time instants for $nV^{(0)}$, $nAU^{(0)}$, and $QP^{(0)}_{REF}$ computation. Although optimum adjustment to $R^{(0,\{1,2\})}$ or nTF were not achieved, neither overflows nor underflows ocurred in most of the assessed video sequences. However, if the highest temporal resolution buffer was only considered for QP estimation (as in SB-CC), a suitable adaptation to both $R^{(0,2)}$ and nTF would be achieved at the expense of a higher underflow risk at the lowest temporal resolution buffer. In short, when the target bit rate distributions among temporal resolutions are not optimally distributed, the proposed method for IL-MB control makes its best to provide a good trade-off between quality consistency and buffer control in all the involved buffers.

Layer (d,t)	$\mathbf{R^{(d,t)}}$ **Dev. (%)**	$\mathbf{\Delta\mu_{PSNR}}$ **(dB)**	$\mathbf{\Delta\overline{\sigma}_{PSNR,j}}$ **(dB)**	**Bit Rate Error (%)**	**#O/#U**	$\mu_{\mathbf{V}}$ **(%)**
(0,1)	+10	0.80	0.18	1.60	0/0	31.90
	+5	0.44	0.02	1.05	0/0	41.56
	+2	0.14	0.00	0.89	0/0	49.93
	-2	-0.28	-0.01	1.75	0/0	59.36
	-5	-0.62	0.03	2.17	0/0	64.51
	-10	-1.20	0.14	2.69	1/0	71.01
(0,2)	0	-0.04	0.22	2.15	1/0	67.42
		0.03	0.04	1.52	0/0	61.16
		0.00	0.01	1.06	0/0	57.29
		-0.08	0.00	0.66	0/0	51.78
		-0.17	0.03	0.72	0/0	48.84
		-0.34	0.17	0.74	0/0	45.09
(1,2)	0	-0.07	0.02	0.75	0/0	55.30
		-0.04	0.01	0.97	0/0	55.23
		-0.04	0.01	0.84	0/0	55.18
		-0.06	0.00	0.91	0/0	54.98
		-0.09	-0.01	0.77	0/0	54.59
		-0.16	-0.02	0.70	0/0	54.88
(2,2)	+10	0.81	0.24	2.29	0/0	29.98
	+5	0.47	0.04	1.29	0/0	39.88
	+2	0.17	0.01	0.42	0/0	48.80
	-2	-0.27	-0.01	1.34	0/0	58.38
	-5	-0.61	0.02	1.88	0/0	64.87
	-10	-1.18	0.17	2.47	0/0	72.20
(2,3)	0	-0.01	0.21	1.87	0/0	67.68
		0.05	0.03	1.28	0/0	61.34
		0.01	0.01	0.78	0/0	57.04
		-0.08	0.00	0.51	0/0	49.75
		-0.17	0.03	0.60	0/0	47.06
		-0.33	0.21	0.77	0/0	43.25

Table 5.5: Average results achieved by the proposed MB-CC VBR controller for different target bit rate deviations at layers $(0, 1)$ and $(2, 2)$. Incremental results are given with respect to those achieved by CQP-CC encoding.

5.4 Summary and Conclusions

In this chapter an IL-MB approach built on top of the baseline VBR controller described in the previous chapter has been proposed. Given a dependency layer, the proposed method aims to deliver HRD-compliant sub-streams with different temporal resolutions. In doing so, temporal scalability is fully exploited by reducing the number of dependency layers required to provide the same spatial or quality level for decoding terminals requiring different frame rates. For this purpose, the proposed IL-MB VBR controller estimates, on a frame basis, the most appropriate QP value such that a set of virtual buffers, each one associated with a temporal resolution of the same dependency layer, is maintained at secure levels, while minimizing the distortion of the corresponding sub-streams. Furthermore, the decision rules suggested for simultaneously controlling the set of virtual buffers might be used in any other RC algorithm for SVC.

In order to assess the performance of the proposed IL-MB VBR controller, three RC-SVC encoders were compared on a simulated mobile live streaming scenario: SB-BC, SB-CC, and the proposed MB-CC. The output bit rates generated by CQP-CC encoding for the target spatio-temporal resolutions were used as target bit rates for these RC-SVC encoders.

In terms of average PSNR, the performance achieved by SB-CC and MB-CC was similar to that of CQP-CC encoding and higher than that of SB-BC, which is encoding more layers for the same target bit rate. In terms of quality consistency, all the RC-SVC encoders achieved results similar to CQP-CC encoding.

From the target bit rate adjustment and buffer fullness behavior points of view, the average output bit rate errors achieved by the three VBR controllers at every spatio-temporal resolution were generally bellow 2%, which is an acceptable error in practical SVC applications. However, in some situations undesirable buffer levels (and consequently QP fluctuations) and high bit rate errors happened in particular spatio-temporal layers for the SB-BC and SB-CC cases. Specifically, for the SB-BC

VBR controller, the results indicated that the target bit rates were not high enough to encode all the layers. For the SB-CC VBR controller, the results demonstrated the need of simultaneously controlling all the involved buffers within a dependency layer.

Finally, the proposed IL-MB VBR controller was also assessed in terms of sensitivity to the target bit rate deviations with respect to those obtained by CQP-CC encoding. The results allowed us to conclude that target bit rate deviations above 5% might produce noticeable loss of quality consistency, high bit rate errors, and buffer levels close to either overflow or underflow.

Chapter 6

GP-Based QP Increment Estimation Design

As briefly described in Chapters 4 and 5, we have used GP regression for the estimation of $\Delta QP^{(d)}$. In particular, we have designed four specific GPs in this thesis: two (K-picture and NK-picture) GPs for IL-SB control and other two for IL-MB control. In Chapter 4 we presented the first two models, while in Chapter 5 we focused on the last two.

The aim of this chapter is to describe the general methodology conducted to properly use GPs for our specific regression problem consisting of making predictions of $\Delta QP^{(d)}$ subject to some practical constraints. Specifically, as in many machine learning tasks, the proposed methodology for GP parameter selection relies on three well-differenced stages: generation of the training data set, training, and validation.

This chapter is organized as follows. In Section 6.1 we discuss the motivation behind the features selected as components of the input vector to the GPs. In Section 6.2 we explain the reasons that motivated the use of GPs for our regression problem. The procedure followed for the generation of the training data set is described in Section 6.3. The training and validation processes are explained in Sections 6.4 and 6.5, respectively. Additionally, the methodology followed to design the post-

processing stage of the output of the GPs for NK pictures is also described in this last section. Finally, in Section 6.6 some conclusions are drawn.

6.1 Input Vector Selection

There are many parameters that can potentially influence the selection of a suitable $\Delta QP^{(d)}$ value at a specific dependency layer d. They include, for example, measures of actual buffer fullness and AU output bits, target buffer fullness, buffer size, reference QP value, video content properties, GoP size, type of buffer control, dependency and temporal layer identifiers, and others. In order to reach a good compromise between performance and computational cost, we have selected four parameters as components of the input vector to the GPs: $nV^{(d)}$, $nAU^{(d)}$, nTF, and BD. The reasons for selecting these ones and rejecting others are given next.

The normalized versions of both buffer fullness $nV^{(d)}$ and AU output bits $nAU^{(d)}$ have to be considered in order to guarantee long-term average bit rate adaptation while maintaining the buffer occupancy at secure levels. In fact, similar parameters to these ones have already been used successfully in previous works on the same subject, as those described in [Rezaei et al., 2008].

The normalized target buffer fullness nTF is used by the rate controller to push the buffer occupancy toward that reference point. Although in VBR scenarios it is common to operate with target buffer fullness values between 40% and 60% of the buffer size, we decided to consider this parameter because its influence on the selection of $\Delta QP^{(d)}$ becomes crucial when it takes either lower or higher values, since the risk of underflow or overflow, respectively, increases dramatically and must be controlled.

The buffer size BD is related to the region of the R-D space where the rate controller can operate; in other words, it determines the operating point between the constant-rate region (small buffer size) and the constant-quality region (large buffer size). Thus, the larger the buffer size, the smoother the QP variation should be, so

that the visual quality consistency is high.

Although not as components of the input vector, the type of buffer control and the temporal layer identifier have been taken into account by considering four specific GP designs. In particular, for each type of IL buffer control (SB and MB), two different GP-based $\Delta QP^{(d)}$ estimation models were designed, one for the temporal base layer, and the other for the temporal enhancement layers.

Other parameters were considered and discarded for the sake of the *performance-complexity* trade-off, in particular: the reference QP value, some video complexity measures, the GoP size, and the dependency layer identifier. Although all of these parameters have an undeniable influence on the proper selection of $\Delta QP^{(d)}$, their contribution does not turn out to be essential in a VBR scenario where a long-term average bit rate adaptation is sufficient. On the other hand, if they were considered, both the complexity of the GP training process and the operation complexity would considerably increase due to the increment of the input vector dimension.

6.2 Why GPs for Regression?

Supervised learning is a category of machine learning that consists of inferring a mapping function from a supervised (previously labeled) training data. This function aims to generate the desired outputs from so-called test inputs. Supervised learning can be divided into two algorithm types: regression (for continuous outputs), and classification (for discrete outputs).

Parametric models learn through a supervised learning process by adapting their parameters. These models have been typically used for regression and classification. Although they provide some interesting advantages such as easy interpretability, simple parametric models become inefficient for a large amount of training data, and other more complex methods such as feed-forward Neural Networks may not be easy to use (they need to resort to some overfitting control methods and a rigorous cross-validation process to choose the number of neurons). On the other hand, kernel

methods, such as Support Vector Machines and GPs are able to provide flexible solutions that are more suitable in practice [Rasmussen, 2003].

