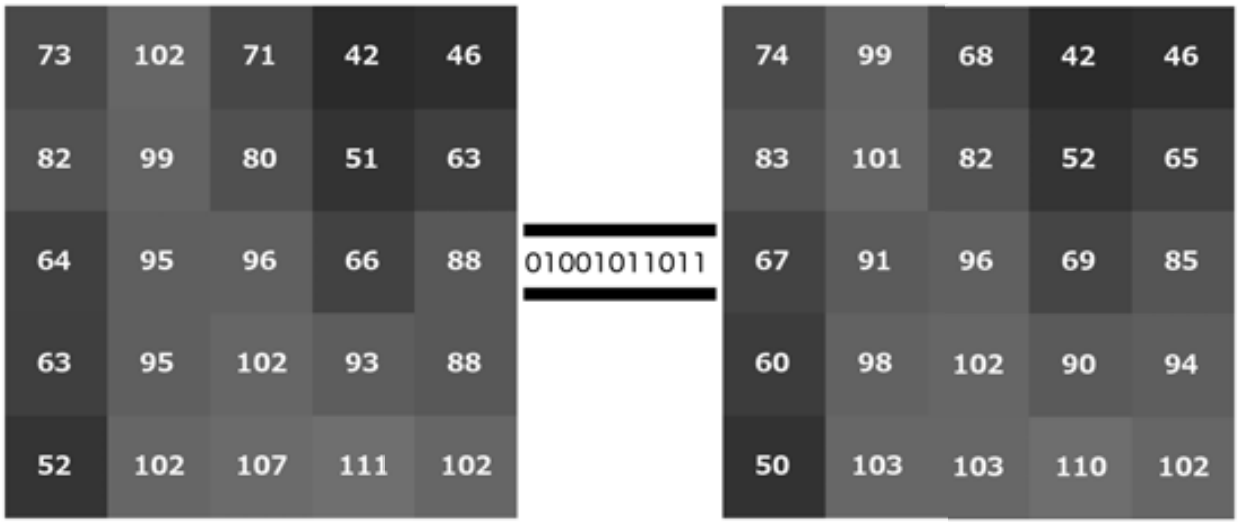
Universitat Autònoma de Barcelona
Escola d'Enginyeria
Departament d'Enginyeria de la Informació i Comunicació

Interactive transmission and visually lossless strategies for JPEG2000 imagery

By Leandro Jiménez Rodríguez

Supervised by Francesc Aulí Llinàs



Submitted to Universitat Autònoma de Barcelona in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Bellaterra, April 2014

Universitat Autònoma de Barcelona

Departament d'Enginyeria de la Informació i de les Comunicacions

Interactive transmission and visually lossless strategies for JPEG2000 imagery

Submitted to Universitat Autònoma de Barcelona
in partial fulfillment of the requirements for the
degree of Doctor of Philosophy in Computer Science

by Leandro Jiménez Rodríguez

Bellaterra, April 2014

Supervisor:
Dr. Francesc Aulí Llinàs

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Bellaterra, April 2014

Dr. Francesc Aulí Llinàs
(Principal Adviser)

Committee:

Dr. Victor Francisco Sanchez Silva
Dr. Joan Bartrina Rapesta
Dr. Valero Laparra Pérez-Muelas
Dr. David Megías Jiménez (substitute)
Dr. Javier Ruiz Hidalgo (substitute)

To those who helped me to
make it possible.

Abstract

Every day, videos and images are transmitted over the Internet. Image compression allows to reduce the total amount of data transmitted and accelerates the delivery of such data. In video-on-demand scenarios, the video has to be transmitted as fast as possible employing the available channel capacity. In such scenarios, image compression is mandatory for faster transmission. Commonly, videos are coded allowing quality loss in every frame, which is referred to as lossy compression. Lossy coding schemes are the most used for Internet transmission due its high compression ratios. Another key feature in video-on-demand scenarios is the channel capacity. Depending on the capacity a rate allocation method decides the amount of data that is transmitted for every frame. Most rate allocation methods aim at achieving the best quality for a given channel capacity. In practice, the channel bandwidth may suffer variations on its capacity due traffic congestion or problems in its infrastructure. This variations may cause buffer under-/over-flows in the client that causes pauses while playing a video. The first contribution of this thesis is a JPEG2000 rate allocation method for time-varying channels. Its main advantage is that allows fast processing achieving transmission quality close to the optimal. Although lossy compression is the most used to transmit images and videos in Internet, when image quality loss is not allowed, lossless compression schemes must be used. Lossless compression may not be suitable in scenarios due its lower compression ratios. To overcome this drawback, visually lossless coding regimes can be used. Visually lossless compression is a technique based in the human visual system to encode only the visually relevant data of an image. It allows higher compression ratios than lossless compression achieving losses that are not perceptible to the human eye. The second contribution of this thesis is a visually lossless coding scheme aimed at JPEG2000 imagery that is already coded. The proposed method permits the decoding and/or transmission of images in a visually lossless regime.

Cada día, vídeos e imágenes se transmiten por Internet. La compresión de imágenes permite reducir la cantidad total de datos transmitidos y acelera su entrega. En escenarios de vídeo-bajo-demanda, el vídeo debe transmitirse lo más rápido posible usando la capacidad disponible del canal. En éstos escenarios, la compresión de imágenes es mandataria para transmitir lo más rápido posible. Comúnmente, los videos son codificados permitiendo pérdida de calidad en los fotogramas, lo que se conoce como compresión con pérdida. Los métodos de compresión con pérdida son los más usados para transmitir por Internet dados sus elevados factores de compresión. Otra característica clave

en escenarios de video-bajo-demanda es la capacidad del canal. Dependiendo de la capacidad, un método de asignación de ratio asigna la cantidad de datos que deben ser transmitidos por cada fotograma. La mayoría de estos métodos tienen como objetivo conseguir la mejor calidad posible dado un ancho de banda. A la práctica, el ancho de banda del canal puede sufrir variaciones en su capacidad debido a congestión en el canal o problemas en su infraestructura. Estas variaciones pueden causar el desbordamiento o vaciado del buffer del cliente, provocando pausas en la reproducción del vídeo. La primera contribución de esta tesis es un método de asignación de ratio basado en JPEG2000 para canales variantes en el tiempo. Su principal ventaja es el procesado rápido consiguiendo una calidad casi óptima en la transmisión. Aunque la compresión con pérdida sea la más usada para la transmisión de imágenes y vídeos por Internet, hay situaciones donde la pérdida de calidad no está permitida, en éstos casos la compresión sin pérdida debe ser usada. La compresión sin pérdida puede no ser viable en escenarios debido sus bajos factores de compresión. Para superar este inconveniente, la compresión visualmente sin pérdida puede ser usada. La compresión visualmente sin pérdida es una técnica que está basada en el sistema de visión humano para codificar sólo los datos de una imagen que son visualmente relevantes. Esto permite mayores factores de compresión que en la compresión sin pérdida, consiguiendo pérdidas no perceptibles al ojo humano. La segunda contribución de esta tesis es un sistema de codificación visualmente sin pérdida para imágenes JPEG2000 que ya han sido codificadas previamente. El propósito de este método es permitir la decodificación y/o transmisión de imágenes en un régimen visualmente sin pérdida.

Cada dia, vídeos i imatges es transmeten per Internet. La compressió d'imatges permet reduir la quantitat total de dades transmeses i accelera la seva entrega. En escenaris de vídeo-sota-demanda, el vídeo s'ha de transmetre el més ràpid possible utilitzant la capacitat disponible del canal. En aquests escenaris, la compressió d'imatges es mandatària per transmetre el més ràpid possible. Comunament, els vídeos són codificats permeten pèrdua de qualitat en els fotogrames, el que es coneix com ha compressió amb pèrdua. Els mètodes de compressió amb pèrdua són els més utilitzats per transmetre per Internet degut als seus elevats factors de compressió. Un altre característica clau en els escenaris de vídeo-sota-demanda és la capacitat del canal. Dependent de la capacitat, un mètode de assignació de rati assigna la quantitat de dades que es deu transmetre per cada fotograma. La majoria d'aquests mètodes tenen com a objectiu aconseguir la millor qualitat possible donat un ample de banda. A la pràctica, l'ample de banda del canal pot sofrir variacions en la seva capacitat degut a la congestió en el canal o problemes en la seva infraestructura. Aquestes variacions poden causar el desbordament o buidament del buffer del client, provocant pauses en la reproducció del vídeo. La primera contribució d'aquesta tesis es un mètode d'assignació de rati basat en JPEG2000 per a canals variants en el temps. La seva principal avantatja és el ràpid processament aconseguint una qualitat quasi òptima en la transmissió. Encara que la compressió amb pèrdua sigui la més usada per la transmissió d'imatges i vídeo per Internet, hi ha situacions on la pèrdua de qualitat no està permesa, en aquests casos la compressió sense pèrdua ha de ser utilitzada. La compressió sense pèrdua pot no ser viable en alguns escenaris degut als seus baixos factors de compressió. Per superar aquest inconvenient, la compressió visualment sense pèrdua pot ser utilitzada. La compressió

visualment sense pèrdua és una tècnica basada en el sistema de visió humana per codificar només les dades visualment rellevants. Això permet factors de compressió majors que els de la compressió sense pèrdua, aconseguint pèrdues no perceptibles a l'ull humà. La segona contribució d'aquesta tesis és un sistema de codificació visualment sense pèrdua per a imatges JPEG2000 prèviament codificades. El propòsit d'aquest mètode es permetre la descodificació i/o transmissió de imatges dins en un règim visualment sense pèrdua.

Acknowledgements

This work couldn't be finished without the help of many people, just being there or helping me out with the research.

First I would like to thank all my family and friends for be there unconditionally and helped me out at some part of this research.

Also, I wish to thank all the members and former members of the Group on Interactive Coding of Images that I had the pleasure to met. The have helped me the most, sharing their experience, giving ideas and having good times.

I would like to thank professor Michael W. Marcellin, thanks to his experience, guidance and knowledge this research had been completed. I'm also very grateful to him for being a great host at the University of Arizona.

And finally I want to thank Francesc. Without his help, patience, guidance and great ideas this dissertation couldn't be possible. I'm very glad to have him as a advisor, he did an excellent work. Thanks for all your help.

Thank you all again.

Contents

Chapter 1

Introduction

1.1 Introduction

Nowadays, millions of images and video sequences are captured, stored and transmitted using mobile phones via the Internet. Also, services of video streaming are becoming more popular. The streaming of videos, requires that the transmission and processing of the images as fast as possible given the bandwidth of the network. In such scenario, image compression is helpful since it reduces the amount of data that have to be transmitted.

Compression also improves the quality of the video transmission in scenarios with a network with limited capacity. In addition to compression, the transmission of video sequences requires an algorithm that decides the amount of data that is delivered for every frame. The main objective of most transmission algorithms is to achieve the best image quality given a limited network capacity. In practice, most internet connections has fluctuations in its capacity over time due traffic congestion or problems related with the infrastructure. This may cause pauses when playing streamed video since some algorithms may not take into account this situation.

The quality achieved by transmission algorithms is related with the compression scheme that they employ. The most common coding schemes for image compression are lossless and lossy compression. Lossless compression does not produce numerical losses in the recovered image, but the compression ratios that it achieves aren't

high. On the other hand, lossy compression, allows high compression ratios at the expense of some quality losses. Most transmission schemes use lossy compression due its high compression ratios, and fastest transmissions and processing than lossless compression.

Recently, a coding scheme that fits between lossy and lossless compression has appeared. It is commonly referred to as visually lossless since it produces images that have no distortion visible to the human eye, though they may contain numerical differences with the original image. This allows the achievement of high compression factors.

1.2 Motivation

There exist two main coding systems for video coding: interframe, which exploits dependencies between frames, and intraframe, which only exploits dependencies in the same frame. The most advanced compression standards for interframe and intraframe are H.264/AVC[h264] and JPEG2000[j2k], respectively.

This thesis proposes an approach for video transmission that handles time-varying channels when transmitting video coded with the JPEG2000 standard. JPEG2000 standard was approved by the Joint Photographic Experts Group committee in 2000. This standard consists in thirteen parts. The relevant parts for video coding and transmission are Part 1, Part 3, and Part 9. Part 1 is the Core coding system, which defines the coding scheme and the syntax of the codestream. Part 3 defines the file syntax of JPEG2000 video. Part 9 describes the interactive transmission of images, also known as JPEG2000 interative protocol (JPIP).

In addition to the transmission of video in time-varying channels, this work also aims at improving the visual quality of images using visually lossless techniques. We deal with the problem of images that are already encoded and has to be transmitted using a visually lossless mode without re-encoding them. We do not want to re-encode the images because this could be a drawback for large image repositories. In this thesis we present a method able to decode and transmit JPEG2000 imagery in a visually lossless mode.

1.3 Contributions

The first contribution of this thesis is an algorithm that assigns the right amount of data to every frame. The proposed method is based on the method called FAst rate allocation through STeepest descent (FAST) algorithm, which achieves near optimal results when transmitting video but does not consider changes in the channel bandwidth. Our contribution is an adaptation of the FAST algorithm that considers possible changes in the channel capacity and prevents buffer under-/over-flows.

Our second objective is to improve the visual quality of transmitted JPEG2000 images. This objective has been divided in two tasks. First, the visually lossless scheme proposed in by H. Oh, A. Bilgin, and M. Marcellin has been adapted in scenarios where images have already been encoded without visually lossless methods. To do so, a visually lossless decoder is proposed. It decodes the image until reaching a specific threshold that indicates that all the visually relevant data are decoded. By analyzing images from the visually lossless decoder, a method that does not require the decoding of image to transmit visually lossless data is also proposed.

1.4 Memory structure

This thesis is presented as compendium of publications.It is structured as chapters that contains our publications. The last chapter provides some conclusions and future work.

The first contribution of this thesis is presented in Chapter 2 and has been published in:

"L. Jimenez-Rodriguez, F. Auli-Llinas, and M.W. Marcellin, FAST rate allocation for JPEG2000 video transmission over time-varying channels, IEEE Trans. Multimedia, vol. 15, Issue 1, pp. 15-26, Jan 2013."

The first part of the second contribution is described in Chapter 3.1 and corresponds to the paper:

"L. Jimenez-Rodriguez, F. Auli-Llinas, M.W. Marcellin, and J. Serra-Sagrista, "Visually Lossless JPEG 2000 Decoder," in Proc. IEEE Data Compression Conference,

Mar. 2013, pp. 161-170."