In particular, GPs are non-parametric models that actually provide state-of-the-art performance regression [Rasmussen, 1996] and classification [Naish-Guzman and Holden, 2008] problems. GPs have interesting advantages that make them suitable for the purposes of the thesis. Some of these advantages include: high accuracy, no overfitting, and simple model selection scheme. However, large training data sets make full GPs unfeasible due to the associated computational complexity. For n input data samples, training a full GP requires $\mathcal{O}(n^3)$ computation time, which may be significantly high when using databases with more than a few thousand samples (see [Rasmussen and Williams, 2006]). Because of the large amount of samples used for the training our $\Delta QP^{(d)}$ estimation model (about one million samples), we had to resort to some sparse approximation. Among the approximate GPs proposed in the literature, the so-called sparse pseudo-inputs GP (SPGP) due to Snelson and Gharahmani [Snelson and Ghahramani, 2006] is the regression method we applied in this thesis. SPGP represents the state of the art, and is used as a benchmark by other approximations [Lázaro-Gredilla, 2010]. The mean predictions of SPGP can also be expressed in terms of Equations (4.16) and (4.17). However, unlike full GPs that use all training samples as centers, SPGP selects a smaller set of m adequate locations from the input space representing the real data well. Thus, SPGP is able to reduce the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(m^2 n)$.

6.3 Generation of the Training Data Set

A training data set consisting of pairs

$$\left\{\mathbf{X}^{(d)}, \Delta QP^{*(d)}\right\},$$

where $\mathbf{X}^{(d)}$ is the input feature vector previously defined as

$$\mathbf{X}^{(d)} = \left(nV^{(d)}, nAU^{(d)}, nTF, BD\right)^T,$$

and $\Delta QP^{*(d)}$ is the desired output QP increment, should be generated in order to properly train GPs for our purposes. The generation of these training pairs is actually a key step in the success of the proposed approach. This section is devoted to describe this process.

Training samples were extracted from a representative set of video sequences exhibiting a large variety of spatio-temporal contents, so that the trained GPs work properly for any type of input sequence. This set of video sequences used for training consisted of two parts.

1. Some of the well-known sequences commonly used in the field; specifically [Xiph.org, 2011]: *Akiyo*, *City*, *Container*, *Crew*, *Hall*, *Highway*, *Ice*, *News*, *Paris*, *Silent*, *Soccer*, and *Tempete*. We used 300 pictures per sequence and some of them were upsampled and/or downsampled in order to get QCIF, CIF and 4×CIF (4CIF) resolutions.

2. Some sequences extracted from high-quality DVDs; specifically: *Airshow* (documentary), *Cities* (documentary), and *Ice Age* (cartoon). In this case, we used 900 pictures per sequence that were downsampled to get QCIF and CIF resolutions from the original SDTV format. These sequences are available on-line in [Sanz-Rodríguez, 2011].

Notice that none of these training sequences was used in the performance assessment of the VBR controller described in previous chapters.

For each training sequence, a reduced number of consecutive GoP pairs were selected along the video sequence. The first GoP of each pair was used to initialize the average texture and motion complexities (a complete GoP is needed because initial average texture and motion complexities are required for each spatio-temporal layer). The second GoP was used to actually extract training data pairs $\left\{\mathbf{X}^{(d)}, \Delta QP^{*(d)}\right\}$. In order to obtain training samples for a variety of scenarios, each GoP pair was encoded using Φ different configurations. These Φ different configurations involved several encoder- and RC-related parameters: number of dependency layers, spatial

resolutions, GoP size, target bit rate, target buffer level, and buffer size.

6.3.1 Getting Initial Average Complexities

Given an encoding configuration ϕ, a baseline QP, denoted as $QP_{R_\phi}^{(d)}$, was chosen for each dependency layer d so that the corresponding target bit rate for the whole sequence $R_\phi^{(d)}$ would be generated. Then, the first GoP of each GoP pair was encoded $\mathcal{A}$ times, each one using a different QP increment with respect to $QP_{R_\phi}^{(d)}$, i.e., $\left\{QP_{R_\phi}^{(d)} + \Delta QP_\alpha^{(d)}\right\}_{\alpha=1\ldots\mathcal{A}}$, and the computed average texture and motion complexities for each QP increment were stored as initial complexities for the subsequent process. Specifically, in our experiments the number of encodings for a given baseline QP was $\mathcal{A} = 11$, using QP increments from -5 to 5.

6.3.2 Generating Training Pairs

As previously mentioned, once the initial average texture and motion complexities had been obtained for every layer, the second GoP was used to extract the training pairs. For each picture j of the second GoP, the aim was to determine the optimum QP increment for a wide range of potential conditions concerning the buffer occupancy and the adjustment to the AU target bits. In order to achieve this variety of encoding conditions, the multiple encoding process initiated for the first GoP continued along the second GoP for the same set of $\mathcal{A}$ quantization values. As a result, before encoding the jth picture, all the previous pictures had been encoded $\mathcal{A}$ times, so that a set of $\mathcal{A}$ input vectors would be available:

$$\mathbf{X}_{j,\phi,\alpha}^{(d)} = \left(nV_{j,\phi,\alpha}^{(d)}, nAU_{j,\phi,\alpha}^{(d)}, nTF_\phi, BD_\phi\right)^T,$$

where variables $nV_{j,\phi,\alpha}^{(d)}$ and $nAU_{j,\phi,\alpha}^{(d)}$ summarize the encoding state after the $(j-1)$th picture. Then, for each one of the $\mathcal{A}$ possible input vectors, the challenge was to find the optimum $\Delta QP^{*(d)}$ so that each involved buffer n, with $n = \max\left[t_{min}^{(d)}, t\right] \ldots t_{max}^{(d)}$, was maintained at secure levels, while minimizing the distortion of the corresponding sub-streams. To this end, a second set of $\mathcal{B}$ quantization increments

$\left\{\Delta QP_{\beta}^{(d)}\right\}_{\beta=1,\ldots,\mathcal{B}}$ with respect to $\left\{QP_{R_\phi}^{(d)} + \Delta QP_{\alpha}^{(d)}\right\}$ was used to encode the jth picture. Particularly, in our experiments a total of $\mathcal{B} = 23$ quantization increments from -11 to 11 were used to find the optimum $\Delta QP^{*(d)}$.

Finally, for each input vector $\mathbf{X}_{j,\phi,\alpha}^{(d)}$, the QP increment $\Delta QP_{\beta}^{(d)}$ that minimized certain cost function Γ was chosen as the optimum one:

$$\Delta QP^{*(d)} = \underset{\Delta QP_{\beta}^{(d)}}{\operatorname{argmin}} \Gamma\left(\Delta QP_{\beta}^{(d)}\right). \tag{6.1}$$

The cost function has been designed *ad hoc* for this problem aiming at properly balance several conflicting factors: quality consistency, buffer control, and QP consistency. Specifically Γ adopts the form that follows:

$$\begin{aligned}\Gamma\left(\Delta QP_{\beta}^{(d)}\right) &= \lambda_1\theta\left(\frac{\sum_n \frac{D_j^{(d)} - \overline{D}^{(d,n)}}{255}}{t_{max}^{(d)} - \max\left[t_{min}^{(d)}, t\right] + 1}\right)^2 + \\ &\quad \lambda_2\left(\frac{\sum_n\left(\frac{V_{j+1}^{(d,n)}}{BD_\phi \times R_\phi^{(d,n)}} - nTF_\phi\right)}{t_{max}^{(d)} - \max\left[t_{min}^{(d)}, t\right] + 1}\right)^2 + \\ &\quad \lambda_3\left(\frac{\Delta QP_q^{(d)}}{\Delta QP_{MAX}^{(d)}}\right)^2.\end{aligned} \tag{6.2}$$

The first term monitors the quality consistency by means of the squared mean of the differences between the distortion $D_j^{(d)}$ of the current picture and the average distortion $\overline{D}^{(d,n)}$ of each sub-stream (d, n). The distortion metric used was the *mean of the absolute error* between the original and reconstructed luminance pictures.

The second term considers the buffer control through the squared mean of the differences between the normalized current buffer level $V_{j+1}^{(d,n)}/BD_\phi \times R_\phi^{(d,n)}$ corresponding to each sub-stream (d, n) and the normalized target buffer fullness nTF_ϕ.

The third term watches over the QP consistency by means of the squared ratio of the considered ΔQP and the maximum allowed QP deviation $\Delta QP_{MAX}^{(d)}$, which

was set to 11 QP units in our experiments. The motivation for this third term comes from the fact that, in same cases, due to the high coding efficiency of H.264/SVC at high spatio-temporal layers, several QP increments yield quite similar distortion and number of output bits because of the low energy of the AC transformed coefficients.

The weight vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)^T$ was selected by means of a validation process (described in Section 6.5) to achieve the best trade-off among the three terms of the cost function. In order to obtain more meaningful values for the weights, the first term of the cost function was scaled by introducing an additional factor θ such that its dynamic range would be similar to those of the second and third terms. In particular, θ was set to 100 in our experiments. Finally, as we are only interested in the relative weights, the three weights were made to sum up to one.

Before starting out the network training, a set of possible weight vectors for the cost function was pre-established by considering different trade-offs among quality consistency, buffer control, and QP consistency. Specifically, those weight vectors previously selected are given in Table 6.1. By using Equations (6.1) and (6.2) for data labeling, several sets of training data, one per weight vector, were generated for SB control by setting $t_{min}^{(d)} = t_{max}^{(d)}$, as well as for MB control by setting $t_{min}^{(d)} = t_{max}^{(d)} - 2$ so that three buffers at the most could be simultaneously controlled[1]. Additionally, for each training data set, a reduced amount of values for both the normalized target buffer fullness nTF and buffer size BD were selected so that a wide range of VBR applications would be covered; specifically, nTF and BD were sampled in the following ranges: $0.1 \leq nTF \leq 0.9$ and $1 \leq BD \leq 3$.

For any of the pre-established cost function weight vectors used for either SB or MB labeling, the following conclusions were drawn from the training data distributions:

1) Figures 6.1(a) and 6.1(b) show superimposed training data distributions for

[1]For a frame rate of 25 Hz, $t_{min}^{(d)} = t_{max}^{(d)} - 2$ means that the minimum output frame rate of encoded video meeting the HRD constraints is 6.25 Hz, which is sufficient for practical applications [Wien and Schwarz, 2005].

	$\boldsymbol{\lambda}_a$	$\boldsymbol{\lambda}_b$	$\boldsymbol{\lambda}_c$
λ_1	0.90	0.75	0.50
λ_2	0.09	0.24	0.49
λ_3	0.01	0.01	0.01

Table 6.1: Weight vectors, denoted as $\boldsymbol{\lambda}_a$, $\boldsymbol{\lambda}_b$, $\boldsymbol{\lambda}_c$, for the cost function in Equation (6.2) used for training data labeling.

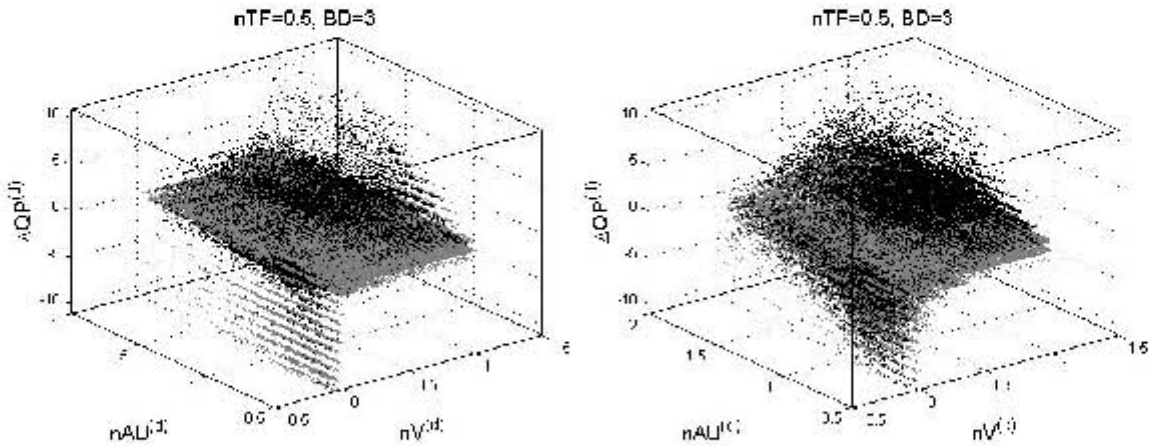
both K and NK pictures with SB labeling. Each figure was obtained for a different weight vector: Figure 6.1(a) comes from the weight vector finally selected for K pictures in validation, while Figure 6.1(b) uses the weight vector finally selected for NK pictures. As can be observed, in any case the data distributions were different enough to justify the design of two specific GPs.

2) As shown in Figures 6.2(a) and 6.2(b), the SB training data distributions for each dependency layer were similar enough to each other to justify the use of the same GP for all the layers considered. Figure 6.2(a) shows the data for K pictures and the corresponding weight vector, while Figure 6.2(b) focuses on NK pictures.

In the following two sections we describe the training and validation processes performed for both (SB and MB) GP parameter selections. However, to make the reading easier, both processes are particularized for K and NK-SB GP regression.

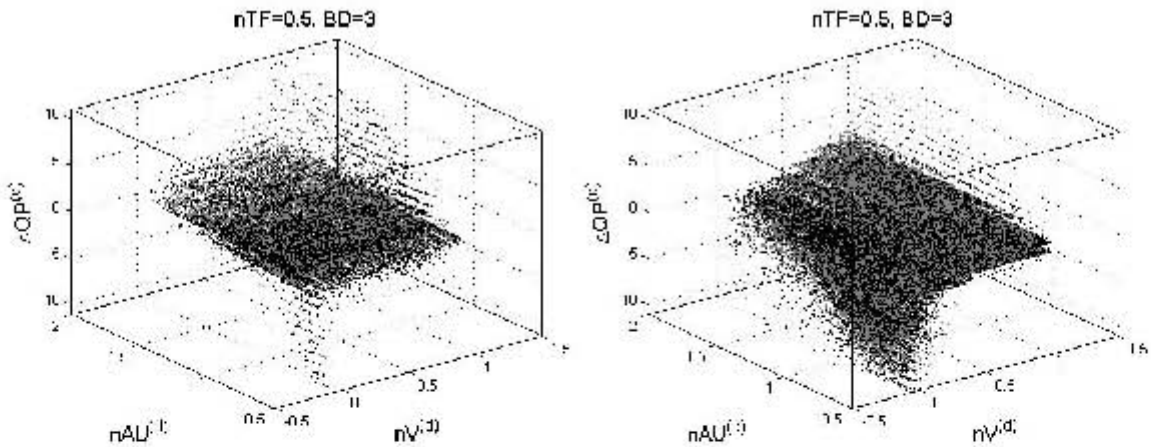
6.4 Training

Two sets of input feature vectors considering the frame type (K and NK picture) were labeled three times using the weight vectors in Table 6.1. Thus, a total of six training data sets were generated. Regression for each one was then performed using the SPGP toolbox for Matlab [Snelson, 2005]. This toolbox provides adequate values for the parameters of Equations (4.16) and (4.17) for a given data set. In particular,



(a) (b)

Figure 6.1: SB training data distributions for K pictures (black) and NK pictures (gray), with $nTF = 0.5$ and $BD = 3$. (a) Weight vector $\boldsymbol{\lambda}_a$. (b) Weight vector $\boldsymbol{\lambda}_b$. High-quality plots are available on-line in [Sanz-Rodríguez, 2011].



(a) (b)

Figure 6.2: SB training data distributions for the dependency base layer (black) and the dependency enhancement layers (gray), with $nTF = 0.5$ and $BD = 3$. (a) K pictures and weight vector $\boldsymbol{\lambda}_a$. (b) NK pictures and weight vector $\boldsymbol{\lambda}_b$. High-quality plots are available on-line in [Sanz-Rodríguez, 2011].

the weights $\mathbf{w}$, bias w_0, and hyperparameters $\{\sigma, \mathbf{b}, \mathbf{C}\}$ are selected by maximizing the marginal likelihood. This method is used in conjunction with a numerical optimization routine based on conjugate gradients to find good[2] GP settings.

Specifically, several solutions were found for each data set by training several times with different number of basis functions, specifically from $M = 4$ to $M = 16$, three random initializations[3] for the center matrix hyperparameter $\mathbf{C}$, denoted as $\mathbf{C}_0 = \{\mathbf{C}_0^1, \mathbf{C}_0^2, \mathbf{C}_0^3\}$, and 1500 iterations for GP parameter optimization process. Table 6.2 summarizes the procedure followed to generate the set of either K-SB or NK-SB GPs for validation. As can be observed in the left part of the table, 39 candidates per training data set and, therefore, 117 candidates per frame type were obtained.

6.5 Validation

The validation process was organized in two well-differenced tasks. In the first, the GP parameter selection process was conducted. In the second, a proper configuration of the post-processing stage given in Equation (4.18) was determined. Both tasks are described in detail hereunder.

6.5.1 GP Parameter Selection

The aim of this validation task was to find, for each frame type, the best combination of GP parameters $\{\boldsymbol{\lambda}, M\}$. However, because of the large amount of candidates to be assessed, a simpler but efficient search had to be performed, which consisted of two validation stages.

[2]Note that for most non-trivial GPs (which is the case), optimization over hyperparameters is not a convex problem, so the usual precautions against bad local minima should be taken.

[3]$\mathbf{b}$ is initialized to be a fixed percentage of the input data variance, and σ is set to the output data variance.

Training						Validation						
						First Stage				Second Stage		
$\boldsymbol{\lambda}$	M	$\mathbf{C}_0$	Subtotal	Subtotal	Total	$\boldsymbol{\lambda}$	M	$\mathbf{C}_0$	Total	$\{\boldsymbol{\lambda}_K, \boldsymbol{\lambda}_{NK}\}$	M	Total
$\boldsymbol{\lambda}_a$	4	$\mathbf{C}_0^1$	3	39	117	$\boldsymbol{\lambda}_a$	4	$\mathbf{C}_0^*$	39	$\{\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_a\}$	4	117
		$\mathbf{C}_0^2$									...	
		$\mathbf{C}_0^3$									16	
	...	...	...				...	...		$\{\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_b\}$	4	
	16	$\mathbf{C}_0^1$	3				16	$\mathbf{C}_0^*$			...	
		$\mathbf{C}_0^2$									7	
		$\mathbf{C}_0^3$									...	
$\boldsymbol{\lambda}_b$	4	$\mathbf{C}_0^1$	3	39		$\boldsymbol{\lambda}_b$	4	$\mathbf{C}_0^*$			16	
		$\mathbf{C}_0^2$								$\{\boldsymbol{\lambda}_a, \boldsymbol{\lambda}_c\}$	4	
		$\mathbf{C}_0^3$									...	
	...	...	...				...	...			16	
	16	$\mathbf{C}_0^1$	3				16	$\mathbf{C}_0^*$		...	...	
		$\mathbf{C}_0^2$								$\{\boldsymbol{\lambda}_c, \boldsymbol{\lambda}_a\}$	4	
		$\mathbf{C}_0^3$									...	
$\boldsymbol{\lambda}_c$	4	$\mathbf{C}_0^1$	3	39		$\boldsymbol{\lambda}_c$	4	$\mathbf{C}_0^*$			16	
		$\mathbf{C}_0^2$								$\{\boldsymbol{\lambda}_c, \boldsymbol{\lambda}_b\}$	4	
		$\mathbf{C}_0^3$									...	
	...	...	...				...	...			16	
	16	$\mathbf{C}_0^1$	3				16	$\mathbf{C}_0^*$		$\{\boldsymbol{\lambda}_c, \boldsymbol{\lambda}_c\}$	4	
		$\mathbf{C}_0^2$									...	
		$\mathbf{C}_0^3$									16	

Table 6.2: Summary of the training and validation processes for K-SB and NK-SB GP parameter selection. The best GP pair is highlighted in gray.

1. In the first stage the set of trained K-SB GPs was split into 39 subsets resulting from 13 combinations $\{\boldsymbol{\lambda}, M\}$ and three different GPs coming from different random initializations $\mathbf{C}_0$. The next step was to find, for each subset, the initialization that provided the best result in terms of marginal likelihood (available after training). Table 6.2 shows the set of GPs finally selected in this first validation stage, where $\mathbf{C}_0^*$ denotes the initialization that provided the maximum marginal likelihood. The same procedure was also performed for the NK-SB GPs.