The second part of this contribution is issued in:

"L. Jimenez-Rodriguez, F. Auli-Llinas, and M.W. Marcellin, "Visually lossless strategies to decode and transmit JPEG2000 imagery", IEEE Signal Process. Lett., vol. 21, no. 1, pp. 35-38, Jan. 2014."

Chapter 2

JPEG2000 video transmission over time-varying channels

2.1 FAST rate allocation for JPEG2000 video transmission over time-varying channels

@ARTICLE{6202344,
author={Jimenez-Rodriguez, L. and Auli-Llinas, F. and Marcellin, M.W.},
journal={Multimedia, IEEE Transactions on},
title={FAST Rate Allocation for JPEG2000 Video Transmission Over Time-Varying Channels},
year={2013},
volume={15},
number={1},
pages={15-26},
doi={10.1109/TMM.2012.2199973},
ISSN={1520-9210},}

FAST Rate Allocation for JPEG2000 Video Transmission Over Time-Varying Channels

Leandro Jiménez-Rodríguez, Francesc Aulí-Llinàs, *Member, IEEE*, and Michael W. Marcellin, *Fellow, IEEE*

***Abstract*—This work introduces a rate allocation method for the transmission of pre-encoded JPEG2000 video over time-varying channels, which vary their capacity during video transmission due to network congestion, hardware failures, or router saturation. Such variations occur often in networks and are commonly unpredictable in practice. The optimization problem is posed for such networks and a rate allocation method is formulated to handle such variations. The main insight of the proposed method is to extend the complexity scalability features of the FAst rate allocation through STeepest descent (FAST) algorithm. Extensive experimental results suggest that the proposed transmission scheme achieves near-optimal performance while expending few computational resources.**

***Index Terms*—JPEG2000, rate allocation, time-varying channels, video transmission.**

I. Introduction

VIDEO transmission has been a prominent research topic for the last few decades. Its deployment in myriad applications, such as teleconferencing, video broadcasting, video-on-demand, and surveillance systems, manifests the consolidation of such technology in our everyday lives.

Three elements are key in the design of a video transmission scheme: the coding system, the network characteristics, and the requirements of the application. Two main families of *coding systems* are currently available for the coding and transmission of images and video: interframe and intraframe. H.264/AVC [1] is the most advanced interframe standard that exploits dependencies among frames to efficiently compress video. JPEG2000 [2] is the most advanced intraframe standard for the coding of images and video without considering frame dependencies. Both standards have been adopted in different scenarios, and both provide powerful tools for transmission of video over a network.

The *characteristics of the network* establish channel properties such as constant or variable channel capacity [3] and communication error rate [4], among others. The *application requirements* may introduce several demands on the transmission scheme: servers that deliver pre-encoded video [5] can use substantially different mechanisms than servers that encode and transmit video on-the-fly [6]; decoders with limited resources may raise challenging constraints [7]; and the use of smart proxies [8] or peer-to-peer (P2P) networks [9], [10] triggers new possibilities to efficiently transmit video.

Despite the large amalgam of scenarios created by the combination of these elements, all video transmission schemes pursue the same goal: to provide the best possible video quality to the end-user. When the distortion measure is mean squared error (MSE), one of two criteria is typically selected to optimize the quality of transmitted video [11]: 1) minimization of the average MSE (MMSE); and 2) provision of (pseudo-)constant quality, which is more commonly expressed as the minimization of the maximum MSE (MMAX) [12]. Of the two, subjective experiments suggest that MMAX may be more relevant perceptually [13]. Although this has been extensively discussed in the literature [14], [15], and even hybrid approaches have been proposed [16], video transmission schemes are generally focused on the optimization of one of these two criteria depending on the requirements of the application.

MMSE and MMAX are achieved by means of reducing/increasing the number of bytes transmitted for each frame, which is referred to as variable bitrate (VBR) video. Intuitively, VBR video delivers more bytes for those frames that are more difficult to compress (high spatial activity and/or motion) than for those frames that are easier to compress. The process that decides the number of bytes that are transmitted for each frame is called rate allocation, which is a key piece of video transmission schemes. Rate allocation methods must take into account the optimization criterion together with the coding system, the network characteristics, and the constraints imposed by the application.

This work considers a video-on-demand scenario that transmits pre-encoded JPEG2000 video to clients over a time-varying channel. To allow VBR video, the client has a limited buffer capacity to absorb irregularities in the sizes of compressed frames. We assume that the buffer size may vary from client to client, and that the channel capacity may vary over time in an unpredictable manner. We adopt JPEG2000 as the coding system since its fine grain quality scalability facilitates rate allocation of pre-encoded video. Furthermore, it is employed in many motion imagery applications, such as Digital Cinema distribution, television production, and surveillance.

Manuscript received December 12, 2011; revised March 15, 2012; accepted May 01, 2012. Date of publication May 17, 2012; date of current version December 12, 2012. This research was carried out when M. W. Marcellin was visiting professor and Marie Curie Fellow at the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona. This work was supported in part by the Universitat Autònoma de Barcelona, by the Spanish Government (MICINN), by the European Union, by FEDER, and by the Catalan Government, under Grants UAB-472-01-2/09, RYC-2010-05671, FP7-PEOPLE-2009-IIF-250420, TIN2009-14426-C02-01, and 2009-SGR-1224. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Feng Wu.

L. Jiménez-Rodríguez and F. Aulí-Llinàs are with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Bellaterra 08193, Spain (e-mail: ljimenez@deic.uab.es; fauli@deic.uab.es).

M. W. Marcellin is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721 USA (e-mail: marcellin@ece.arizona.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2012.2199973

The rate allocation algorithm introduced in this paper builds on our previous approach FAst rate allocation through STeepest descent (FAST) [17]. The main shortcoming of FAST is that, as originally formulated, it can *not* absorb variations in channel capacity during transmission. Such variations occur frequently in real-world scenarios that transmit data over the Internet or wide area networks due to congestion, irregularities in network conditions, etc.

An important feature required in time-varying channels is that the rate allocation method absorbs channel irregularities while guaranteeing that the transmission does not violate any existing buffer limits at the client. Important aspects of the optimization problem are that the variations in channel capacity are not known *a priori,* and that the server cannot interrupt the video transmission to compute new frame rates when channel conditions vary. The method introduced in this work extends two particular features of FAST to deal with such variations: scalability in terms of complexity, and the roughly linear relation between computational load and the number of frames. This permits the introduction of efficient strategies that can handle variations in channel capacity without penalizing performance. Furthermore, the proposed method preserves interesting features of the original FAST algorithm such as optimization for MMSE or MMAX, and low memory requirements.

The paper is organized as follows. Section II describes the fundamentals of video transmission schemes. Section III establishes the optimization problem that arises in time-varying channels and describes the proposed algorithm. Section IV assesses the performance of the proposed algorithm through extensive experimental results. Section V provides concluding remarks.

II. Overview of Video Transmission Schemes

The simplest scheme to transmit video is to use a constant bit rate (CBR) policy that transmits the same rate (bits/frame) for all frames of the video sequence. Although CBR schemes maintain constant client buffer occupancy throughout the whole transmission, the video quality is not optimized.

Using variable bit rate (VBR) policies can provide the opportunity to optimize video quality. Nonetheless, VBR schemes introduce constraints to the optimization problem that have to be addressed carefully. Specifically, let R^{total} be the total rate (bits/sequence) used to satisfy a client request for a sequence, and let N be the number of frames of the sequence. For now, we assume that the channel capacity is fixed at constant W bits per second and that the rendering pace is $\mathcal{F}$ frames per second (fps). Then, R^{total} is determined as $R^{total} = (W/\mathcal{F}) \cdot N$. Suppose that the codestream for the ith frame is scalable and can be truncated at points j corresponding to increasing bitrates r_{ij} (bits/frame) and decreasing distortions d_{ij}, with $1 \leq i \leq N$ and $1 \leq j \leq Q_i$, where Q_i is the number of truncation points available for frame i. When the optimization criterion is MMSE, the objective of the optimization problem is to find the truncation points $\mathbf{x} = \{x(1), x(2), \ldots, x(N)\}$ corresponding to bitrates $r_{ix(i)}$ and distortions $d_{ix(i)}$ that minimize the distortion, do not exceed the bit budget, and respect the client buffer size, i.e.,

$$\min_{\mathbf{x}} \sum_{i=1}^{N} d_{ix(i)} \tag{1}$$

such that

$$\sum_{i=1}^{N} r_{ix(i)} \leq R^{total} \tag{2}$$

and

$$\begin{aligned} B^{\min} &\leq \frac{R^{total}}{N} \cdot f - \sum_{i=1}^{f} r_{ix(i)} \\ &\leq B^{\max} - \frac{R^{total}}{N} \\ \forall\, f,\ 1 \leq f \leq N & \end{aligned} \tag{3}$$

where $B^{\min}$ and $B^{\max}$ denote the minimum and maximum capacity of the client buffer, respectively, with $B^{\min} < B^{\max}$. The middle expression of inequality (3), $B(f) = R^{total}/N \cdot f - \sum_{i=1}^{f} r_{ix(i)}$, represents the buffer occupancy at the instant just after frame f is rendered. As discussed above, the total available transmission rate for the sequence is R^{total}, and the channel capacity W is assumed constant. So the rate transmitted per frame rendering period can be expressed as $W/\mathcal{F} = R^{total}/N$. It is worth noting that the frame period is constant (for example, $1/30$ second, corresponding to $\mathcal{F} = 30$ frames/second) even though the time to transmit the data for each frame is variable due to VBR encoding. Thus, data can be seen as entering the buffer at the constant rate of R^{total}/N bits per frame period. The total rate received up to frame f is therefore $R^{total}/N \cdot f$. On the other hand, each time a frame is rendered, its data are removed from the buffer, emptying $r_{ix(i)}$ bits from the buffer for frame i. Thus, the total rate emptied from the buffer up to frame f is expressed as $\sum_{i=1}^{f} r_{ix(i)}$. The difference between the filling and emptying corresponds to the buffer occupancy, $B(f)$ as expressed in the middle term of inequality (3). Recalling again that $B(f)$ is the buffer occupancy *just after* frame f is removed from the buffer, the right hand expression of (3) can be understood. During the frame period that occurs after frame f is removed, and before frame $f+1$ is removed, R^{total}/N bits will be added to the buffer as described above. There must be room in the buffer to accommodate these data, so the buffer occupancy just after frame f is rendered must be no greater than $B^{\max} - R^{total}/N$.

In practice, a certain amount of buffering delay is required to partially fill the buffer prior to any frames being rendered. In order to avoid cluttering the notation by including this delay and the resulting initial data in the buffer, we take $t = 0$ be the instant just before the first frame is rendered. Furthermore, we take $B(f)$ to be the buffer occupancy *relative* to the amount of data initially buffered, say B^0. The amount of data actually in the buffer just after frame f is rendered is then $B^0 + B(f)$. In our experiments, we fill the buffer half way prior to rendering the first frame. Thus, for a buffer size of S, we set $B^0 = S/2$, $B^{\max} = S/2$, and $B^{\min} = -S/2$.

With respect to the MMAX criterion, the formulation of the optimization problem is the same except that the objective function (1) is replaced by

$$\min_{\mathbf{x}} \left(\max_{i=1}^{N} d_{ix(i)} \right). \tag{4}$$

It has been shown that both optimization problems [i.e., that of (1) (2) (3), and that of (4) (2) (3)] can be solved within the same optimization framework [14], so some methods proposed in the literature are able to address both MMSE and MMAX.

Many schemes for video transmission have been explored since the mid-1990s [18], [19]. Three main approaches have proven effective to tackle the optimization problem above: dynamic programming techniques [7], Lagrange relaxation methods [20], [21], and steepest descent algorithms [8], [22], [23]. Commonly, dynamic programming techniques construct a trellis structure that contains all solutions to the problem. The application of the Viterbi algorithm over the trellis reaches the optimal solution. The main disadvantage of this approach is that it requires high memory resources to build the trellis, and high computational load to search the trellis. Lagrange relaxation methods reduce computational requirements by relaxing the constraints of the optimization problem.

The use of steepest descent techniques leads to more efficient rate allocation methods. The steepest descent algorithm employed by our previous work FAST [17] selects a trivial valid solution to the problem (potentially poor), and then iteratively makes small changes to the solution following some heuristic. The heuristic for the steepest descent when the optimization criterion is MMSE is the Lagrange cost [24]. Generally speaking, the Lagrange cost measures the compression efficiency achieved at different truncation points of the compressed codestream. In the JPEG2000 framework [2], the Lagrange cost is embodied in the distortion-rate slope. If r_{ij} and d_{ij}, respectively, denote the rate and distortion at the jth truncation point for frame i, the distortion-rate slope at this point is defined as

$$S_{ij} = \frac{d_{i(j-1)} - d_{ij}}{r_{ij} - r_{i(j-1)}}. \tag{5}$$

Truncation points are represented as quality layers within the JPEG2000 codestream. The distortion-rate slope of each layer can be recorded within the codestream. If layer fragmentation is desired, distortion-rate slopes at intra-layer fragmentation points can be estimated using a linear form as described in [17]. Accordingly, more truncation points for frame i can be added. The use of S_{ij} allows FAST to exclude codestream segments with low distortion-rate slopes (less valuable segments in terms of rate-distortion performance), leaving room for those segments with higher distortion-rate slopes. If heuristic S_{ij} is replaced by d_{ij}, the objective of the algorithm is altered so that it seeks the solution that has the lowest maximum distortion (MMAX) [17].

III. JPEG2000 Video Transmission Over Time-varying Channels

A. Optimization Problem

We now address the optimization problem that arises in time-varying channels. The capacity of a TCP/IP communication link is commonly determined using the amount of data accepted by the receiving node divided by the round trip time, i.e., $W = RWIN/RTT$, where $RWIN$ is the receive window and RTT denotes the round-trip time [25, Ch. 3.7].

In general, this provides a reliable enough estimate of the channel capacity (or TCP/IP throughput), which can be used by applications such as the proposed rate allocation algorithm. Evidently, each implementation may use different low-level routines to determine the channel capacity, although most are based on the aforementioned principle. FAST-TVC then considers the capacity as a parameter given by the network layer. Our experience indicates that most low-level network routines provide estimates of the channel capacity that are reliable enough to be used by applications. Although it may depend on each implementation, in general these routines detect variations on the channel bandwidth in fractions of a second, so the impact on the convergence and performance of the proposed algorithm is negligible.