2. The second stage aimed to find the best combination $\{\boldsymbol{\lambda}, M\}$ for both (K-SB

and NK-SB) $\Delta QP^{(d)}$ prediction models. Nevertheless, it was performed in a different way, by means of a more robust validation process, which consisted of a set of encoding tests aiming to find that pair of K-SB and NK-SB GPs that achieved the best quality consistency without incurring in overflows and underflows. It should be noticed that the $\Delta QP^{(d)}$ prediction model is a part of the RC scheme, so the overall system should be evaluated in a practical coding scenario. Furthermore, for comparison purposes, a reference RC method was used to give us an initial idea of the global performance of the proposed VBR controller. Ideally, the reference scheme should be able to solve the optimal RC problem with buffer constraints described in Section 3.2. Nevertheless, this resulted in a very high computational complexity, so we decided to use as reference the nearly constant quality method based on CQP encoding, although it does not guarantee HRD compliance.

Since the K-SB and NK-SB GPs must be jointly assessed in a real encoding and there were a lot of possible combinations $\{\boldsymbol{\lambda}_K, M_K, \boldsymbol{\lambda}_{NK}, M_{NK}\}$, a methodology to gradually discard candidate GP pairs had to be conducted to find the best pair. Specifically, three steps were followed:

2.1. The first step consisted of selecting a reduced set of candidate GP pairs, where the two GPs of each pair shared the same M value ($M_K = M_{NK}$). Thus, a total of 117 pairs of $\Delta QP^{(d)}$ prediction models would be experimentally assessed, as indicated in Table 6.2. Then, the following H.264/SVC encoder and RC configurations were used:

   a) No. of pictures: 900.
   b) GoP size/intra period: {8/8,16/16} pictures.
   c) GoP structure: Hierarchical B pictures.
   d) Search range for motion estimation: 16×16 pixels.
   e) No. of dependency layers: $D = 3$.
      i) $d = 0$: QCIF, $f_{out}^{(0,t_{max}^{(0)})} = 25$ Hz.

ii) $d = 1$: CIF, $f_{out}^{(1,t_{max}^{(1)})} = 25$ Hz.

iii) $d = 2$: 4CIF, $f_{out}^{(2,t_{max}^{(2)})} = 25$ Hz.

f) Symbol mode: CABAC.

g) RC parameters.

i) Target buffer fullness: $nTF = \{40, 50\}\%$.

ii) Buffer size: $BD = \{1.5, 3\}$ s.

The video sequences used in these experiments were: *Akiyo*, *Container*, *Hall*, *Highway*, *Ice*, *News*, *Paris*, *Silent*, *Tempete*, *Airshow*, *Ice Age*, *Cities*. In particular, the last three sequences show many scene changes, so they are challenging from the RC point of view. In order to reduce the amount of encoding tests, each sequence would be encoded using only one configuration with particular values of GoP size/intra period, nTF and BD. A CQP encoding was first executed and the resulting output bit rates were then used as target bit rates for the VBR control algorithm to be validated.

2.2. A second step was performed to find the best combination $\{\boldsymbol{\lambda}_K, \boldsymbol{\lambda}_{NK}\}$ regardless of the number M of basis functions for data modeling. To this end, a representative subset of the above video sequences, some characterized by stationary video complexity and others containing scene cuts, were encoded. The results with respect to CQP encoding in terms of quality (mean PSNR and PSNR standard deviation), buffer control and target bit rate adjustment, indicated that for K pictures the weight vector should be $\boldsymbol{\lambda}_a$, while that for NK pictures should be $\boldsymbol{\lambda}_b$. Notice that the weight vector chosen for $\Delta QP^{(d)}$ estimation in K pictures gives much more priority to the quality consistency term of the cost function given in Equation (6.2), so the resulting GP hardly modifies the QP unless the buffer fullness is very close to overflow or underflow (see Figures 4.3 and 4.4). Nevertheless, since generally most of the encoded pictures belong to

temporal enhancement layers, a more even balance between quality consistency and buffer control is provided by the weight vector selected for NK pictures.

2.3. Once the set of candidate GP pairs was reduced to 13 (one per M value), a third step consisting of 156 encoding tests (13 encodings per sequence multiplied by 12 sequences) was finally conducted to find the most appropriate number of Gaussian-type functions. In spite of that many of the candidates achieved similar encoding performance, the experimental results led us to finally select, for SB rate control, K and NK GPs composed of a linear combination of $L = 7$ Gaussian-type functions (shaded in Table 6.2).

Similar training and validation processes were performed for MB rate control, where $t_{min}^{(d)}$ was set to $t_{max}^{(d)} - 2$ so that HRD-compliant sub-streams associated with three different temporal resolutions could be provided at each dependency layer. In this case, we selected GPs composed of $L = 10$ basis functions using those training data sets labeled with the weight vectors $\boldsymbol{\lambda}_a$ and $\boldsymbol{\lambda}_b$ for K and NK pictures, respectively. One example of $\Delta QP^{(d)}$ modeling for IL-MB control has already been shown in Figure 5.3.

The resulting GP parameters are given in Appendix B.

6.5.2 Post-Processing Stage Configuration

During the parameter selection process, some unnecessary short-term QP fluctuations at NK pictures were observed in cases of stationary video complexity when the buffer level approached the target buffer fullness. Among other alternatives, the preferred solution to those fluctuations was a simple post-processing stage of the output of the GP for NK pictures that expands the input region $\left(nV^{(d)}, nAU^{(d)}\right)$ for which the output is $\Delta QP^{(d)} = 0$. However, besides the recommended solution given

in Equation (4.18), three configurations were also assessed, specifically:

$$\Delta QP_a^{(d)} = \begin{cases} -2 & \text{if } \Delta QP^{(d)} = -3 \\ -1 & \text{if } \Delta QP^{(d)} = -2 \\ 0 & \text{if } \Delta QP^{(d)} = -1 \\ 0 & \text{if } \Delta QP^{(d)} = 1 \\ 1 & \text{if } \Delta QP^{(d)} = 2 \\ 2 & \text{if } \Delta QP^{(d)} = 3, \end{cases} \tag{6.3}$$

$$\Delta QP_b^{(d)} = 0 \quad \text{if } \left|\Delta QP^{(d)}\right| = 1, \tag{6.4}$$

$$\Delta QP_c^{(d)} = 0 \quad \text{if } \left|\Delta QP^{(d)}\right| = 1 \vee \left|\Delta QP^{(d)}\right| = 2. \tag{6.5}$$

The IL-MB VBR control scheme was used as baseline RC to test each of these post-processing alternatives at the output of the NK-MB GP. In order to determine the most suitable solution, the H.264/SVC encoder and RC configurations previously used for MB GP parameter selection were performed. Finally, the experimental results indicated that the post-processing stage given in Equation (4.18) achieved the best performance in both stationary and time-varying complexity scenarios. Some representative behaviors of the encoder buffer occupancy, PSNR, and QP time evolutions corresponding to the two higher temporal resolutions of the spatial base layer are depicted, respectively, in Figures 6.3(a) and 6.3(b) for a stationary complexity sequence (*Container*), and in Figures 6.4(a) and 6.4(b) for a sequence with scene changes (*Ice Age*). In particular, these sequences were encoded with GoP size/intra period equal to 8/8 pictures, $nTF = 40\%$, and $BD = 3$ s.

According to the results shown, two conclusions were drawn: 1) for stationary complexity sequences, the proposed post-processing stage notably reduced those unnecessary short-term QP fluctuations and, therefore, improved the visual quality consistency while maintaining the buffer at secure levels; and 2) for non-stationary complexity sequences, despite reducing some QP fluctuations, the performance in terms of quality consistency and buffer control using post-processing was similar to that achieved without using post-processing.

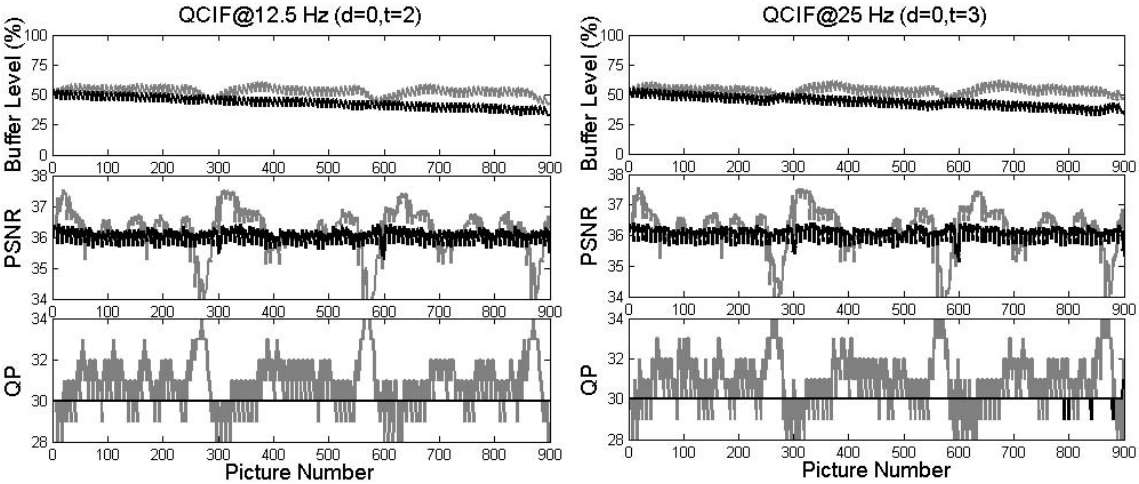
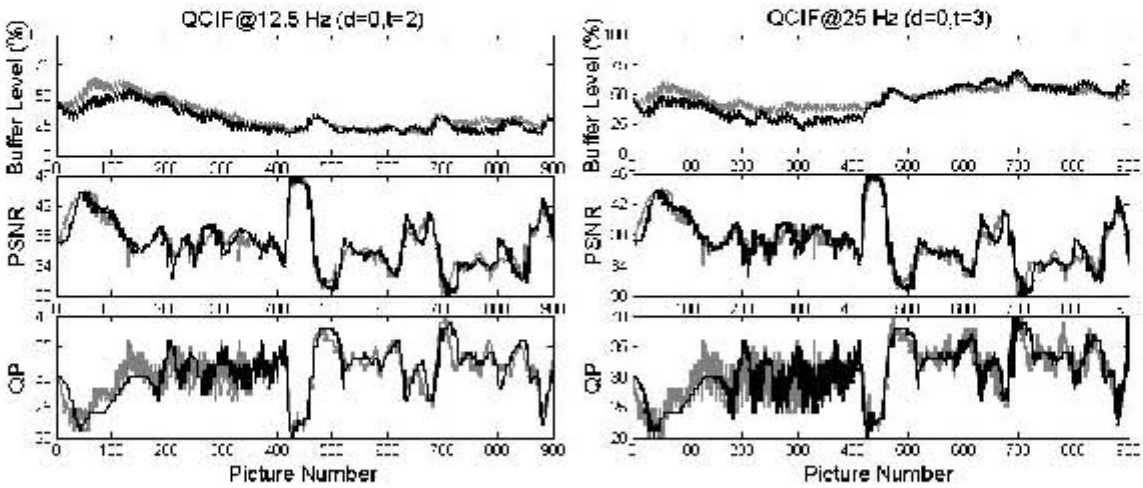


Figure 6.3: Encoder buffer occupancy, PSNR, and QP time evolutions with post-processing (black) and without post-processing (gray) for the sequence *Container*. (a) Spatio-temporal resolution: QCIF@12.5 Hz ($d = 0, t = 2$). (b) Spatio-temporal resolution: QCIF@25 Hz ($d = 0, t = 3$). High-quality plots are available on-line in [Sanz-Rodríguez, 2011].

6.6 Summary and Conclusions

The GPs we used to make $\Delta QP^{(d)}$ predictions required a carefully designed methodology able to find suitable weights, bias, and hyperparameters. Specifically, three stages were followed: training data set generation, training and validation.

The first task focused on selecting good components of the input vector to the GPs. In particular, the normalized versions of both buffer fullness and AU output bits, the target buffer occupancy and the buffer size were the four parameters that we considered essential to provide a good compromise between coding efficiency and computational complexity. Once the feature space was selected, a set of input vectors was generated and then labeled using a cost function that balanced three conflicting factors: quality consistency, buffer control, and QP consistency. Some cost function weight vectors were pre-established in order to generate several training data sets.



(a) (b)

Figure 6.4: Encoder buffer occupancy, PSNR, and QP time evolutions with post-processing (black) and without post-processing (gray) for the sequence *Ice Age*. (a) Spatio-temporal resolution: QCIF@12.5 Hz ($d = 0, t = 2$). (b) Spatio-temporal resolution: QCIF@25 Hz ($d = 0, t = 3$). High-quality plots are available on-line in [Sanz-Rodríguez, 2011].

The use of GPs for training data modeling was also justified. Particularly, GPs provide some interesting features, such as high accuracy and no overfitting, among others, thus making them suitable for our purposes. Nevertheless, since full GPs could not be directly applied when using a large amount of training samples due to their high computational cost, we had to resort to the approximate GP called SPGP to significantly reduce the training complexity. For each generated training set, several GPs were trained with different numbers of basis functions and random initializations. However, because of the large number of trained GPs, a four-step-based validation stage was performed to gradually discard unsuitable candidates.

Finally, in order to reduce unnecessary QP fluctuations that occurred in some stationary complexity situations, a post-processing stage located at the output of the NK-SB and NK-MB GPs was also carefully configured. Among other assessed

configurations, that proposed in Equation (4.18) provided the best trade-off between the performance in stationary video complexity and that achieved in time-varying situations.

It is worth noticing that the methodology described in this chapter may be adapted to solve other RC problems related to either VBR or CBR video. For instance, the training data can be modified to consider other buffer sizes required by the application; a cost function weight vector that gives more priority to the buffer control term may provide a suitable solution for a CBR application; or even more terms of the cost function can be added to consider other practical constraints. Furthermore, the process followed to generate representative training samples is not exclusively subject to a GP regression, but other methods might be applied.

Chapter 7

Conclusions and Further Work

In this chapter we summarize the contributions of the thesis and provide references to the associated papers. Then, the main conclusions are described. And finally, the chapter ends with a discussion about some interesting future lines of work.

7.1 A Summary of Contributions

The main contributions of this thesis are briefly described next:

- Since consecutive pictures within the same scene often present similar degrees of complexity, the proposed VBR controller allows for just an incremental variation of QP with respect to a reference value obtained from previously encoded pictures, thus preventing unnecessary QP fluctuations (for the sake of visual quality consistency). For this purpose, an effective method for the estimation of the required QP increment (instead of the QP value itself) has been developed. In particular, the QP increment prediction at each dependency layer in an H.264/SVC encoder is estimated by means of a regression method based on GPs, which was specially designed for this purpose from some observations drawn from a discrete set of encoding states (or samples).

This contribution has been described in Chapter 4, and has generated a conference paper [Sanz-Rodríguez and Díaz-de-María, 2010] and a journal paper [Sanz-Rodríguez and Díaz-de-María, 2011c].

- Although the RC algorithm relying on the QP increment estimation is not novel, the proposed regression method is novel. In particular, a carefully designed procedure for generating a representative set of pairs *coding state–desired QP increment* has been proposed. Furthermore, the cost function designed for data labeling is flexible enough so that it can be adapted to any other application requirements.

  A general methodology to properly design GPs for this problem has been described in Chapter 6 and published in [Sanz-Rodríguez and Díaz-de-María, 2011c].

- Temporal scalability allows for frame rate reduction by allowing the decoder to ignore the higher temporal layers within a dependency layer. However, this kind of scalability is not totally exploited by the current RC algorithms since the HRD requirement is only satisfied for the highest temporal resolution sub-stream of every dependency layer. A novel framework that aims to deliver several HRD-compliant temporal resolutions within a particular dependency layer has been proposed. Instead of using the typical SVC encoder configuration consisting of a dependency layer per temporal resolution, a compact configuration that does not require additional dependency layers for providing different HRD-compliant temporal resolutions has been proposed. Specifically, this approach for rate-controlled SVC uses a set of virtual buffers within a dependency layer so that their levels can be simultaneously controlled for overflow and underflow prevention, while minimizing the reconstructed video distortion of the corresponding sub-streams.

  This contribution has been described in Chapter 5, and has generated a conference paper [Sanz-Rodríguez et al., 2009]. Additionally, a journal paper has

been recently submitted [Sanz-Rodríguez and Díaz-de-María, 2011a].

7.2 Conclusions

In this thesis we have developed an efficient VBR controller for real-time H.264/SVC video coding applications with buffer constraints. In particular, HRD compliance and quality consistency for the highest frame rate sub-stream of every dependency layer are achieved by means of a GP regression method that is able to reduce unnecessary QP fluctuations, while maintaining the buffer at secure levels. To this end, the proposed GP regression method predicts the required QP increment with respect to a reference QP value obtained from previously encoded pictures.

Two real-time application scenarios were simulated to assess the performance of the VBR controller with respect to two well-known RC approaches: a nearly constant quality method, and a recently proposed CBR algorithm for H.264/SVC. The experimental results showed that our proposal achieved a good performance in terms of quality consistency, buffer control, and long-term adjustment to the target bit rate.

The low computational cost is another property of this VBR control algorithm since it does not need to estimate the frame complexity and, furthermore, it does not require updating any model parameter after encoding each picture. Although the QP increment prediction using GPs might involve higher complexity, we can benefit from the discrete nature of QP in H.264/SVC to implement approximated estimations based on look-up tables without coding performance degradation.

Furthermore, an IL-MB approach has been proposed that, unlike the typical SVC encoder configurations consisting of a dependency layer per each target spatio-temporal resolution, does not require additional dependency layers to deliver different HRD-compliant temporal resolutions for a given spatial resolution, thus improving the coding efficiency since, for the same target bit rate, less layers are encoded.

Therefore, when using the proposed IL-MB framework on top of the VBR con-

troller, both HRD compliance and quality consistency are also satisfied for lower frame rate sub-streams within a particular dependency layer. Specifically, the IL-MB VBR controller estimates the most appropriate QP value, on a frame basis, so that a set of virtual buffers (one per temporal resolution sub-stream) within a dependency layer is maintained at secure levels and the corresponding sub-streams produce consistent visual quality. Finally, the decision rules suggested for simultaneous control of multiple buffers within a dependency layer might also be employed on top of any baseline RC method for SVC.

7.3 Future Research Lines

Some promising lines of work opened by this thesis are briefly described in the sequel.

- In order to assess the performance of the RC algorithm described in this thesis, CQP encoding was first used to obtain the set of QP values that best approached some pre-established target bit rates, and then those QPs were used as initial QP values for the VBR controller. Nevertheless, in real-time video coding applications, it is not feasible to perform a first encoding pass to determine the desired initial QPs, so they must be previously estimated. Although some initial QP estimation algorithms for H.264/SVC have been proposed, neither of them actually takes into account the HRD constraints at the beginning of the encoding process. A research work that aims to find an effective method for initial QP estimation with buffer constraints has already been initiated [Sanz-Rodríguez and Díaz-de-María, 2011b], but further improvements are still needed.

- In order to guarantee robust performance in terms of quality consistency and buffer control, the proposed IL-MB framework requires proper target bit rates for the lower temporal resolution sub-streams to be known in advance. An effective method to estimate such target bit rates is left for future work.

- Likely two-pass video streaming would be an interesting application scenario for an IL-MB control since the first encoding pass could be used to find the optimal distribution of the total target bit rate among temporal layers.

- For simplicity, the VBR controller designed assumed that all buffers share the same values of buffer size (in seconds) and target buffer fullness. The behavior of the algorithm when using different RC parameters for each target spatio-temporal resolution might be studied as well.

Appendix A

Estimation of the Access Unit Target Bits

This appendix aims to describe in detail the proposed bit budget model for VBR control in H.264/SVC. To this end, it takes into account the following two assumptions: long-term target bit rate adjustment for the sake visual quality consistency, and buffer constraints.

In order for the sub-stream (d, k) at $f_{out}^{(d,k)}$ Hz to satisfy the target bit rate constraint $R^{(d,k)}$, the bit budget $G^{(d,t,k)}$ for an AU with layer identifier (d, t) can be expressed as follows (according to a target bit model similar to that in [Ma et al., 2003]):

$$G^{(d,t,k)} = \beta \widetilde{G}^{(d,t,k)} + (1 - \beta)\widehat{G}^{(d,t,k)}, \tag{A.1}$$

where $\widetilde{G}^{(d,t,k)}$ is a bit budget estimation based on video complexity considerations, $\widehat{G}^{(d,t,k)}$ is a bit budget estimation based on buffer constraints, and β is a constant parameter that establishes a proper balance between these two models.