For simplicity, we assume piecewise constant capacity. To this end, let C be the number of transmission intervals, each with constant channel capacity, that occur during the transmission of a video sequence. Let $W_c, 1 \le c \le C$, be the channel capacity during transmission interval c in bits/second. The total rate available to satisfy the client request is then $R'^{total} = \sum_{c=1}^{C} R_c^{total}$, where R_c^{total} is the total rate in transmission interval c, i.e., $R_c^{total} = W_c \cdot (\mathcal{T}_{c+1} - \mathcal{T}_c)$, with $\mathcal{T}_c$ representing the instant in time at which the channel capacity changes to W_c. For simplicity, we assume that the channel capacity can only change the instant just after a frame is rendered. We then seek the frame truncation points $\mathbf{x} = \{x(1), x(2), \ldots, x(N)\}$ to achieve

$$\min_{\mathbf{x}} \sum_{i=1}^{N} d_{ix(i)} \tag{6}$$

subject to

$$\sum_{i=1}^{N} r_{ix(i)} \le R'^{total} \tag{7}$$

and

$$\begin{aligned} B^{\min} &\le B_c + \frac{W_c}{\mathcal{F}} \cdot (f - f_c + 1) - \sum_{i=f_c}^{f} r_{ix(i)} \\ &\le B^{\max} - \frac{W_c}{\mathcal{F}} \\ &\forall\, f,\ f_c \le f < f_{c+1} \text{ and } \forall\, c,\ 1 \le c \le C \end{aligned} \tag{8}$$

where f_c is the first frame rendered after $\mathcal{T}_c$ and B_c is the initial buffer occupancy for the cth transmission interval determined as

$$B_c = \sum_{k=1}^{c-1} W_k \cdot (\mathcal{T}_{k+1} - \mathcal{T}_k) - \sum_{i=1}^{f_c - 1} r_{ix(i)}. \tag{9}$$

It is worth emphasizing that time $\mathcal{T}_c$ falls just after frame $f_c - 1$ is rendered. As in (3), the middle expression of the inequality in (8) represents the buffer occupancy the instant just after frame f is rendered. As in the previous section, replacing the objective function (6) that seeks MMSE by that of (4), the optimization criterion becomes MMAX.

B. Proposed Approach

The method proposed below to tackle the optimization problem of (6)(7)(8) is named FAST for time-varying channels (FAST-TVC). The main idea behind FAST-TVC is to use a greedy approach that assumes that the channel capacity will remain fixed throughout the entire transmission of the video. This approach is reasonable, since in a real-time scenario channel capacity changes are not known *a priori*. When a change does occur, the rates of all non-transmitted frames are recomputed, taking into account the buffer occupancy at the time of the change, but assuming again that there will be no further capacity changes. Assuming infinite computational resources, this would mean simply executing, at each bandwidth change, an instance of FAST with appropriate parameter settings. The main difficulty of this approach is that, absent infinite computational resources, the time required to compute a new solution may be non-negligible and/or unpredictable.

To this end, let t_c denote the time—not known *a priori*—required by the rate allocation algorithm to reach a solution. When a variation on the channel occurs at $\mathcal{T}_c$, the server continues the transmission of video from $\mathcal{T}_c$ to $\mathcal{T}_c + t_c$ employing frame rates as computed at the beginning of the previous transmission interval $c-1$. This could violate the limits of the client buffer. In practice, if t_c is sufficiently small, the limits of the buffer are not trespassed except in rare occasions. In such cases, if t_c were known, the server could compute a CBR strategy for use during this period that would avoid buffer violations. This calculation could be performed in negligible time. Instead of using the previous solution, a CBR strategy might always be employed from $\mathcal{T}_c$ to $\mathcal{T}_c + t_c$, though the result achieved in both cases is similar since few frames are transmitted during this period. More important is the fact that t_c determines the range of frames that will be re-optimized in response to the bandwidth change, denoted by $[f'_c, N]$. If t_c were known, f'_c could be determined as follows: Let f^*_c be the smallest f such that

$$\sum_{k=1}^{c-1} W_k \cdot (\mathcal{T}_{k+1} - \mathcal{T}_k) + W_c \cdot t_c < \sum_{i=1}^{f} r_{ix(i)}. \quad (10)$$

Then

$$f'_c = f^*_c + 1. \quad (11)$$

The left side of inequality (10) is the total number of bits received at the client up to time $\mathcal{T}_c + t_c$. The right side is the total number of bits received at the client up to and including frame f. Therefore, frame f^*_c is the last frame that begins to be received prior to time $\mathcal{T}_c + t_c$, and so is the first frame to finish being received at the client after $\mathcal{T}_c + t_c$. Thus, f^*_c *cannot* be considered by the rate allocation algorithm because at the moment the algorithm finishes execution f^*_c is already partially delivered. The frame after f^*_c (i.e., f'_c) is the first considered by the algorithm since its transmission begins after $\mathcal{T}_c + t_c$. Fig. 1 shows an example that illustrates these quantities. In this figure, the arrows above the horizontal line indicate the moment at which the *final bit* of a frame is deposited into the buffer. The arrows below the line indicate the moment at which a frame is removed from the buffer to be rendered.

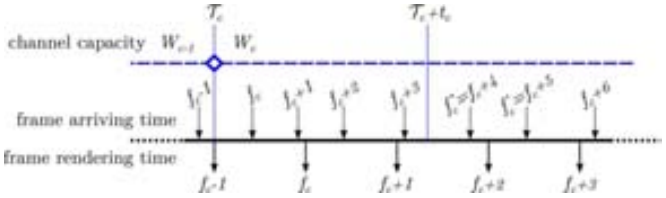


Fig. 1. Example time line of frame arrival and frame rendering times.

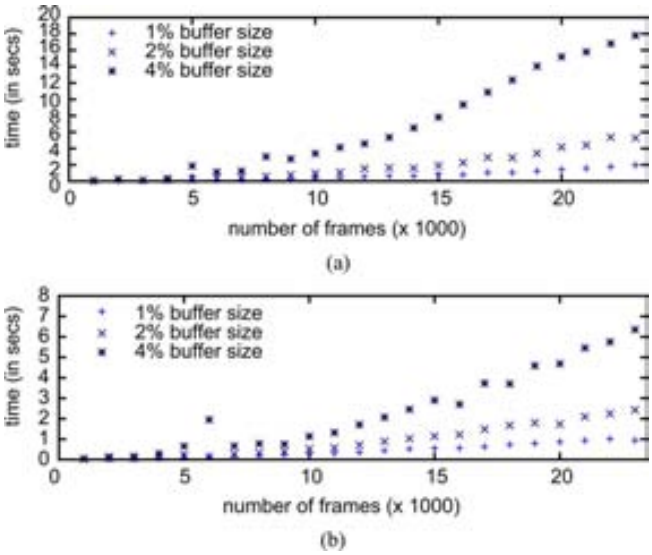


Fig. 2. Evaluation of the computational load versus the number of frames for different video sequences, buffer sizes, and optimization criteria. (a) MMSE optimization for "Batman." (b) MMAX optimization for "Willow."

Considering the quantities discussed above, the optimization problem of (6)–(8) can be re-formulated to take into account the time required by the rate allocation algorithm to reach a solution as

$$\min_{\mathbf{x}} \sum_{i=f'_c}^{N} d_{ix(i)} \quad (12)$$

subject to

$$\sum_{i=f'_c}^{N} r_{ix(i)} \leq R'^{total} - \sum_{i=1}^{f'_c-1} r_{ix(i)} \quad (13)$$

and

$$B^{\min} \leq B_c + \frac{W_c}{\mathcal{F}} \cdot (f - f_c + 1) - \sum_{i=f_c}^{f} r_{ix(i)} \leq B^{\max} - \frac{W_c}{\mathcal{F}} \; \forall \, f, \; f_c \leq f \leq N. \quad (14)$$

Inequality (13) represents the rate constraint for the frames in $[f'_c, N]$. The left side of this inequality is the number of bits to be transmitted for these frames. The right side is the remaining bit budget. Expression (14) is the buffer constraint, which is repeated from expression (8).

Key to tackling the optimization problem is then to determine t_c *before* the algorithm is actually executed. We propose three approaches to do so. The first approach uses a novel feature of the original FAST algorithm: scalability in terms of complexity. This type of scalability refers to the ability of the algorithm to provide successively improved solutions (in terms of the chosen

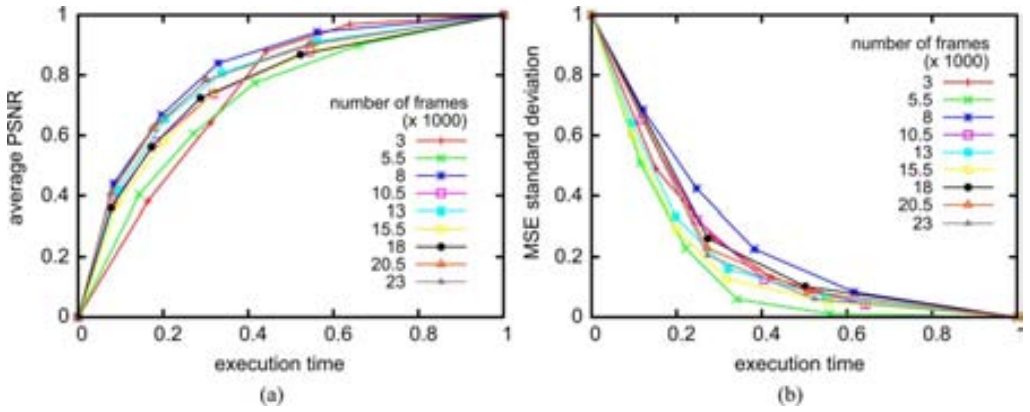


Fig. 3. Evaluation of the complexity scalability for the MMSE and MMAX criteria. (a) MMSE optimization for "Batman." (b) MMAX optimization for "Willow."

optimization criterion) as more time is spent in its execution. Complexity scalability allows the server to stop the rate allocation procedure after a predetermined period of time. Thus, t_c can be set by the server arbitrarily, depending on the system load, or by using any other indicator of the operating system. This approach is referred to as "constant t_c."

As shown in the next section, the complexity of determining frame rates varies significantly depending on the video sequence, the client buffer size, and the number of frames to be optimized. Hence, the "constant t_c" strategy may lead to significant suboptimalities. The second approach is to estimate the time that the rate allocation algorithm will need to finish the execution as a function of the number of frames to be optimized. The estimation is based on the roughly linear relation between the computational load required by FAST and the number of frames that are optimized. This can be seen from Fig. 2(a), which depicts the time spent by FAST when optimizing subsequences having different numbers of frames. Each subsequence is a clip from the "Batman" movie.[1] This figure reports computational load for three different buffer sizes, which are expressed as a percentage of R^{total}. The optimization criterion is MMSE. The time spent by the algorithm increases roughly linearly with the number of frames. This observation also holds for other video sequences and the MMAX optimization criteria [see Fig. 2(b)]. Let T denote the time spent by the algorithm when all N frames of the sequence are initially optimized at the beginning of transmission interval $c = 1$. The algorithm execution time can then be approximated as $t'_c = N' \cdot T/N$, with N' denoting the number of non-transmitted frames at time $\mathcal{T}_c$. This approach is referred to as "estimated t_c." We note that when employing this strategy, the algorithm is terminated at time t'_c even if it has not yet converged. Suboptimality due to this is typically negligible.

Although the "estimated t_c" strategy allows enough time for the algorithm to reach the optimal solution, it does not provide any mechanism to regulate the time spent by the rate allocation procedure. This may be critical when, for instance, the system load is high and resources have to be distributed among different processes. Furthermore, the value of t'_c may be too large, jeopardizing the client buffer as described above. Our third approach combines both the "constant t_c" and "estimated t_c" strategies to allow the server to regulate the time spent by the algorithm without sacrificing performance significantly. The main insight behind this approach comes from the observation that the performance metric improves more rapidly at the beginning of execution than when the algorithm is near convergence.

Fig. 3(a) depicts the MSE performance metric for solutions provided by the FAST procedure as a function of the time spent by the algorithm. This figure depicts results for a variety of subsequence lengths when the buffer size is 1% of R^{total}. Similar results hold for other buffer sizes and sequences. Both axes of the figure are normalized to allow comparison among different plots. Note that the average PSNR increases very rapidly at the beginning of execution, reaching near-optimal performance in half the time required by the algorithm to converge. Results are similar for the MMAX criterion, and are reported in Fig. 3(b) as the MSE standard deviation as a function of algorithm execution time.

These figures and our experience with other sequences indicate that 60% and 80% of the total time is enough to reach a solution very close to the optimal one, respectively for MMSE and MMAX. This can be exploited by the server to set $t_c = \min(t''_c, P \cdot t'_c)$, where the first term t''_c is the maximum time allowed by the server (which may depend on the system load or other indicators). The second term $P \cdot t'_c$, with $P = 0.6$ for MMSE and $P = 0.8$ for MMAX, is set to allow the algorithm to reach a near-optimal solution without spending computational resources unnecessarily. This third strategy is referred to as "weighted t_c."

C. Algorithm

The optimization procedure that seeks the solution to (12)(13)(14) is embodied in Algorithm 1. The algorithm assumes that there is no significant delay between the change in the channel capacity and its detection. As stated previously, when the server first receives a request from a client, it computes frame rates for all frames of the sequence. In the algorithm this is carried out by the procedure "computeFrameRates" (line 7), which receives the first and last frame numbers for the subsequence to be optimized, the channel capacity, buffer limits, current buffer occupancy, and maximum execution time. The procedure "computeFrameRates" is an implementation of the original FAST algorithm as described in [17], which returns solution $\mathbf{x}$ and execution time T. Frames are then transmitted according to the current solution until the end of the sequence

[1]See Section IV for a description of the video sequences and the experimental setup employed herein.

is reached or a change in the channel capacity occurs. When the channel capacity changes, the algorithm sets $t_{c^\star}$ in line 17 using one of the three strategies described above. While frame rates for the new channel capacity are being computed in line 20 using the maximum execution time $t_{c^\star}$, frames until $f'_{c^\star}$ are transmitted in the loop of lines 21–24. This process is carried out until all frames are transmitted.

Algorithm 1: FAST-TVC

1: receive client request

2: $c^\star \leftarrow 1$ /* *current transmission interval* */

3: $f'_{c^\star} \leftarrow 1$ /* *first frame transmitted in interval* $c^\star$ */

4: $B_{c^\star} \leftarrow 0$ /* *buffer occupancy at the beginning of interval* $c^\star$ */

5: $i^\star \leftarrow 1$ /* *currently transmitted frame* */

6: $W_{c^\star} \leftarrow \text{currentChannelCapacity}$

7: $\mathbf{x}, T \leftarrow \text{computeFrameRates}(f'_{c^\star}, N, W_{c^\star}, B^{\min}, B^{\max}, B_{c^\star}, \infty)$

8: **repeat**

9: **while** the channel capacity remains constant AND $i^\star \leq N$ **do**

10: transmit frame $i^\star$ using $r_{i^\star x(i^\star)}$

11: $i^\star \leftarrow i^\star + 1$

12: **end while**

13: **if** $i^\star \leq N$ **then**

14: $c^\star \leftarrow c^\star + 1$

15: $W_{c^\star} \leftarrow \text{currentChannelCapacity}$

16: $N' \leftarrow N - i^\star$

17: $t_{c^\star} \leftarrow \text{estimateAlgorithmTime}(T, N')$/* *using "constant* t_c*," "estimated* t_c*," or "weighted* t_c*"* */

18: $f'_{c^\star} \leftarrow$ according to (11) using $t_{c^\star}$

19: $B_{c^\star} \leftarrow$ according to (9)

20: $\mathbf{x} \leftarrow \text{computeFrameRates}(f'_{c^\star}, N, W_{c^\star}, B^{\min}, B^{\max}, B_{c^\star}, t_{c^\star})$

21: **while** (in parallel with line 20) $i^\star < f'_{c^\star}$ **do**

22: transmit frame $i^\star$ using $r_{i^\star x(i^\star)}$

23: $i^\star \leftarrow i^\star + 1$

24: **end while**

25: **end if**

26:**until** $i^\star > N$

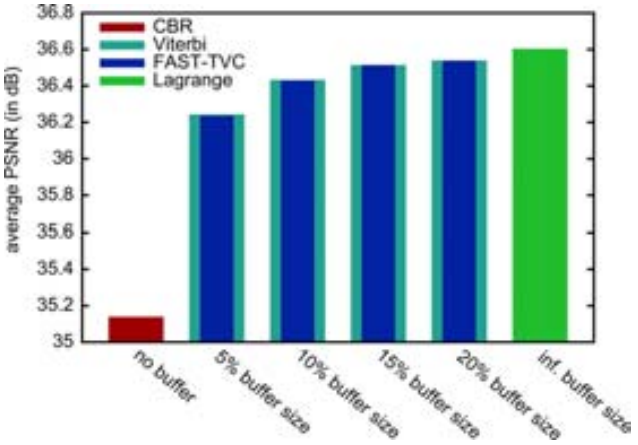


Fig. 4. Average PSNR achieved by FAST-TVC, CBR, Viterbi, and the Lagrange method when transmitting 2000 frames of the "StEM" sequence over a time-varying channel. The optimization criterion is MMSE.

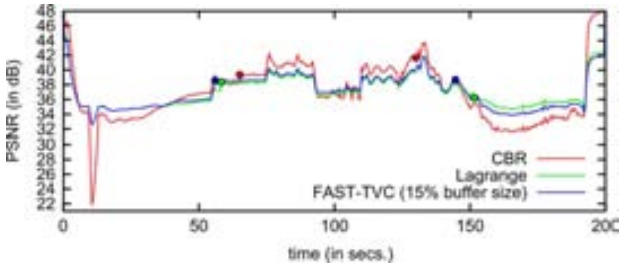


Fig. 5. Frame-by-frame PSNR achieved by FAST-TVC, CBR, and the Lagrange method for the "StEM" sequence. The optimization criterion is MMSE. There are three transmission intervals. For each method, the first frame transmitted after reoptimization (i.e., f'_2 and f'_3) is marked by a dot.

IV. Experimental Results

A. Coding Performance

FAST-TVC is assessed in terms of coding performance on five different sequences. Each frame of each sequence is compressed with 24 quality layers obtained by using the same 24 distortion-rate slope thresholds for each frame. Coding parameters are: 5 levels of 9/7 wavelet transform, with codeblocks of size 64×64.Table I describes the characteristics of the five video sequences employed in the experiments, as well as the transmitted range of frames, the rendering pace, and the transmission intervals. We first focus on the transmission of 2000 frames[2] of the "StEM" sequence over a channel that changes its capacity twice. The purpose of this first experiment is to appraise the coding performance of FAST-TVC compared to other strategies that obtain optimal performance, namely, the Viterbi algorithm [7], and the Lagrange method [17]. As described in Section II, the Viterbi algorithm is not practical since it requires enormous computational resources. Nonetheless, it provides optimal performance and provides a good reference to assess the performance of FAST-TVC. The Lagrange method is also impractical since it does not consider the restriction on the buffer size [expressions (3), (8), and (14)]. But, it yields the maximum performance that could be achieved if there were *no* buffer limits. The

[2]This experiment uses only 2000 frames to allow the use of our Viterbi implementation. The execution time and memory requirements of longer sequences exceed the resources of our servers.

TABLE I
CHARACTERISTICS OF THE VIDEO SEQUENCES EMPLOYED IN THE EXPERIMENTS, AND CONDITIONS OF THE CHANNEL IN EACH TRANSMISSION INTERVAL. FOR SIMPLICITY, ONLY THE LUMINANCE COMPONENT IS EMPLOYED (IMAGES ARE 8-BIT, GRAY SCALE)

sequence	frame size	subsequence rendering pace	transmission intervals $\mathcal{T}_c$ in seconds, W_c in Mbps	total
"StEM"	2048×857	[1150, 3149] 10 fps	$\mathcal{T}_1 = 0$, $\mathcal{T}_2 = 65$, $\mathcal{T}_3 = 130$ $W_1 = 3.52$, $W_2 = 4.8$, $W_3 = 4$	200 secs. 102.6 MB
"Batman"	590×325	[1, 24000] 10 fps	$\mathcal{T}_1 = 0$, $\mathcal{T}_2 = 300$, $\mathcal{T}_3 = 1150$, $\mathcal{T}_4 = 1400$, $\mathcal{T}_5 = 1950$, $\mathcal{T}_6 = 2150$ $W_1 = 3.04$, $W_2 = 2.8$, $W_3 = 2.48$, $W_4 = 2.64$, $W_5 = 2.72$, $W_6 = 3.12$	2400 secs. 836 MB
"Willow"	720×432	[290, 40290] 10 fps	$\mathcal{T}_1 = 0$, $\mathcal{T}_2 = 500$, $\mathcal{T}_3 = 1000$, $\mathcal{T}_4 = 2000$, $\mathcal{T}_5 = 2500$, $\mathcal{T}_6 = 3000$ $W_1 = 0.8$, $W_2 = 0.72$, $W_3 = 0.68$, $W_4 = 0.64$, $W_5 = 0.76$, $W_6 = 0.84$	4000 secs. 372.5 MB
"Giants of Africa"	720×432	[1, 53580] 10 fps	$\mathcal{T}_c = \{0, 100, 200, 400, 550, 650, 750, 1000, 1150, 1300, 1500, 1850, 1900, 2000, 2200, 2400, 2750, 2850, 3000, 3200, 3500, 3750, 3900, 4050, 4200, 4500, 4650, 4800, 4950, 5200\}$ $W_c = \{0.5, 0.51, 0.53, 0.54, 0.53, 0.55, 0.56, 0.58, 0.54, 0.53, 0.49, 0.51, 0.5, 0.49, 0.48, 0.55, 0.58, 0.56, 0.55, 0.54, 0.53, 0.49, 0.46, 0.48, 0.5, 0.51, 0.5, 0.49, 0.48, 0.5\}$	5357.9 secs. 221.02 MB
"Toy Story"	720×432	[1, 113168] 10 fps	$\mathcal{T}_c = \{0, 400, 500, 1000, 1500, 1700, 2200, 3000, 3300, 3700, 3900, 4500, 4800, 5200, 5700, 6000, 6200, 6500, 6800, 7300, 7700, 8200, 8500, 8850, 9200, 9450, 9700, 10100, 10750, 10950\}$ $W_c = \{0.63, 0.65, 0.69, 0.68, 0.7, 0.69, 0.66, 0.64, 0.63, 0.6, 0.61, 0.6, 0.59, 0.58, 0.6, 0.56, 0.59, 0.55, 0.54, 0.58, 0.6, 0.61, 0.64, 0.66, 0.69, 0.68, 0.64, 0.63, 0.6, 0.59\}$	11316.7 secs. 564 MB

TABLE II
CODING PERFORMANCE EVALUATION FOR FAST-TVC, CBR, AND THE LAGRANGE METHOD. THREE DIFFERENT STRATEGIES TO COMPUTE t_c ARE EMPLOYED BY FAST-TVC

			"StEM"		"Batman"		"Willow"		"Giants of Africa"		*average*
	buffer size:		5%	7%	1%	2%	4%	5%	5%	6%	*4.38%*
MMSE av. MSE	CBR		19.42		3.32		8.20		35.75		*16.67*
	FAST-TVC	"constant t_c"	16.60	15.40	3.19	3.17	6.98	6.88	31.91	31.74	*14.48*
		"weighted t_c"	15.55	14.94	3.10	3.07	6.80	6.60	27.42	27.35	*13.10*
		"estimated t_c"	15.26	14.89	3.09	3.07	6.78	6.60	27.42	27.33	*13.06*
	Lagrange		13.88		3.01		6.29		26.97		*12.54*
	buffer size:		15%	20%	5%	7%	12%	15%	16%	17%	*13.38%*
MMAX MSE st. dev.	CBR		21.92		1.50		7.61		33.27		*16.08*
	FAST-TVC	"constant t_c"	6.43	1.02	0.84	0.77	8.66	5.68	27.95	26.25	*9.7*
		"weighted t_c"	3.09	0.99	0.57	0.47	3.83	3.16	5.99	5.04	*2.89*
		"estimated t_c"	3.08	0.99	0.57	0.47	3.83	3.11	5.97	5.03	*2.88*
	Lagrange		0		0		0		0		*0*

performance of the CBR strategy is also reported for comparison purposes.

To provide an upper bound on the performance that can be obtained, in this first experiment, the execution time required by the algorithms is not considered (i.e., t_c is set to 0). The resulting performance cannot be obtained in practice without essentially infinite computational resources. Fig. 4 reports the average MSE of all frames of the sequence for the aforementioned methods when the optimization criterion is MMSE. Viterbi and FAST-TVC obtain virtually the same coding performance regardless of the client buffer size. As expected, the larger the buffer, the closer the performance of Viterbi and FAST-TVC is to that of the Lagrange method. Similar results hold for other video sequences and for the MMAX criterion. These experiments suggest that, under these circumstances, FAST-TVC achieves near-optimal performance.

Fig. 5 reports, for the same conditions as above and a buffer size of 15%, the PSNR achieved for each frame. The first frame transmitted in the second and third transmission intervals (i.e., f_2' and f_3') is marked with a dot in this figure. The quality of frames within each transmission interval can be seen to depend on its corresponding channel capacity. It is worth noting that, for the buffer size shown, frames transmitted with FAST-TVC have quality very similar to those transmitted with the Lagrange method. Contrarily, the quality of the simple CBR strategy often varies significantly from that of the Lagrange strategy.

As mentioned above, the first experiment reports results when algorithm execution time is ignored (i.e., $t_c = 0$). The aim of the next experiment is to assess the coding performance of FAST-TVC in a more realistic scenario. This test transmits 24 000, 40 290, and 53 580 frames, respectively, from the "Batman", "Willow", and "Giants of Africa" video sequences. The channel changes capacity 5, 5, and 29 times after the start of transmission, respectively, for "Batman", "Willow", and "Giants of Africa" (see Table I for more details). The three strategies described in Section III to determine t_c, namely "estimated t_c," "constant t_c", and "weighted t_c," are put into practice in this experiment. Additionally, performance for the CBR and Lagrange methods are also reported for comparison purposes.

The results of this experiment are reported in Table II for both the MMSE and MMAX criteria. The buffer sizes chosen for MMAX are generally larger than those for MMSE since MMAX commonly requires more buffer space to provide better pseudo-constant quality [17]. The corresponding values of T and t_c are reported in Table IV. For the "constant t_c" strategy,

TABLE III
CODING PERFORMANCE AND COMPUTATIONAL TIME EVALUATION FOR OPTIMIZING MMSE WITH THREE DIFFERENT BUFFER SIZES. RESULTS ARE REPORTED AS AVERAGE MSE AND SECONDS FOR "TOY STORY" TRANSMITTED OVER A CHANNEL THAT CHANGES CAPACITY 30 TIMES

buffer size	policy		T	t_c	$\sum t_c$	av. MSE
0.5%	CBR		0	0	*0*	40.8
	FAST-TVC	"constant t_c"	5.6	2.5	*72.5*	36.9
		"weighted t_c"	5.6	3.2, 3.2, 3.1, 3.0, 2.9, 2.8, 2.6, 2.5, 2.4, 2.3 2.1, 2.0, 1.9, 1.7, 1.6, 1.6, 1.5, 1.4, 1.2, 1.1 0.9, 0.8, 0.7, 0.6, 0.5, 0.5, 0.3, 0.1, 0.1	*48.6*	35.9
		"estimated t_c"	5.9	5.7, 5.7, 5.4, 5.4, 5.2, 4.9, 4.6, 4.3, 4.2, 4.1 3.7, 3.6, 3.3, 3.1, 2.9, 2.8, 2.6, 2.4, 2.1, 1.9 1.6, 1.5, 1.2, 1.1, 0.9, 0.8, 0.6, 0.3, 0.2	*86.1*	35.2
	Lagrange		0	0	*0*	31.0
0.8%	CBR		0	0	*0*	40.7
	FAST-TVC	"constant t_c"	10.1	2.5	*72.5*	35.0
		"weighted t_c"	10.2	5.9, 5.9, 5.6, 5.5, 5.4, 5.1, 4.7, 4.5, 4.3, 4.2 3.8, 3.7, 3.4, 3.2, 2.9, 2.8, 2.7, 2.5, 2.1, 1.9 1.7, 1.5, 1.2, 1.0, 0.9, 0.8, 0.6, 0.2, 0.1	*88.1*	34.2
		"estimated t_c"	10.2	9.9, 9.7, 9.3, 9.2, 9.1, 8.7, 7.9, 7.6, 7.2, 7.0 6.4, 6.1, 5.8, 5.3, 4.9, 4.7, 4.4, 4.1, 3.7, 3.3 2.8, 2.5, 2.0, 1.7, 1.5, 1.2, 0.9, 0.3, 0.2	*147.4*	33.8
	Lagrange		0	0	*0*	31.0
1%	CBR		0	0	*0*	40.8
	FAST-TVC	"constant t_c"	17.8	2.5	*72.5*	34.6
		"weighted t_c"	18.1	10.5, 10.4, 9.9, 9.8, 9.6, 9.1, 8.3, 7.9, 7.6, 7.4 6.8, 6.5, 6.0, 5.5, 5.1, 4.9, 4.6, 4.2, 3.8, 3.4 2.9, 2.5, 2.1, 1.8, 1.5, 1.3, 0.9, 0.3, 0.2	*154.8*	34.2
		"estimated t_c"	17.8	17.1, 16.9, 16.2, 16.0, 15.8, 14.9, 13.7, 13.1, 12.5, 12.1, 11.1, 10.6, 9.9, 9.1, 8.5, 8.2, 7.7, 7.2, 6.3 5.7, 4.8, 4.2, 3.5, 2.9, 2.6, 2.2, 1.6, 0.7, 0.4	*255.5*	33.1
	Lagrange		0	0	*0*	31.0