The first term, $\widetilde{G}^{(d,t,k)}$, are the target bits for an AU that are estimated according

to the predicted relative complexity of the current temporal layer, i.e.:

$$\widetilde{G}^{(d,t,k)} = \underbrace{\left(\frac{R^{(d,k)}}{f_{out}^{(d,k)}} - \frac{\sum_{u=0}^{k} \left(\tilde{h}^{(d,u)} N^{(d,u)} \right)}{\sum_{u=0}^{k} N^{(d,u)}} \right)}_{A} \underbrace{\frac{\overline{C}_{TEX}^{(d,t)} \sum_{u=0}^{k} N^{(d,u)}}{\sum_{u=0}^{k} \left(\overline{C}_{TEX}^{(d,u)} N^{(d,u)} \right)}}_{B} + \tilde{h}^{(d,t)}, \quad \text{(A.2)}$$

where

- the factor denoted as A represents the nominal target texture bits for the AU, which is computed as the difference between the nominal bit budget and an average of the predicted AU header plus motion data bits corresponding to the temporal layers from 0 to k.
- The factor denoted as B is a weight factor calculated by dividing the AU texture complexity predicted for the current temporal layer, namely $\overline{C}_{TEX}^{(d,t)}$ (Equation (4.7)), by an average of the predicted AU texture complexities corresponding to the temporal layers from 0 to k.
- $\tilde{h}^{(d,t)}$ is a prediction of the header plus motion data bits for the current temporal layer. It should be noticed that $\tilde{h}^{(d,t)}$ can be viewed as a predicted AU motion complexity, namely $\overline{C}_{MOT}^{(d,t)}$ (Equation (4.8)).
- And $N^{(d,u)}$ denotes the total number of pictures per GoP with layer identifier (d, u).

The second term, $\widehat{G}^{(d,t,k)}$, represents the estimation of the target bits for the AU derived from the difference between the current buffer level $V^{(d,k)}$ and the target buffer fullness $TF^{(d,t,k)}$, i.e.:

$$\widehat{G}^{(d,t,k)} = \frac{R^{(d,k)}}{f_{out}^{(d,k)}} + \gamma \left(TF^{(d,t,k)} - V^{(d,k)} \right), \quad \text{(A.3)}$$

where

- γ is a constant that establishes a proper trade-off between the QP variation and the target buffer fullness adaptation.

- $V^{(d,k)}$ is updated using Equation (4.6).
- And $TF^{(d,t,k)}$ can be set to the target buffer fullness nTF in VBR applications.

Furthermore, the target bit model of Equations (A.1), (A.2), and (A.3) can be written as a sum of four terms:

$$G^{(d,t,k)} = G^{(d,k)}_{NOM} + \Delta G^{(d,t,k)}_{TEX} + \Delta G^{(d,t,k)}_{MOT} + \Delta G^{(d,t,k)}_{BUF},$$

where $G^{(d,k)}_{NOM}$ is the nominal bit budget, that is:

$$G^{(d,k)}_{NOM} = \frac{R^{(d,k)}}{f^{(d,k)}_{out}},$$

and $\Delta G^{(d,t,k)}_{TEX}$, $\Delta G^{(d,t,k)}_{MOT}$, and $\Delta G^{(d,t,k)}_{BUF}$ stand for texture, motion, and buffer-related bit increments, respectively, which obey the following expressions:

$$\begin{aligned}
\Delta G^{(d,t,k)}_{TEX} &= \beta \frac{R^{(d,k)}}{f^{(d,k)}_{out}} \left(\frac{\overline{C}^{(d,t)}_{TEX} \sum_{u=0}^{k} N^{(d,u)}}{\sum_{u=0}^{k} \left(\overline{C}^{(d,u)}_{TEX} N^{(d,u)} \right)} - 1 \right), \\
\Delta G^{(d,t,k)}_{MOT} &= \beta \left(\overline{C}^{(d,t)}_{MOT} - \frac{\overline{C}^{(d,t)}_{TEX} \sum_{u=0}^{k} \left(\overline{C}^{(d,u)}_{MOT} N^{(d,u)} \right)}{\sum_{u=0}^{k} \left(\overline{C}^{(d,u)}_{TEX} N^{(d,u)} \right)} \right), \\
\Delta G^{(d,t,k)}_{BUF} &= (1-\beta)\gamma \left(TF^{(d,t,k)} - V^{(d,k)} \right).
\end{aligned}$$

In VBR applications the following values for β and γ are suggested: $\beta \geq 0.9$ and $\gamma \leq 0.1$. Specifically, β was set to 1 in our experiments.

Appendix B

GP Parameters

The weights, bias and hyperparameters of Equations (4.16) and (4.17) corresponding to every GP regression method are the following:

i) K-SB GP

$$w_0 = -1.94234, \qquad \mathbf{w} = \begin{pmatrix} 5.52647 \\ 2.12748 \\ 1.05972 \\ -0.68032 \\ -4.75214 \\ -2.70089 \\ -6.01180 \end{pmatrix}, \qquad \sigma = 21.15637, \qquad \mathbf{b} = \begin{pmatrix} 4.21361 \\ 0.10821 \\ 0.37478 \\ 0.05849 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0.34878 & 2.24208 & 0.32736 & 2.57098 \\ 0.64341 & 4.02300 & 0.56932 & -4.81181 \\ 0.75362 & 1.56418 & 0.47553 & 3.07934 \\ 0.72347 & -0.25308 & -0.10081 & -0.12420 \\ -0.99480 & -0.34192 & -1.39094 & 1.72556 \\ 0.06001 & 1.14999 & 3.47226 & -2.24075 \\ 0.40772 & 2.43468 & 0.39291 & 2.68413 \end{pmatrix}.$$

ii) NK-SB GP

$$w_0 = -0.41095, \qquad \mathbf{w} = \begin{pmatrix} 73.04401 \\ -10.16582 \\ -23.92454 \\ -0.09401 \\ -67.15312 \\ 26.35348 \\ 1.65317 \end{pmatrix}, \qquad \sigma = 20.34306, \qquad \mathbf{b} = \begin{pmatrix} 2.34136 \\ 0.17469 \\ 1.66224 \\ 0.14163 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0.48170 & -0.18319 & 0.33508 & -0.20148 \\ 0.80986 & -0.12825 & 0.24415 & 0.45383 \\ 0.62855 & 0.77388 & 0.47196 & 2.75271 \\ 0.24348 & 1.16350 & 0.18820 & 2.71590 \\ 0.44971 & -0.22937 & 0.35083 & -0.19297 \\ 0.63746 & 0.66580 & 0.44850 & 2.63895 \\ 1.51031 & 1.34230 & 0.36623 & 1.02694 \end{pmatrix}.$$

iii) K-MB GP

$$w_0 = -2.11439, \qquad \mathbf{w} = \begin{pmatrix} -27.67614 \\ 0.52361 \\ 2.91606 \\ -3.49830 \\ 2.55764 \\ 0.41080 \\ 1.76009 \\ -23.30955 \\ 46.91092 \\ -2.39885 \end{pmatrix}, \qquad \sigma = 34.22354, \qquad \mathbf{b} = \begin{pmatrix} 2.32497 \\ 0.19492 \\ 1.30232 \\ 0.02554 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0.43803 & 1.27831 & 0.13142 & 2.61346 \\ 0.76851 & 1.13763 & 0.65991 & 2.79565 \\ -0.75232 & 0.79498 & 1.60194 & 1.76489 \\ -1.23805 & -0.62409 & -0.45549 & 2.01148 \\ 0.26089 & 2.77186 & 0.38882 & 0.19505 \\ 0.66948 & 3.32571 & 0.31369 & 2.04133 \\ 0.92787 & 1.04185 & -0.27238 & 1.67820 \\ 0.29267 & 1.88389 & 0.28556 & 2.79760 \\ 0.35347 & 1.49620 & 0.20293 & 2.77878 \\ -0.39515 & 0.50965 & 1.25654 & 0.14932 \end{pmatrix}.$$

iv) NK-MB GP

$$w_0 = -0.25419, \qquad \mathbf{w} = \begin{pmatrix} 794.01560 \\ -3.44210 \\ -1.92897 \\ 1.70157 \\ -0.30032 \\ -1.02440 \\ -793.73353 \\ 0.29583 \\ 0.70230 \\ 0.04244 \end{pmatrix}, \qquad \sigma = 15.75732, \qquad \mathbf{b} = \begin{pmatrix} 5.70021 \\ 0.47508 \\ 1.96225 \\ 0.22148 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0.19710 & 1.71061 & 0.12047 & 3.04580 \\ -0.67315 & -0.68530 & -0.17373 & 1.42105 \\ 0.39981 & -0.66020 & 0.89182 & -0.90448 \\ 0.58803 & 1.82533 & 0.24637 & -0.95955 \\ 0.66092 & 0.77316 & 0.57093 & 3.35614 \\ 0.70296 & 1.74486 & -0.15198 & 0.65384 \\ 0.19696 & 1.71090 & 0.12112 & 3.04637 \\ 0.88774 & 0.42078 & 0.61288 & 1.74001 \\ 0.92236 & 2.50876 & 0.15902 & 2.95167 \\ -0.12642 & 0.67930 & 0.67757 & 1.23198 \end{pmatrix}.$$

Bibliography

[Amon et al., 2007] Amon, P., Rathgen, T., and Singer, D. (2007). File format for scalable video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1174–1185.

[Amonou et al., 2007] Amonou, I., Cammas, N., Kervadec, S., and Pateux, S. (2007). Optimized rate-distortion extraction with quality layers in the scalable extension of H.264/AVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1186–1193.

[Anselmo and Alfonso, 2007] Anselmo, T. and Alfonso, D. (2007). Buffer-based constant bit-rate control for scalable video coding. In *Picture Coding Symposium, 2007. PCS 2007.*

[Bai et al., 2002] Bai, J., Liao, Q., Lin, X., and Zhuang, X. (2002). Rate-distortion model based rate control for real-time VBR video coding and low-delay communications. *Signal Processing: Image Communication*, 17(2):187–199.