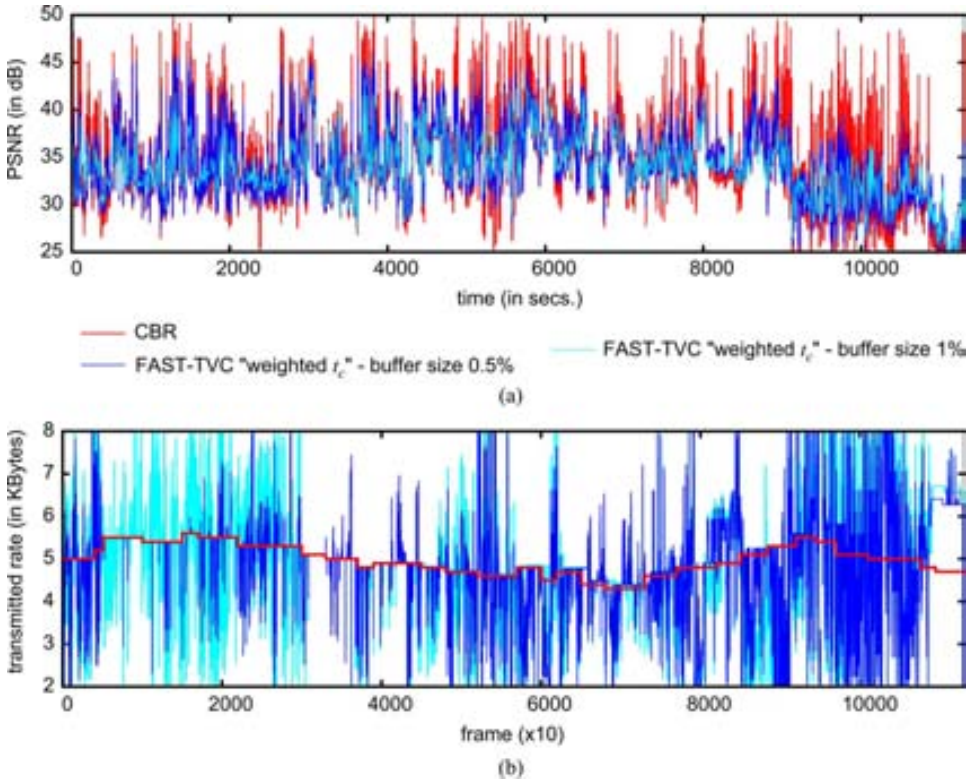


Fig. 6. (a) Frame-by-frame PSNR and (b) transmitted rate achieved by FAST-TVC and CBR method for the "Toy Story" sequence. The optimization criterion is MMSE.

t_c is chosen so that the total time spent by the algorithm is lower than that spent by the other two strategies. This choice of t_c illustrates the degradation on performance that the "constant t_c" strategy might produce when small values for t_c are used. The fixed part of the "weighted t_c" strategy is chosen to be larger than the variable part (i.e., $t_c'' > P \cdot t_c'$) to let this strategy

TABLE IV
COMPUTATIONAL TIME EVALUATION FOR FAST-TVC. THE FIRST COLUMN OF EACH CRITERIA REPORTS THE COMPUTATIONAL TIME (IN SECONDS) SPENT TO OPTIMIZE ALL FRAMES OF THE SEQUENCE (i.e., T). THE FOLLOWING COLUMNS REPORT t_c. THE LAST COLUMN REPORTS THE TOTAL TIME SPENT BY THE ALGORITHM TO RE-COMPUTE FRAME RATES IN RESPONSE TO CHANNEL CAPACITY VARIATIONS (i.e., $\sum_c t_c$)

		MMSE						*total*
"StEM" (buffer 5%)	"constant t_c"	2.089	0.075					*0.150*
	"weighted t_c"	2.088	0.881	0.406				*1.287*
	"estimated t_c"	2.220	1.562	0.722				*2.284*
"StEM" (buffer 7%)	"constant t_c"	3.502	0.075					*0.150*
	"weighted t_c"	3.478	1.599	0.719				*2.318*
	"estimated t_c"	3.664	2.665	1.185				*3.850*
"Batman" (buffer 1%)	"constant t_c"	2.542	0.250					*1.250*
	"weighted t_c"	2.527	1.330	0.794	0.639	0.283	0.160	*3.206*
	"estimated t_c"	2.545	2.232	1.334	1.072	0.474	0.270	*5.382*
"Batman" (buffer 2%)	"constant t_c"	4.759	0.250					*1.250*
	"weighted t_c"	4.773	2.516	1.506	1.213	0.529	0.308	*6.072*
	"estimated t_c"	4.753	4.175	2.499	2.013	0.878	0.511	*10.076*
"Willow" (buffer 4%)	"constant t_c"	4.691	0.500					*2.500*
	"weighted t_c"	4.711	2.053	1.528	1.022	0.729	0.324	*5.656*
	"estimated t_c"	4.721	3.430	2.552	1.704	1.217	0.539	*9.442*
"Willow" (buffer 5%)	"constant t_c"	8.062	0.500					*2.500*
	"weighted t_c"	8.096	3.522	2.618	1.729	1.288	0.597	*9.754*
	"estimated t_c"	8.088	5.872	4.355	2.882	2.110	0.996	*16.215*
"Giants of Africa" (buffer 5%)	"constant t_c"	127.835	1.750					*50.75*
	"weighted t_c"	127.862	74.949, 73.616, 70.884, 71.393, 70.011, 68.576, 64.879, 63.312, 61.365, 59.019, 54.122, 53.389, 51.763, 47.910, 44.402, 39.505, 37.993, 36.107, 33.166, 27.793, 24.870, 22.772, 20.524, 18.365, 11.299, 9.201, 5.088, 2.046					*1218.319*
	"estimated t_c"	127.863	124.916, 122.694, 118.141, 118.990, 116.687, 114.294, 108.132, 105.520, 102.275, 98.366, 90.203, 88.982, 86.272, 79.853, 74.006, 65.841, 63.322, 60.174, 55.277, 46.322, 41.450, 37.953, 34.208, 30.609, 18.831, 15.334, 8.480, 3.409					*2030.541*
"Giants of Africa" (buffer 6%)	"constant t_c"	130.647	1.750					*50.75*
	"weighted t_c"	130.664	76.414, 75.065, 72.274, 72.799, 71.385, 69.939, 66.153, 64.582, 62.582, 60.202, 55.292, 54.636, 53.036, 49.001, 45.319, 40.401, 38.794, 36.863, 33.906, 28.349, 25.363, 23.238, 20.931, 18.718, 11.501, 9.374, 6.487, 5.182, 2.084					*1249.87*
	"estimated t_c"	130.681	127.428, 125.179, 120.524, 121.400, 119.042, 116.629, 110.317, 107.697, 104.346, 100.372, 92.141, 91.058, 88.357, 81.654, 75.544, 67.156, 64.554, 61.292, 56.136, 47.109, 42.052, 38.478, 34.506, 30.887, 18.720, 15.380, 10.663, 8.517, 3.421					*2080.559*

		MMAX						*total*
"StEM" (buffer 15%)	"constant t_c"	3.507	0.050					*0.100*
	"weighted t_c"	3.533	2.094	0.681				*2.775*
	"estimated t_c"	3.512	2.602	0.846				*3.448*
"StEM" (buffer 20%)	"constant t_c"	8.854	0.050					*0.100*
	"weighted t_c"	8.858	4.022	1.218				*5.240*
	"estimated t_c"	8.820	6.704	2.014				*8.718*
"Batman" (buffer 5%)	"constant t_c"	22.087	0.500					*2.500*
	"weighted t_c"	21.928	15.683	9.653	7.931	3.577	1.998	*38.842*
	"estimated t_c"	22.045	19.708	12.131	9.966	4.495	2.510	*48.810*
"Batman" (buffer 7%)	"constant t_c"	15.781	0.500					*2.500*
	"weighted t_c"	15.650	11.123	6.766	5.495	2.486	1.396	*27.266*
	"estimated t_c"	15.778	14.018	8.527	6.925	3.133	1.758	*34.361*
"Willow" (buffer 12%)	"constant t_c"	13.176	1.000					*5.000*
	"weighted t_c"	13.115	7.956	5.645	3.174	2.168	1.079	*20.022*
	"estimated t_c"	13.143	9.967	7.071	3.976	2.714	1.346	*25.074*
"Willow" (buffer 15%)	"constant t_c"	13.693	1.000					*5.000*
	"weighted t_c"	13.677	8.272	5.918	3.540	2.566	1.196	*21.492*
	"estimated t_c"	13.685	10.347	7.401	4.430	3.219	1.506	*26.903*
"Giants of Africa" (buffer 16%)	"constant t_c"	9.529	2.800					*81.2*
	"weighted t_c"	9.521	7.423, 7.271, 7.040, 7.122, 6.989, 6.858, 6.505, 6.370, 6.251, 6.061, 5.663, 5.616, 5.424, 5.114, 4.746, 4.177, 4.059, 3.761, 3.397, 2.796, 2.523, 2.437, 2.243, 1.966, 1.240, 0.953, 0.622, 0.494, 0.173					*125.294*
	"estimated t_c"	9.529	9.287, 9.097, 8.807, 8.910, 8.744, 8.580, 8.138, 7.970, 7.820, 7.582, 7.085, 7.026, 6.785, 6.397, 5.938, 5.225, 5.078, 4.705, 4.250, 3.498, 3.157, 3.047, 2.807, 2.459, 1.552, 1.192, 0.778, 0.617, 0.214					*156.745*
"Giants of Africa" (buffer 17%)	"constant t_c"	10.089	2.800					*81.2*
	"weighted t_c"	10.089	7.870, 7.716, 7.461, 7.548, 7.399, 7.266, 6.913, 6.757, 6.621, 6.436, 6.003, 5.957, 5.759, 5.500, 5.050, 4.427, 4.321, 4.000, 3.628, 2.966, 2.675, 2.595, 2.374, 2.084, 1.323, 1.009, 0.658, 0.480, 0.048					*132.844*
	"estimated t_c"	10.092	9.840, 9.648, 9.329, 9.438, 9.252, 9.085, 8.644, 8.445, 8.279, 8.047, 7.506, 7.448, 7.201, 6.877, 6.315, 5.536, 5.403, 5.002, 4.537, 3.709, 3.345, 3.244, 2.968, 2.606, 1.654, 1.262, 0.823, 0.600, 0.061					*166.104*

achieve near-optimal performance. Evidently, when $t''_c < P \cdot t'_c$, the "weighted t_c" strategy becomes equivalent to the "constant t_c" strategy. Results for MMSE are reported as the average MSE achieved for all frames of the sequence, while results for

MMAX are reported via the MSE standard deviation of frames. Table II presents the results for both criteria. For completeness, results achieved for transmission of the "StEM" sequence are also given in this table.

The results reported in Table II suggest that the three strategies proposed to control the time spent by FAST-TVC achieve significantly better results than the CBR strategy. The "estimated t_c" strategy achieves the best results, and "weighted t_c" is only 2% worse than "estimated t_c," on average. Results indicate that the larger the buffer size, the lower the average MSE, or MSE standard deviation achieved for the MMSE and MMAX criteria, respectively. This suggests that the use of FAST-TVC (either "weighted t_c" or "estimated t_c") with large enough buffers would achieve virtually the same performance as that achieved by the Lagrange method. On the other hand, the "constant t_c" strategy leads to lower performance improvements. This is because t_c is not selected considering the characteristics of the video sequence, the number of frames to be optimized or the channel conditions, which may give too little time to the allocation algorithm to optimize the sequence.

The third test transmits the "Toy Story" video sequence over a channel that changes capacity 29 times. This test employs three buffer sizes and the MMSE criterion. The last column of Table III reports the achieved results, in terms of average MSE. These results correspond with previous experiments, suggesting that the "weighted t_c" strategy achieves virtually same performance as that of the "estimated t_c" strategy, while the larger the buffer size the closer the solution to the Lagrange method. Fig. 6(a) and (b) reports, respectively, the PSNR and the transmitted rate achieved by FAST-TVC "weighted t_c " and the CBR policy for the same conditions as before with buffer sizes 0.5% and 1%. The PSNR achieved by CBR is irregular, producing quick quality changes among consecutive frames. The use of a buffer and FAST-TVC obtains more regular PSNR. The larger the buffer size, the fewer abrupt quality changes. The Lagrange method (not depicted in the figure to avoid cluttering) achieves only a slightly more regular PSNR than FAST-TVC with buffer size 1%. The achievement of regular quality comes at the expense of more variable transmitted rate. Note in Fig. 6(b) that the strategy with the largest variations on the transmitted frame rate is FAST-TVC with buffer size 1%.

B. Computational Load

The proposed FAST-TVC algorithm is implemented in Java and executed on a Java Virtual Machine v1.6 using GNU/Linux v2.6. The server is an Intel Xeon E5520 CPU at 2.3 GHz. Time results are reported as CPU processing time, in seconds. Table IV reports the execution time spent by the three strategies of FAST-TVC, for transmission of the video sequences "StEM", "Batman", "Willow", and "Giants of Africa" under the same conditions as described above. Table III reports results for "Toy Story". The first column for each strategy reports the time spent when the client request is received and all frames of the sequence are optimized, i.e., T. The following columns report the execution time spent by the algorithm when a variation on the channel occurs (i.e., t_c). The last column reports the sum of all execution times excepting T. Recall that the percentage of time given to the "weighted t_c" strategy is 60% and 80% for the MMSE and MMAX strategies, respectively.

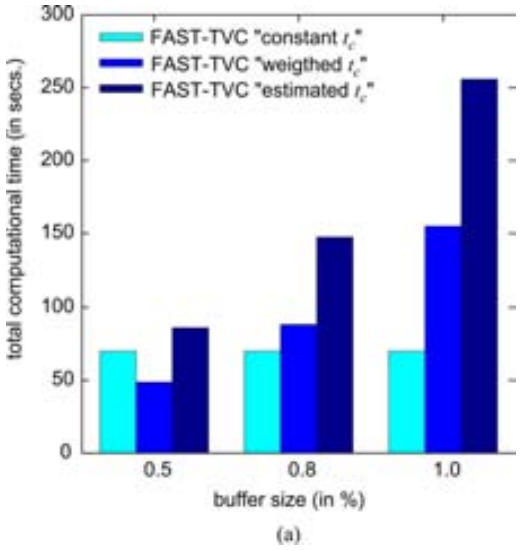


Fig. 7. Computational time spent by the three strategies of FAST-TVC in response to channel capacity variations (i.e., $\sum_c t_c$) when optimizing the "Toy Story" sequence for MMSE with three different buffer sizes.

Experimental results suggest that, even though the "weighted t_c" strategy spends 40% and 20% less time than the "estimated t_c" strategy, respectively, for MMSE and MMAX, its coding performance is almost unaffected compared to "estimated t_c." As stated previously, these savings on computational load are achieved due to the fast convergence of the rate allocation algorithm.

Fig. 7 depicts the computational time spent by the three strategies of FAST-TVC when transmitting "Toy Story" with the same buffer sizes and channel conditions as used before. Note that the larger the buffer, the more time required by the strategies "weighted t_c" and "estimated t_c". The "constant t_c" strategy spends the same computational time regardless of the buffer size. It is worth noting that, under these conditions, "constant t_c" spends more time on average than "weighted t_c" when the buffer size is 0.5% although the solution achieved by "weighted t_c" is better than that of "constant t_c" (see Table III). This is because "weighted t_c" spends a variable amount of time depending on the number of frames to be optimized. In particular, less computational time is used by "weighted t_c" for capacity variations that occur near the end of the sequence due to the smaller number of frames to be considered. This indicates that distribution of the computational time carried out by "weighted t_c" is adequately balanced considering the conditions of the channel and video sequence at the instant the channel variation occurs.

V. Conclusions

Rate allocation is of paramount importance in video transmission schemes to optimize video quality. Applications that transmit video over local area networks, Internet, or dedicated networks, may experience variations on channel conditions due

to network saturation, TCP congestion, or router failures. This work proposes a rate allocation algorithm for the transmission of JPEG2000 video named FAST-TVC. The proposed method is built on our previous FAST algorithm, extending and exploiting some of its features. The main insight behind FAST-TVC is to employ complexity scalability and the roughly linear relation between computational load and number of frames to re-compute frame rates once a variation on the channel capacity takes place.

Experimental results indicate that FAST-TVC achieves virtually the same coding performance as that of the optimal Viterbi algorithm (when the Viterbi algorithm is computationally feasible). When the server needs to control the resources dedicated to the rate allocation algorithm depending on system load or other indicators, FAST-TVC can use one of three proposed strategies. The first strategy is named "constant t_c" and provides a constant execution time to the algorithm. Although this strategy achieves a non-negligible gain in coding performance with respect to a constant-rate strategy, results vary significantly depending on the video sequence, buffer size, and channel conditions. The second strategy is named "estimated t_c" referring to its ability to estimate the total time that FAST-TVC requires to finish its execution. This allows FAST-TVC to achieve more consistent results, but does not supply any mechanism to reduce computational time when the server is busy. The "weighted t_c" strategy is a compromise between the previous two: it achieves virtually same results as "estimated t_c," and reduces computational load significantly. Experimental results evaluating the computational costs of FAST-TVC indicate that very few computational resources are expended. These characteristics makes FAST-TVC a suitable method for the transmission of pre-encoded JPEG2000 video in real-world applications.

REFERENCES

[1] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: Tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, Jan. 2004.

[2] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Norwell, MA: Kluwer, 2002.

[3] C.-Y. Hsu, A. Ortega, and A. R. Reibman, "Joint selection of source and channel rate for VBR video transmission under ATM policing constraints," *IEEE J. Select. Areas Commun.*, vol. 15, no. 6, pp. 1016–1028, Aug. 1997.

[4] C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE J. Select. Areas Commun.*, vol. 17, no. 5, pp. 756–773, May 1999.

[5] R. Rejaie, M. Handley, and D. Estrin, "Layered quality adaptation for internet video streaming," *IEEE J. Select. Areas Commun.*, vol. 18, no. 12, pp. 2530–2543, Dec. 2000.

[6] J. C. Dagher, A. Bilgin, and M. W. Marcellin, "Resource-constrained rate control for motion JPEG2000," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1522–1529, Dec. 2003.

[7] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximations," *IEEE Trans. Image Process.*, vol. 3, no. 1, pp. 26–40, Jan. 1994.

[8] Z. Miao and A. Ortega, "Scalable proxy caching of video under storage constraints," *IEEE J. Select. Areas Commun.*, vol. 20, no. 7, pp. 1315–1327, Sep. 2002.

[9] D. Jurca, J. Chakareski, J.-P. Wagner, and P. Frossard, "Enabling adaptive video streaming in P2P systems," *IEEE Commun. Mag.*, vol. 45, no. 6, pp. 108–114, Jun. 2007.

[10] E. Setton and B. Girod, *Peer-to-Peer Video Streaming*. New York: Springer, 2007.

[11] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, Nov. 1998.

[12] D. T. Hoang, "Fast and efficient algorithms for text and video compression," Ph.D. dissertation, Brown Univ., Providence, RI, 1997.

[13] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR video: Tradeoffs and potentials," *Proc. IEEE*, vol. 86, no. 5, pp. 952–973, May 1998.

[14] G. M. Schuster, G. Melnikov, and A. K. Katsaggelos, "A review of the minimum maximum criterion for optimal bit allocation among dependent quantizers," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 3–17, Mar. 1999.

[15] K.-L. Huang and H.-M. Hang, "Consistent picture quality control strategy for dependent video coding," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1004–1014, May 2009.

[16] S.-Y. Lee and A. Ortega, "Optimal rate control for video transmission over VBR channels based on a hybrid MMAX/MMSE criterion," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Aug. 2002, vol. 2, pp. 93–96.

[17] F. Auli-Llinas, A. Bilgin, and M. W. Marcellin, "FAST rate allocation through steepest descent for JPEG2000 video transmission," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1166–1173, Apr. 2011.

[18] A. R. Reibman and B. G. Haskell, "Constraints on variable bit-rate video for ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 4, pp. 361–372, Dec. 1992.

[19] C.-T. Chen and A. Wong, "A self-governing rate buffer control strategy for pseudoconstant bit rate video coding," *IEEE Trans. Image Process.*, vol. 2, no. 1, pp. 50–59, Jan. 1993.

[20] S.-W. Wu and A. Gersho, "Rate-constrained optimal block-adaptive coding for digital tape recording of HDTV," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 1, pp. 100–112, Mar. 1991.

[21] J. Choi and D. Park, "A stable feedback control of the buffer state using the controlled Lagrange multiplier method," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 546–558, Sep. 1994.

[22] Z. Miao and A. Ortega, "Optimal scheduling for streaming of scalable media," in *Proc. IEEE Asilomar Conf. Signals, Systems, and Computers*, Nov. 2000, vol. 2, pp. 1357–1362.

[23] Y. Sermadevi and S. S. Hemami, "Efficient bit allocation for dependent video coding," in *Proc. IEEE Data Compression Conf.*, Mar. 2004, pp. 232–241.

[24] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, pp. 399–417, 1963.

[25] J. F. Kurose and K. W. Ross, *Computer Networking. A Top-Down Approach*. Reading, MA: Addison-Wesley, 2008.

Leandro Jiménez-Rodríguez received the B.E. and M.S. degrees in computer engineering from the Universitat Autònoma de Barcelona, Bellaterra, Spain, in 2008 and 2010, respectively. He is currently pursuing the Ph.D. degree.

In 2009 he was awarded with a doctoral fellowship from the Universitat Autònoma de Barcelona that funds his current doctoral studies. His research interests include scalable video coding systems and video transmission.

Francesc Aulí-Llinàs (S'06–M'08) received the B.Sc. and B.E. degrees in computer management engineering and computer engineering from the Universitat Autònoma de Barcelona, Bellaterra, Spain, in 2000 and 2002, respectively, and for which he was granted with two extraordinary awards of Bachelor. In 2004 and 2006, respectively, he received the M.S. degree and the Ph.D. degree (with honors), both in computer science, from the Universitat Autònoma de Barcelona.

He is a Ramón y Cajal Fellow at the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona. Since 2002 he has been consecutively awarded with doctoral and postdoctoral fellowships in competitive calls. From 2007 to 2009 he carried out two research stages of one year each with the group of David Taubman, at the University of New South Wales (Australia), and with the group of Michael Marcellin, at the University of Arizona, Tucson. He is the main developer of BOI, a JPEG2000 Part 1 implementation that was awarded with a free software mention from the Catalan Government in April 2006. His research interests include a wide range of image coding topics, including highly scalable image and video coding systems, rate-distortion optimization, distortion estimation, and interactive transmission, among others.

Michael W. Marcellin (S'81–M'87–SM'93–F'02) received the B.S. degree (summa cum laude) in electrical engineering from San Diego State University, San Diego, CA, in 1983, where he was named the most outstanding student in the College of Engineering. He received the M.S. and Ph.D. degrees in electrical engineering from Texas A&M University, College Station, in 1985 and 1987, respectively.

Since 1988, he has been with the University of Arizona, Tucson, where he holds the title of Regents' Professor of Electrical and Computer Engineering, and of Optical Sciences. From 2001 to 2006, he was the Litton Industries John M. Leonis Professor of Engineering. He is currently the International Foundation for Telemetering Professor of Electrical and Computer Engineering at the University of Arizona. He is a major contributor to JPEG2000, the emerging second-generation standard for image compression. He is coauthor of the book *JPEG2000: Image Compression Fundamentals, Standards and Practice* (Norwell, MA: Kluwer, 2002).

Dr. Marcellin is a member of Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi. He is a 1992 recipient of the National Science Foundation Young Investigator Award, and a corecipient of the 1993 IEEE Signal Processing Society Senior (Best Paper) Award. He has received teaching awards from NTU (1990, 2001), IEEE/Eta Kappa Nu student sections (1997), and the University of Arizona College of Engineering (2000). In 2003, he was named the San Diego State University Distinguished Engineering Alumnus. He is the recipient of the 2006 University of Arizona Technology Innovation Award.

Chapter 3

Visually lossless

3.1 Visually Lossless JPEG 2000 Decoder

```
@INPROCEEDINGS{6543052,
author={Jimenez-Rodriguez, L. and Auli-Llinas, F. and Marcellin, M.W. and Serra-Sagrista, J.},
booktitle={Data Compression Conference (DCC), 2013},
title={Visually Lossless JPEG 2000 Decoder},
year={2013},
pages={161-170},
doi={10.1109/DCC.2013.25},
ISSN={1068-0314},}
```

2013 Data Compression Conference

Visually Lossless JPEG 2000 Decoder

Leandro Jiménez-Rodríguez[†], Francesc Aulí-Llinàs[†],
Michael W. Marcellin[‡], and Joan Serra-Sagristà[†]

[†] Department of Information and Communications Engineering
Universitat Autònoma de Barcelona, Barcelona, Spain
[‡] Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ, USA

Abstract

Visually lossless coding is a method through which an image is coded with numerical losses that are not noticeable by visual inspection. Contrary to numerically lossless coding, visually lossless coding can achieve high compression ratios. In general, visually lossless coding is approached from the point of view of the encoder, i.e., as a procedure devised to generate a compressed codestream from an original image. If an image has already been encoded to a very high fidelity (higher than visually lossless – perhaps even numerically lossless), it is not straightforward to create a "just" visually lossless version without fully re-encoding the image. However, for large repositories, re-encoding may not be a suitable option. A visually lossless *de*coder might be useful to decode, or to parse and transmit, only the data needed for visually lossless reconstruction. This work introduces a decoder for JPEG 2000 codestreams that identifies and decodes the minimum amount of information needed to produce a visually lossless image. The main insights behind the proposed method are to estimate the variance of the codeblocks before the decoding procedure, and to determine the visibility thresholds employing a well-known model from the literature. The main advantages are faster decoding and the possibility to transmit visually lossless images employing minimal bitrates.

I. Introduction

The last decades have experienced astounding growth in the use of images due to powerful capturing sensors such as those found in digital cameras, medicine instruments, or remote sensing devices. This has resulted in large repositories of images that have to be stored and transmitted, bringing new techniques and standards to compress such data sets efficiently. In general, images are encoded using a lossy or lossless coding scheme that permits the recovery of the original image with or without information loss, respectively. Lossless, or numerically lossless, methods commonly achieve moderate compression ratios, whereas lossy methods achieve higher compression ratios at the expense of image fidelity. In the last years, a new image compression modality employing the best of these two types of compression regimes has appeared. This modality is commonly called visually lossless coding due to its ability to compress an image making use of lossy coding techniques in such a way that the information loss is not noticeable by the human visual system (HVS). The main advantage of visually lossless coding methods is that they achieve compression ratios higher than those achieved by numerically lossless techniques, but look to a human observer as if they were compressed losslessly, i.e., without any loss in quality.