[Berger, 1971] Berger, T. (1971). *Rate distortion theory: A mathematical basis for data compression.* Prentice-Hall, Englewood Cliffs, NY.

[Bjøntegaard, 2001] Bjøntegaard, G. (2001). Calculation of average PSNR differences between RD curves. *VCEG contribution, VCEG-M33, Austin.*

[Chen and Ngan, 2007a] Chen, Z. and Ngan, K. N. (2007a). Recent advances in rate control for video coding. volume 22, pages 19–38, New York, NY, USA. Elsevier Science Inc.

[Chen and Ngan, 2007b] Chen, Z. and Ngan, K. N. (2007b). Towards rate-distortion tradeoff in real-time color video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(2):158–167.

[Chiang and Zhang, 1997] Chiang, T. and Zhang, Y.-Q. (1997). A new rate control scheme using quadratic rate distortion model. *Circuits and Systems for Video Technology, IEEE Transactions on*, 7(1):246–250.

[Cho et al., 2009] Cho, Y., Liu, J., Kwon, D.-K., and Kuo, C.-C. (2009). Joint quality-temporal (Q-T) bit allocation for H.264/SVC. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 2361–2364.

[Czuni et al., 2006] Czuni, L., Csaszar, G., and Licsar, A. (2006). Estimating the optimal quantization parameter in H.264. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 330–333.

[Dai et al., 2003] Dai, M., Loguinov, D., and Radha, H. (2003). Statistical analysis and distortion modeling of MPEG-4 FGS. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, pages III–301–4.

[Dai et al., 2006] Dai, M., Loguinov, D., and Radha, H. (2006). Rate-distortion analysis and quality control in scalable internet streaming. *Multimedia, IEEE Transactions on*, 8(6):1135–1146.

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

[Ding, 1997] Ding, W. (1997). Joint encoder and channel rate control of VBR video over ATM networks. *Circuits and Systems for Video Technology, IEEE Transactions on*, 7(2):266–278.

[Ding and Liu, 1996] Ding, W. and Liu, B. (1996). Rate control of MPEG video coding and recording by rate-quantization modeling. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(1):12–20.

[Eleftheriadis et al., 2006] Eleftheriadis, A., Civanlar, M., and Shapiro, O. (2006). Multipoint videoconferencing with scalable video coding. *Journal of Zhejiang University - Science A*, 7:696–705. 10.1631/jzus.2006.A0696.

[Hang and Chen, 1997] Hang, H.-M. and Chen, J.-J. (1997). Source model for transform video coder and its application. I. Fundamental theory. *Circuits and Systems for Video Technology, IEEE Transactions on*, 7(2):287–298.

[He et al., 2001] He, Z., Kim, Y. K., and Mitra, S. (2001). Low-delay rate control for DCT video coding via ρ-domain source modeling. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(8):928–940.

[He and Mitra, 2001] He, Z. and Mitra, S. (2001). A unified rate-distortion analysis framework for transform coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(12):1221–1236.

[Huynh-Thu and Ghanbari, 2008] Huynh-Thu, Q. and Ghanbari, M. (2008). Temporal aspect of perceived quality in mobile video broadcasting. *Broadcasting, IEEE Transactions on*, 54(3):641–651.

[ISO/IEC, 1993] ISO/IEC (1993). MPEG Test Model 5. *ISO/IEC JTC/SC29/WG11, MPEG Test Model 5.*

[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.*

[ISO/IEC, 1999] ISO/IEC (1999). Coding of audio-visual objects - Part 2: Visual. *ISO/IEC 14496-2, MPGEG-4 Visual Version 1.*

[Itti, 2004] Itti, L. (2004). Automatic foveation for video compression using a neurobiological model of visual attention. *Image Processing, IEEE Transactions on*, 13(10):1304–1318.

[ITU-T, 1995] ITU-T (1995). Video coding for low bitrate communication. *ITU-T Draft Recommendation H.263 Version 1.*

[Jagmohan and Ratakonda, 2003] Jagmohan, A. and Ratakonda, K. (2003). MPEG-4 one-pass VBR rate control for digital storage. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(5):447–452.

[Jiang and Ling, 2006] Jiang, M. and Ling, N. (2006). Low-delay rate control for real-time H.264/AVC video coding. *Multimedia, IEEE Transactions on*, 8(3):467–477.

[Jing and Chau, 2006] Jing, X. and Chau, L.-P. (2006). A novel intra-rate estimation method for H.264 rate control. In *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on.*

[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 et al., 2005] Kamaci, N., Altunbasak, Y., and Mersereau, R. (2005). Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion models. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(8):994–1006.

[Kim, 2003] Kim, H. M. (2003). Adaptive rate control using nonlinear regression. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(5):432–439.

[Kim et al., 1999] Kim, W. J., Yi, J. W., and Kim, S. D. (1999). A bit allocation method based on picture activity for still image coding. *Image Processing, IEEE Transactions on*, 8(7):974–977.

[Kim et al., 2001] Kim, Y. K., He, Z., and Mitra, S. (2001). A novel linear source model and a unified rate control algorithm for H.263/MPEG-2/MPEG-4. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 3, pages 1777–1780.

[Kwon et al., 2007] Kwon, D.-K., Shen, M.-Y., and Kuo, C.-C. J. (2007). Rate control for H.264 video with enhanced rate and distortion models. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(5):517–529.

[Lakshman et al., 1998] Lakshman, T., Ortega, A., and Reibman, A. (1998). VBR video: tradeoffs and potentials. *Proceedings of the IEEE*, 86(5):952–973.

[Lam and Goodman, 2000] Lam, E. and Goodman, J. (2000). A mathematical analysis of the DCT coefficient distributions for images. *Image Processing, IEEE Transactions on*, 9(10):1661–1666.

[Lázaro-Gredilla, 2010] Lázaro-Gredilla, M. (2010). *Sparse Gaussian Processes for Large-Scale Machine Learning*. PhD thesis, Universidad Carlos III de Madrid.

[Lee et al., 2010] Lee, H., Lee, Y., Lee, D., Lee, J., and Shin, H. (2010). Implementing rate allocation and control for real-time H.264/SVC encoding. In *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on*, pages 269–270.

[Lee et al., 2000] Lee, H.-J., Chiang, T., and Zhang, Y.-Q. (2000). Scalable rate control for MPEG-4 video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 10(6):878–894.

[Leontaris and Tourapis, 2007] Leontaris, A. and Tourapis, A. M. (2007). Rate control for the Joint Scalable Video Model (JSVM). *Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-W043, San Jose, California.*

[Li, 2001] Li, W. (2001). Overview of fine granularity scalability in MPEG-4 video standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(3):301–317.

[Liebl et al., 2004] Liebl, G., Wagner, M., Pandel, J., and Weng, W. (2004). An RTP payload format for erasure-resilient transmission of progressive multimedia streams. *Document draft-ietf-avt-uxp-07.txt.*

[Lin and Ortega, 1998] Lin, L.-J. and Ortega, A. (1998). Bit-rate control using piecewise approximated rate-distortion characteristics. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(4):446–459.

[Liu et al., 2010a] Liu, J., Cho, Y., Guo, Z., and Kuo, J. (2010a). Bit allocation for spatial scalability coding of H.264/SVC with dependent rate-distortion analysis. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(7):967–981.

[Liu et al., 2010b] Liu, M., Guo, Y., Li, H., and Chen, C.-W. (2010b). Low-complexity rate control based on ρ-domain model for scalable video coding. In *Image Processing, 2010. ICIP 2010. IEEE International Conference on*, pages 1277–1280.

[Liu et al., 2008] Liu, Y., Li, Z. G., and Soh, Y. C. (2008). Rate control of H.264/AVC scalable extension. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(1):116–121.

[Ma et al., 2005] Ma, S., Gao, W., and Lu, Y. (2005). Rate-distortion analysis for H.264/AVC video coding and its application to rate control. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(12):1533–1544.

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

[Mansour et al., 2008] Mansour, H., Krishnamurthy, V., and Nasiopoulos, P. (2008). Rate and distortion modeling of medium grain scalable video coding. In *Image Processing, 2008. ICIP 2008. IEEE International Conference on*, pages 2564–2567.

[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.

[Mohsenian et al., 1999] Mohsenian, N., Rajagopalan, R., and Gonzales, C. A. (1999). Single-pass constant- and variable-bit-rate MPEG-2 video compression. *IBM Journal of Research and Development*, 43(4):489–509.

[Moscheni et al., 1993] Moscheni, F., Dufaux, F., and Nicolas, H. (1993). Entropy criterion for optimal bit allocation between motion and prediction error information. In *in SPIE Proc. on Visual Communications and Image Processing, Boston, USA*, volume 2094 of *Virtual worlds and multimedia*, pages 235–242. IEEE.

[Naish-Guzman and Holden, 2008] Naish-Guzman, A. and Holden, S. B. (2008). The generalized FITC approximation. In *Advances in Neural Information Processing Systems 20*, pages 1057–1064. MIT Press.

[Ortega, 2000] Ortega, A. (2000). Variable bit-rate video coding. *in Compressed Video over Networks, M.-T. Sun and A. R. Reibman, Eds. New York: Marcel Dekker*, pages 343–382.

[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.

[Pitrey et al., 2009] Pitrey, Y., Babel, M., and Deforges, O. (2009). One-pass bitrate control for MPEG-4 scalable video coding using ρ-domain. *Broadband Multimedia*

Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on, pages 1–5.

[Ramchandran et al., 1994] Ramchandran, K., Ortega, A., and Vetterli, M. (1994). Bit allocation for dependent quantization with applications to multiresolution and mpeg video coders. *Image Processing, IEEE Transactions on*, 3(5):533–545.

[Rasmussen, 1996] Rasmussen, C. E. (1996). *Evaluation of Gaussian Processes and other Methods for Non-linear Regression*. PhD thesis, University of Toronto.

[Rasmussen, 2003] Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

[Rezaei, 2008] Rezaei, M. (2008). *Advances on video coding algorithms for straming applications*. PhD thesis, Tampere University of Technology.