Typically, the first step in the implementation of a visually lossless coding scheme is to model the HVS. One approach to do so is to employ the contrast sensitivity function (CSF), which models the sensitivity of the human eye to contrast variations as a function of spatial frequency. The CSF has been measured with psychophysical

1068-0314/13 $26.00 © 2013 IEEE
DOI 10.1109/DCC.2013.25

experiments in order to find the just-noticeable points (i.e., the visibility thresholds) of a stimulus in a predefined contrast unit. The CSF varies depending on the age and the visual acuity of the subject, as well as on the viewing conditions and the stimuli used in the experiments [1], [2]. Typically, the stimuli employed to measure the sensitivity are generated with transforms that simulate the neurons of the primary visual cortex reception fields (V1) of the HVS. These reception fields are well described by the Gabor filter [3], which is ideal for space-frequency localization, albeit at high computational complexity. An alternative to the Gabor filter is to use the cortex transform proposed by Watson [4], which is invertible, easy to implement, and models V1 accurately and with adjustable parameters. Though being an appropriate tool to model the HVS, the cortex transform is not suitable to image compression because it increases the number of encoded coefficients [5]. Consequently, other transforms such as the discrete cosine transform (DCT) or the discrete wavelet transform (DWT) are more commonly employed in perceptual image compression.

The DWT is a decorrelation technique that has been utilized in vision models [6], [7] due to its well-posed properties for the HVS such as linearity, invertibility, and logarithmically spaced spatial frequencies divided in four orientations. Also, the DWT is one of the most popular transforms employed to perform image compression due to its decorrelation properties, which allow the attainment of high compression ratios. The JPEG 2000 standard [8], for example, utilizes the DWT as the first stage of the coding system. The use of the DWT in JPEG 2000 has permitted the deployment of techniques compatible with the standard that are aimed at the perceptual coding of images [9], [10]. These techniques yield improved visual quality, but are not able to ensure visually lossless performance. In one of the early steps in this direction, Watson et al. measured the visibility thresholds (VTs) for individual wavelet subbands using randomly generated *uniform* noise as a substitute for quantization error [11]. The resulting VTs can then be employed in wavelet-based coding schemes to code the coefficients in the wavelet subbands until the threshold for that subband is reached.

Unfortunately, the use of uniform noise to obtain the VTs of [11] results in non-visually lossless results when these VTs are employed in JPEG 2000 [12]. This is because JPEG 2000 employs a dead-zone uniform scalar quantizer which results in non-uniform quantization noise. Other approaches to obtain VTs (such as [13], [14]) achieve more accurate thresholds, though they still assume uniform noise and/or uniform quantization. A more suitable model of quantization noise for JPEG 2000 was proposed in [15]. Through that model, compressed images are produced that are indistinguishable from the original ones. Furthermore, the coding scheme proposed in [15] achieves superior compression ratios compared to previous visually lossless work done in the framework of JPEG 2000 [16].

The main trend in perceptual image coding has pursued increased accuracy of the HVS model to achieve higher compression ratios without affecting the perceptual quality of images. The main advantages of visually lossless methods are that the images look identical to the original ones, that the coding process can be faster because only the visually relevant information is coded, and that images can be transmitted employing less channel bandwidth. Nonetheless, there exist large repositories of images that have already been encoded using numerically lossless or (very high fidelity) lossy methods. Most perceptual coding methods in the literature are devised from the point of view of

the encoder. Unless the encoder has specifically envisioned it [17], there is generally no mechanisms to decode, or to parse and transmit, a visually lossless image from an already compressed codestream without performing a full decoding and re-encoding. To re-encode all images of large repositories may not be viable due to computational costs, so in some cases the benefits of visually lossless coding methods can not be exploited.

The purpose of this work is to introduce a visually lossless decoder that is able to identify and decode, or parse and transmit, only the information necessary to reconstruct a visually lossless image from a codestream encoded using a conventional JPEG 2000 encoder. The main insights behind the proposed method are to employ variance estimates that only require the decoding of codestream headers, and the use of the perceptual model of [15] to determine VTs.

The paper is organized as follows. Section II overviews the model employed to determine the VTs, and describes the proposed visually lossless JPEG 2000 decoder. Section III appraises the performance of the proposed method through experimental results that assess decoding rate and computational time reduction. The last section summarizes this work and draws lines of future research.

II. Visually lossless decoder for JPEG 2000

A. Determination of visibility thresholds

An important aspect behind the visually lossless method proposed in [15] is the model of quantization distortion employed, which captures with high accuracy the quantization error produced by a dead-zone uniform quantizer. Previous works assumed uniform error over the interval $(-\Delta/2, \Delta/2)$. Instead, [15] models the quantization error of high frequency wavelet subbands (i.e., subbands containing the High-vertical Low-horizontal frequencies (HL), or LH, or HH) by the probability density function (pdf)

$$f(d) = \begin{cases} g(d) + \dfrac{1 - \int_{-\Delta}^{\Delta} g(y)dy}{\Delta} & \text{if } 0 \leq |d| \leq \frac{\Delta}{2} \\ g(d) & \text{if } \frac{\Delta}{2} < |d| \leq \Delta \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $g(\cdot)$ denotes the pdf of coefficients d in a wavelet subband. In [15], $g(\cdot)$ is approximated as a Laplacian distribution with parameters $\mu = 0$ and variance σ^2. Δ is the step size of the quantizer. The first term in the two first lines of (1) indicate that the quantization error produced for coefficients within the deadzone interval (i.e., $(-\Delta, \Delta)$) is equal to the coefficients themselves, since they are reconstructed as zero. The second term in the first line of (1) arises from assuming that wavelet coefficients with $|d| > \Delta$ produce uniformly distributed errors. The resulting density function is depicted in Fig. 1. The low-frequency subband (i.e., LL) is modeled similarly.

This model of quantization distortion is employed to determine VTs for wavelet subbands. To do so, a stimulus image is generated by applying the inverse DWT to wavelet data that contain simulated quantization distortions. The (simulated) quantization distortion is generated in one wavelet subband employing the model of (1) for an assumed coefficient variance and quantization step size Δ. The inverse DWT then produces an

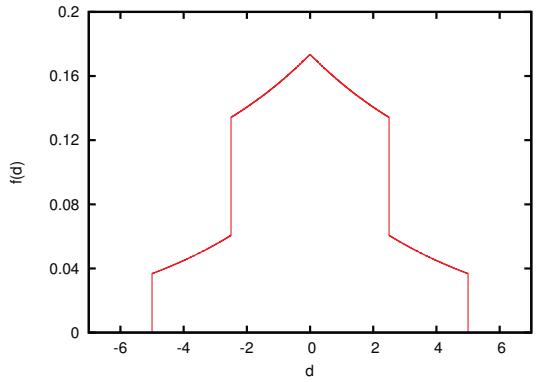


Fig. 1: Model of quantization distortion introduced in [15]. The pdf of wavelet coefficients $f(d)$ is modeled as a Laplacian distribution with parameters $\mu = 0$ and $\sigma^2 = 50$. The step size of the quantizer is $\Delta = 5$.

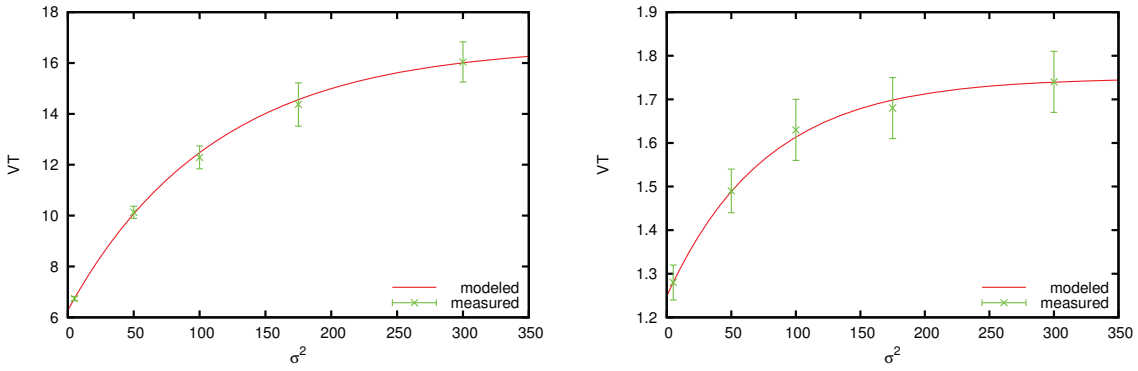
image with a distortion corresponding to quantization error for that subband, variance, and step size. To determine the VT for the subband and variance, a two-alternative forced choice method is used. In this method, the stimulus image and a mid-gray level image are displayed together and a human subject must decide which is the stimulus image. The experiment is iterated varying the step size Δ to find the largest Δ in which the stimulus image can not be distinguished from the mid-gray level image. Δ is determined after 32 iterations of the QUEST staircase procedure described in the Psychophysics Toolbox [18].

In [15], VTs were measured in this fashion for a small set of different values of variance in each subband. A piecewise linear function was then employed to model VTs for different values of variance. In this work, we have employed the same procedure to determine the VTs for a set of variance values in each wavelet subband. However, instead of employing a piecewise linear model for other values of variance, we employ the following logarithmic function

$$\mathrm{VT}(\sigma^2) = (\mathrm{VT}_{max} - \mathrm{VT}_{min}) \cdot \left(1 - \frac{B^{1-\frac{\sigma^2-\sigma^2_{min}}{\sigma^2_{max}}} - 1}{B-1} \right) + \mathrm{VT}_{min} \ , \qquad (2)$$

where VT_{max} is the VT determined experimentally for the maximum variance σ^2_{max} employed for the subband, and VT_{min} is the VT determined experimentally for the minimum variance σ^2_{min}. B determines the shape of the logarithmic function, and is selected to fit the VTs of the subband. Similar to [15], we have determined 5 VTs for each subband corresponding to $\sigma^2 = 5, 50, 100, 175,$ and 300. The parameter B has then been selected to fit the experimentally achieved thresholds.

Fig. 2 depicts the VTs determined for two different wavelet subbands together with the resulting models obtained via (2). Results for other subbands are similar. Table I reports the model parameters obtained for all subbands corresponding to 5 levels of irreversible 9/7 wavelet transform. Subbands HL and LH are reported together since the same VTs



(a) HH_1 (b) HL/LH_2

Fig. 2: Modeling of the VTs determined for wavelet subbands generated by the irreversible 9/7 CDF wavelet transform defined in JPEG 2000. The green points are the VTs determined for five different levels of variance, with the green bars showing ± 1 standard deviation. The red plot depicts the model of Equation (2).

TABLE I: Parameters employed in Equation (2) for each wavelet subband and decomposition level.

	HH			HL/HL		
Level	VT_{min}	VT_{max}	B	VT_{min}	VT_{max}	B
1	6.74	16.04	15	4.00	9.81	25
2	1.83	2.70	30	1.28	1.74	50
3	1.22	1.63	60	0.96	1.24	100
4	1.07	1.41	120	0.93	1.12	200
5	1.06	1.38	240	0.74	0.97	400

are achieved for both. As in [15], the LL subband is assigned a single VT regardless of the variance. The value employed here is 0.81.

B. Decoding procedure

The main stages of a typical JPEG 2000 encoder implementation are: data transformation, data coding, rate-distortion optimization, and codestream re-organization. The first stage transforms the image samples through a wavelet transform and quantizes wavelet coefficients employing a deadzone uniform scalar quantizer with step size Δ. The quantization indices are then grouped in small sets, called codeblocks, that are coded in the second stage by means of a fractional bitplane coding engine that carries out three coding passes per bitplane. One bitplane is defined as the collection of bits from all indices corresponding to the same position of their binary representation. JPEG 2000 and most modern image coding systems code wavelet data in a bitplane-by-bitplane fashion due to its inherent embedding and excellent coding performance. In JPEG 2000 each codeblock is coded independently, producing a quality progressive bitstream that can be truncated

at certain points. Rate-distortion optimization is commonly used to attain a target rate for the final codestream, or to construct quality layers. The main idea behind the optimization process of JPEG 2000 is to selectively include the bitstream segments of codeblocks in the final codestream employing a rate-distortion criterion. The final stage codes auxiliary information and organizes the final codestream using a progression order.

Commonly, the decoding procedure decodes the bitstream corresponding to a codeblock from the most significant bitplane of the codeblock to the least significant bitplane, or until the last coding pass included in the codestream for that codeblock is reached. Rather than decoding all available coding passes from the codeblock, the proposed method stops the decoding procedure upon reaching that bitplane which lies just below the VT determined for that subband. Specifically, let the bitplanes be numbered starting with 0 for the least significant. Then, decoding (starting with the most significant bitplane) is terminated after decoding the earliest bitplane P such that $\Delta 2^P \leq \mathrm{VT}(\sigma^2)$, with Δ denoting the quantization step size of the subband. In practice, the decoder computes P as

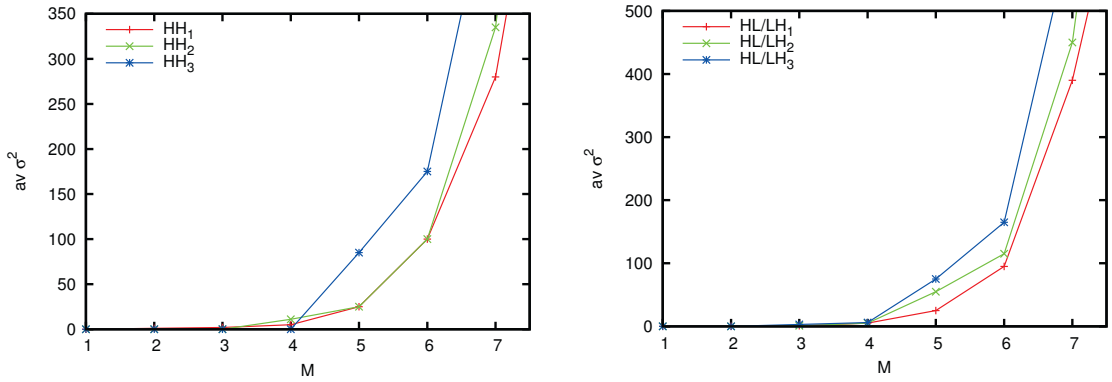
$$P = \left\lfloor log_2 \frac{\mathrm{VT}(\sigma^2)}{\Delta} \right\rfloor \ . \tag{3}$$

Evidently, if the codestream does not contain enough coding passes to reach bitplane P, the decoder stops the procedure at the last available coding pass and then visually lossless quality can not be guaranteed.

The main difficulty to apply the above procedure in practice is that the variances of the codeblocks are not available from the compressed codestream. Variances are not needed to decode the image and so to keep them in the codestream would unnecessarily increase its length. Variances could be estimated via decoding of the whole codestream, but this largely defeats the purpose of the present work. So an alternative estimate of these variances is required. It is important that the employed approach does not require the inclusion of additional information in the codestream since our goal is to obtain a decoder able to handle already encoded images. One piece of information relevant to the variance of a codeblock that can be obtained without decoding any bitplane data is the bitplane number of the most significant bitplane M, which is coded in the headers of the codestream. As seen below, M can be used to provide a reasonable estimate for the variance of a codeblock.