[Rezaei et al., 2008] Rezaei, M., Hannuksela, M., and Gabbouj, M. (2008). Semi-fuzzy rate controller for variable bit rate video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(5):633–645.

[Ribas-Corbera et al., 2003] Ribas-Corbera, J., Chou, P., and Regunathan, S. (2003). A generalized hypothetical reference decoder for H.264/AVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):674–687.

[Ribas-Corbera and Lei, 1999] Ribas-Corbera, J. and Lei, S. (1999). Rate control in DCT video coding for low-delay communications. *Circuits and Systems for Video Technology, IEEE Transactions on*, 9(1):172–185.

[Ronda et al., 1999] Ronda, J., Eckert, M., Jaureguizar, F., and Garcia, N. (1999). Rate control and bit allocation for MPEG-4. *Circuits and Systems for Video Technology, IEEE Transactions on*, 9(8):1243–1258.

[Sanz-Rodríguez, 2011] Sanz-Rodríguez, S. (2011). High-quality plots and sequences, [On-Line] http://www.tsc.uc3m.es/∼sescalona/thesis/.

[Sanz-Rodríguez et al., 2007a] Sanz-Rodríguez, S., de-Frutos-López, M., González-Díaz, I., and Cid-Sueiro, J. (2007a). A rate control algorithm for low-delay H.264 video coding with stored-B pictures. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 1, pages I–1153–I–1156.

[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.

[Sanz-Rodríguez and Díaz-de-María, 2010] Sanz-Rodríguez, S. and Díaz-de-María, F. (2010). RBF-based VBR controller for real-time H.264/SVC video coding. In *Picture Coding Symposium (PCS), 2010*, pages 410–413.

[Sanz-Rodríguez and Díaz-de-María, 2011a] Sanz-Rodríguez, S. and Díaz-de-María, F. (2011a). In-layer multi-buffer framework for rate-controlled Scalable Video Coding. *Submitted to Circuits and Systems for Video Technology, IEEE Transactions on.*

[Sanz-Rodríguez and Díaz-de-María, 2011b] Sanz-Rodríguez, S. and Díaz-de-María, F. (2011b). Rate control initialization algorithm for scalable video coding. In *Image Processing, 2011. ICIP 2011. IEEE International Conference on.*

[Sanz-Rodríguez and Díaz-de-María, 2011c] Sanz-Rodríguez, S. and Díaz-de-María, F. (2011c). RBF-based QP estimation model for VBR control in H.264/SVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, In Press.

[Sanz-Rodríguez et al., 2009] Sanz-Rodríguez, S., Díaz-de-María, F., and Rezaei, M. (2009). Low-complexity VBR controller for spatial-CGS and temporal scalable video coding. In *Picture Coding Symposium, 2009. PCS 2009*, pages 1–4.

[Sanz-Rodríguez et al., 2007b] Sanz-Rodríguez, S., García-García, D., de-Frutos-López, M., and Cid-Sueiro, J. (2007b). Dynamic basic unit size in rate control for real-time H.264 video coding. In *Picture Coding Symposium, 2007. PCS 2007.*

[Schaefer et al., 2005] Schaefer, R., Schwarz, H., Marpe, D., Schierl, T., and Wiegand, T. (2005). MCTF and scalability extension of H.264/AVC and its application to video transmission, storage, and surveillance. In *Proceedings of VCIP 2005, Peking, China*, pages 596 011: 1–12.

[Schuster et al., 1999] Schuster, G. M., Melnikov, G., and Katsaggelos, A. K. (1999). A review of the minimum maximum criterion for optimal bit allocation among dependent quantizers. *IEEE Trans. Multimedia*, 1:3–17.

[Schwarz et al., 2007] Schwarz, H., Marpe, D., and Wiegand, T. (2007). Overview of the scalable video coding extension of the H.264/AVC standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1103–1120.

[Seo et al., 2010] Seo, C.-W., Kang, J. W., Han, J.-K., and Nguyen, T. (2010). Efficient bit allocation and rate control algorithms for hierarchical video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(9):1210–1223.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Communication, Bell System Technical Journal*, 27:379–423.

[Shannon, 1959] Shannon, C. E. (1959). Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record, Part 4*, 7:142–163.

[Shoham and Gersho, 1988] Shoham, Y. and Gersho, A. (1988). Efficient bit allocation for an arbitrary set of quantizers [speech coding]. *Acoustics, Speech, and*

Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on, 36(9):1445–1453.

[Smoot and Rowe, 1996] Smoot, S. and Rowe, L. A. (1996). Study of DCT coefficient distributions. In *Proceedings of the SPIE Symposium on Electronic Imaging*, volume 2657, pages 403–411.

[Snelson, 2005] Snelson, E. (2005). Matlab code for sparse pseudo-input Gaussian processes (SPGP), [On-Line] http://www.gatsby.ucl.ac.uk/ snelson/.

[Snelson and Ghahramani, 2006] Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1259–1266. MIT Press.

[Sun et al., 2008] Sun, Y., Zhou, Y., Feng, Z., and He, Z. (2008). A novel incremental rate control scheme for H.264 video coding. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1612–1615.

[Tang et al., 2006] Tang, C.-W., Chen, C.-H., Yu, Y.-H., and Tsai, C.-J. (2006). Visual sensitivity guided bit allocation for video coding. *Multimedia, IEEE Transactions on*, 8(1):11–18.

[Tao et al., 2000] Tao, B., Dickinson, B., and Peterson, H. (2000). Adaptive model-driven bit allocation for MPEG video coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 10(1):147–157.

[Unterweger and Thoma, 2007] Unterweger, A. and Thoma, H. (2007). The influence of bit rate allocation to scalability layers on video quality in H.264 SVC. In *Picture Coding Symposium, 2007. PCS 2007.*

[Vetro et al., 2003] Vetro, A., Wang, Y., and Sun, H. (2003). Rate-distortion modeling for multiscale binary shape coding based on markov random fields. *Image Processing, IEEE Transactions on*, 12(3):356–364.

[Vieron et al., 2007] Vieron, J., Wien, M., and Schwarz, H. (2007). JSVM 11 software. *24th Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG Meeting, Geneva, Doc. JVT-X203*.

[Wang and Kwong, 2008] Wang, H. and Kwong, S. (2008). Rate-distortion optimization of rate control for H.264 with adaptive initial quantization parameter determination. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(1):140–144.

[Wang et al., 2007] Wang, Y.-K., Hannuksela, M., Pateux, S., Eleftheriadis, A., and Wenger, S. (2007). System and transport interface of SVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1149–1163.

[Wenger et al., 2006] Wenger, S., Wang, Y.-k., and Hannuksela, M. (2006). RTP payload format for H.264/SVC scalable video coding. *Journal of Zhejiang University - Science A*, 7:657–667.

[Westerink et al., 1999] Westerink, P. H., Rajagopalan, R., and Gonzales, C. A. (1999). Two-pass MPEG-2 variable-bit-rate encoding. *IBM Journal of Research and Development*, 43(4):471–488.

[Wiegand et al., 2009] Wiegand, T., Noblet, L., and Rovati, F. (2009). Scalable video coding for IPTV services. *Broadcasting, IEEE Transactions on*, 55:527–538.

[Wiegand et al., 2003] Wiegand, T., Sullivan, G., Bjntegaard, G., and Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560–576.

[Wiegand et al., 2007] Wiegand, T., Sullivan, G., Reichel, J., Schwarz, H., and Wien, M. (2007). Joint Draft ITU-T Rec. H.264 — ISO/IEC 14496-10 / Amd.3 Scalable Video Coding. *24th Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG Meeting, Geneva, Doc. JVT-X201*.

[Wien et al., 2007a] Wien, M., Cazoulat, R., Graffunder, A., Hutter, A., and Amon, P. (2007a). Real-time system for adaptive video streaming based on SVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1227–1237.

[Wien and Schwarz, 2005] Wien, M. and Schwarz, H. (2005). Testing conditions for SVC coding efficiency and JSVM performance evaluation. *16th Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG Meeting, Poznan, Doc. JVT-Q205.*

[Wien et al., 2007b] Wien, M., Schwarz, H., and Oelbaum, T. (2007b). Performance analysis of SVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(9):1194–1203.

[Wu and Kim, 2009] Wu, W. and Kim, H. K. (2009). A novel rate control initialization algorithm for H.264. *Consumer Electronics, IEEE Transactions on*, 55(2):665–669.

[Xie and Zeng, 2006] Xie, B. and Zeng, W. (2006). A sequence-based rate control framework for consistent quality real-time video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 16(1):56–71.

[Xiph.org, 2011] Xiph.org (2011). Test Video Sequences, [on-line] http://media.xiph.org/video/derf/.

[Xu et al., 2007] Xu, L., Gao, W., Ji, X., Zhao, D., and Ma, S. (2007). Rate control for spatial scalable coding in SVC. In *Picture Coding Symposium, 2007. PCS 2007.*

[Yang et al., 2010] Yang, J., Sun, Y., Kline, C., and Sun, S. (2010). Adaptive initial quantization parameter selection for H.264/SVC rate control. In *Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on*, volume 2, pages 723–726.

[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 313–316.

[Yu et al., 1998] Yu, Y., Zhou, J., and Wang, Y. (1998). A fast effective scene change detection and adaptive rate control algorithm. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 2, pages 379–382.

[Yu et al., 2001] Yu, Y., Zhou, J., Wang, Y., and Chen, C. W. (2001). A novel two-pass VBR coding algorithm for fixed-size storage application. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(3):345–356.

[Zhai and Katsaggelos, 2007] Zhai, F. and Katsaggelos, A. K. (2007). Joint source-channel video transmission. *Synthesis Lectures on Image, Video, and Multimedia Processing*, 3:1–136.

[Zhang et al., 2003] Zhang, X. M., Vetro, A., Shi, Y., and Sun, H. (2003). Constant quality constrained rate allocation for FGS-coded video. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(2):121–130.

[Zhao et al., 2002] Zhao, L., Kim, J., and c. Jay Kuo, C. (2002). MPEG-4 FGS video streaming with constant-quality rate control and differentiated forwarding. In *Control and Differentiated Forwarding, Visual Communications and Image Processing 2002, Proceedings of SPIE*, pages 230–241.