Fig. 3 reports the average variance for codeblocks found in the wavelet subbands for a large collection of wavelet-transformed images. Each point in the plot corresponds to the average variance of codeblocks in one wavelet subband that have the same value of M. The results indicate that the variance of codeblocks is strongly related to the wavelet subband and to the bitplane number of the most significant bitplane of the codeblock. Note, for instance, that the average variance of codeblocks with $M < 4$ is almost zero for all subbands, and then the variance increases exponentially as M grows. The proposed decoder employs the average variances reported in Fig. 3 as estimates.

In summary, the proposed decoder works as follows. First, the bitplane number of the most significant bitplane M for a given codeblock is extracted from the codestream headers. Second, the variance of the codeblock is estimated through a lookup table containing the average variances reported in Fig. 3. The wavelet subband of the codeblock and M are used as the indices of this lookup table. Third, the VT for the codeblock is



(a) HH subbands

(b) HL/LH subbands

Fig. 3: Evaluation of the variance of wavelet coefficients in codeblocks depending on their most significant bitplane M, for wavelet subbands produced by 5 levels of decomposition with the irreversible 9/7 CDF wavelet transform. Results are reported as the average variance of all codeblocks with same M, for the 24 images reported in Section III. Decomposition levels 4 and 5 are not depicted since the average variances obtained are higher than 500.

computed using the variance estimate and Equation (2). Fourth, the last bitplane P that the coding engine has to decode for that codeblock is computed via Equation (3). Fifth, the codeblock is decoded from bitplane M to bitplane P. This process is repeated for each codeblock to be decoded.

III. Experimental results

The experimental results carried out to assess the performance of the visually lossless JPEG 2000 decoder employ 24 images from different image corpora. All images are 8 bit, grayscale, with different sizes. Table II reports in the first two columns the images employed in this experiment as well as their sizes. We have not employed large images because the validation procedure described below requires the visualization of 3 versions of the same image simultaneously on the screen. The visually lossless decoder has been implemented in our JPEG 2000 codec BOI [19]. Coding parameters are: 5 levels of wavelet transform, codeblock size of 64×64, and a single quality layer codestream. The results for numerically lossless compression are achieved employing the reversible 5/3 CDF wavelet transform, whereas the remaining results are achieved employing the irreversible 9/7 CDF wavelet transform. The base quantization step size corresponding to bitplane 0 when the 9/7 filter-bank is used are chosen according to the L_2-norm of the synthesis basis vectors of the subband.

The first test validates that the images decoded by the proposed method are visually lossless. A three-alternative forced-choice (3AFC) procedure is used. In this procedure two original images and one decompressed image are displayed side by side on the screen with the position of the decoded image selected randomly. A subject is asked to choose the image which looks different. For each image, the test is repeated 5 times and the

image	size	NL	coder	decoder	[15]
barbara	512 × 512	4.66	2.00	-0.02	1.73
boat	512 × 512	4.01	1.90	-0.09	-
frog	621 × 498	6.25	3.74	+0.08	-
goldhill	512 × 512	4.84	2.51	-0.17	-
lena	512 × 512	4.32	1.87	-0.07	1.43
baboon	512 × 512	6.11	3.49	-0.07	2.70
mountain	640 × 480	6.70	3.85	-0.03	-
peppers	512 × 512	4.62	2.20	-0.23	1.61
zelda	512 × 512	3.99	1.56	-0.06	-
Woman	600 × 800	3.10	1.12	-0.04	-
Portrait	600 × 750	4.69	2.07	-0.15	-
Flowers	600 × 800	3.35	1.34	-0.13	-
Cafeteria	600 × 750	6.10	3.27	-0.02	-
Fishing goods	600 × 800	4.72	2.31	-0.18	-
Fruit Basket	600 × 750	4.48	1.98	-0.22	-
Japanese goods	600 × 800	5.07	2.64	-0.13	-
Tableware	600 × 750	4.50	1.73	-0.09	-
Field fire	600 × 800	4.53	2.22	-0.06	-
Bicycle	600 × 750	5.07	2.34	-0.10	-
Pier	600 × 800	4.79	2.37	-0.09	-
Orchid	600 × 750	3.53	1.08	-0.11	-
Threads	600 × 800	4.13	1.86	-0.08	-
Musicians	600 × 750	5.52	2.61	-0.11	-
Candle	600 × 750	6.15	3.23	-0.17	-
average	-	4.97	2.65	-0.10	-

TABLE II: Evaluation of the decoding rate achieved by the proposed decoder compared to a visually lossless encoder that uses same VTs. Results from [15] and results for numerically lossless (NL) encoding are also reported. All results are in bps.

subject has an unlimited time to examine the images. No viewing distance is enforced. A success ratio of $1/3$ would indicate that the images are visually lossless.

The 3AFC test is performed with a HP ZR2440w monitor that has an In-Plane Switching (IPS) panel, resolution of 1920×1200, static contrast ratio of 1:1000, brightness of 350 cd/m^2, and a dot pitch of 0.27mm. A total of 10 subjects participated in the validation test, making a total of 1200 validations. Images are first compressed with JPEG 2000 to a very high fidelity using the irreversible wavelet transform. Then, the decoder decompresses the visually relevant information from the codestream, discarding the remaining data. The mean frequency at which observers selected the correct image in this test was 0.343 with a standard deviation of 0.034. The achieved mean frequency is within one standard deviation of 1/3 and no outliers were detected, which suggest that the decoder produces visually lossless images.

Next, in the second test, the rate achieved by the proposed decoder is compared against the rate achieved by an encoder that uses the same method as that described for the decoder but using the real variances of the codeblocks. This test compares the accuracy of the variance estimates. The fourth column of Table II reports the rate achieved by the encoder, whereas the fifth column reports the difference between the decoder and encoder rates. Positive values in the fifth column indicate that the decoding rate is larger than the encoding rate. The achieved results suggest that the decoder is able to estimate variances with sufficient precision, resulting in a decoding rate only 0.10 bps less than that achieved by the encoder. The third and sixth columns of this table provide the results

image	NL	proposed	speed up
barbara	0.216	0.144	1.50
boat	0.212	0.139	1.53
frog	0.250	0.214	1.17
goldhill	0.216	0.161	1.34
lena	0.212	0.138	1.54
baboon	0.227	0.195	1.16
mountain	0.253	0.210	1.20
peppers	0.218	0.148	1.47
zelda	0.202	0.122	1.66
Woman	0.251	0.155	1.62
Portrait	0.278	0.191	1.46
Flowers	0.255	0.163	1.56
Cafeteria	0.312	0.237	1.32
Fishing goods	0.295	0.207	1.43
Fruit Basket	0.274	0.191	1.43
Japanese goods	0.296	0.217	1.36
Tableware	0.280	0.181	1.55
Field fire	0.287	0.207	1.38
Bicycle	0.295	0.209	1.41
Pier	0.291	0.212	1.37
Orchid	0.257	0.139	1.85
Threads	0.277	0.196	1.41
Musicians	0.301	0.216	1.39
Candle	0.314	0.174	1.81
average	0.261	0.182	1.46

TABLE III: Evaluation of the computational time employed by a numerically lossless (NL) decoder and the proposed visually lossless decoder. Results are reported in seconds.

achieved by a numerically lossless JPEG 2000, and those achieved in [15] for some of the images.

Compared to the numerically lossless method, the proposed decoder achieves significantly lower decoding rate. The codec introduced in [15] achieves higher compression ratios than those achieved by the proposed method. This is caused because [15] employs coding passes, instead of bitplanes, to decide when to stop the coding process, utilizes the real (sample) variance and distortion produced in each codeblock, and incorporates masking techniques to enhance the efficiency of the perceptual model. We note that, as originally formulated, these techniques can only be used in the encoder.

The third test is aimed to evaluate the computational time savings achieved when the proposed method is employed. Computational time results are obtained with an Intel Core2 Duo CPU at 3 GHz. BOI is implemented in Java and is executed on a JVM version 1.6. Table III reports the computational time spent by the bitplane coding procedure, which is also called tier-1 in JPEG 2000, when decoding the image numerically losslessly, and visually losslessly. On average, the proposed decoder is approximately 46% faster than numerically lossless decoding. These results suggest that the proposed JPEG 2000 visually lossless decoder is able to accelerate the decoding process without penalizing the visual quality of the decoded image.

IV. Conclusions

Visually lossless coding prevents quality losses while achieving high compression ratios. In general, visually lossless coding methods assume that the original image is

available. If the image is already coded, however, most methods are not able to identify the visually relevant information within the codestream without fully re-encoding the image. In this work we propose a method for the decoding of JPEG 2000 codestreams to produce visually lossless images. The main advantage of the proposed method is that it does not require re-encoding and so it can be employed to accelerate the decoding procedure or to transmit the image employing less bitrate than conventional methods. Future research is focused on the improvement of the perceptual model for the decoder by incorporating masking techniques as well as the inclusion of the proposed method in a JPIP-compliant server.

ACKNOWLEDGMENT

This work has been partially supported by the Universitat Autònoma de Barcelona, by the Spanish Government (MINECO), by the European Union, by FEDER, and by the Catalan Government, under Grants UAB-472-01-2/09, RYC-2010-05671, FP7-PEOPLE-2009-IIF-250420, TIN2009-14426-C02-01, TIN2012-38102-C03-03, and 2009-SGR-1224.

REFERENCES

[1] N. Graham, *Visual Pattern Analyzers*. New York: Oxford University Press, 1989.
[2] S. Daly, "Application of a noise-adaptive contrast sensitivity function to image data compression," *Optical Engineering*, vol. 29, no. 8, pp. 977–987, 1990.
[3] C. Taylor, Z. Pizlo, J. Allebach, and C. Bouman, "Image quality assessment with a gabor pyramid model of the human visual system," in *Electronic Imaging'97*. International Society for Optics and Photonics, 1997, pp. 58–69.
[4] A. Watson, "The cortex transform: rapid computation of simulated neural images," *Computer vision, Graphics, and Image Processing*, vol. 39, no. 3, pp. 311–327, 1987.
[5] D. Wu, D. Tan, M. Baird, J. DeCampo, C. White, and H. Wu, "Perceptually lossless medical image coding," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 335–344, 2006.
[6] M. Bolin and G. Meyer, "A perceptually based adaptive sampling algorithm," in *SIGGRAPH 99 Conference Proceedings*. ACM, 1998, pp. 299–309.
[7] M. Masry, S. Hemami, and Y. Sermadevi, "A scalable wavelet-based video distortion metric and applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 260–273, 2006.
[8] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image compression fundamentals, standards and practice*. Norwell, Massachusetts 02061 USA: Kluwer Academic Publishers, 2002.
[9] M. Nadenau and J. Reichel, "Opponent color, human vision and wavelets for image compression," in *Proceedings of the Seventh Color Imaging Conference*, 1999, pp. 237–242.
[10] W. Zeng, S. Daly, and S. Lei, "An overview of the visual optimization tools in JPEG2000," *Signal Processing: Image Communication*, vol. 17, no. 1, pp. 85–104, 2002.
[11] A. Watson, G. Yang, J. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise," *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1164–1175, 1997.
[12] Z. Liu, L. Karam, and A. Watson, "JPEG2000 encoding with perceptual distortion control," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1763–1778, 2006.
[13] M. Ramos and S. Hemami, "Suprathreshold wavelet coefficient quantization in complex stimuli: psychophysical evaluation and analysis," *Journal of the Optical Society of America A*, vol. 18, no. 10, pp. 2385–2397, 2001.
[14] D. Chandler and S. Hemami, "Dynamic contrast-based quantization for lossy wavelet image compression," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 397–410, 2005.
[15] H. Oh, A. Bilgin, and M. Marcellin, "Visually lossless encoding for JPEG2000," *IEEE Transactions on Image Processing*, to appear.
[16] D. Chandler and S. Hemami, "Effects of natural images on the detectability of simple and compound wavelet subband quantization distortions," *Journal of the Optical Society of America A*, vol. 20, no. 7, pp. 1164–1180, 2003.
[17] H. Oh, A. Bilgin, and M. W. Marcellin, "Visually lossless JPEG2000 at fractional resolutions," in *IEEE International Conference on Image Processing*, 2011, pp. 309–312.
[18] D. Brainard, "The psychophysics toolbox," *Spatial vision*, vol. 10, no. 4, pp. 433–436, 1997.
[19] F. Auli-Llinas. (2012) BOI software. [Online]. Available: http://www.deic.uab.es/~francesc

3.2 Visually Lossless Strategies to Decode and Transmit JPEG2000 Imagery

@ARTICLE{6661383,
author={Jimenez-Rodriguez, L. and Auli-Llinas, F. and Marcellin, M.W.},
journal={Signal Processing Letters, IEEE},
title={Visually Lossless Strategies to Decode and Transmit JPEG2000 Imagery},
year={2014},
volume={21},
number={1},
pages={35-38},
doi={10.1109/LSP.2013.2290317},
ISSN={1070-9908},}

Visually Lossless Strategies to Decode and Transmit JPEG2000 Imagery

Leandro Jiménez-Rodríguez, Francesc Aulí-Llinàs, *Member, IEEE*, and Michael W. Marcellin, *Fellow, IEEE*

***Abstract*—Visually lossless coding allows image codecs to achieve high compression ratios while producing images without visually noticeable distortion. In general, visually lossless coding is approached from the point of view of the encoder, so most methods are not applicable to already compressed codestreams. This paper presents two algorithms focused on the visually lossless decoding and transmission of JPEG2000 codestreams. The proposed strategies can be employed by a decoder, or a JPIP server, to reduce the decoding or transmission rate without penalizing the visual quality of the resulting images.**

***Index Terms*—Human visual system, JPEG2000, visibility thresholds, visually lossless coding.**

I. Introduction

VISUALLY lossless coding refers to the ability of an image coding system to identify and encapsulate the information of an image that is visually relevant to a human observer. Often, this is achieved by determining visibility thresholds (VTs) for the human visual system (HVS) that are introduced into the coding system to preserve the visually relevant information [1]. In the context of transform coding, the VT for a particular transform coefficient is the maximum absolute error between the original and the coded coefficient that results in just imperceptible distortion in the image.

The use of visually lossless coding has several advantages. First, images coded in this regime look to a human observer as if they were compressed losslessly. Second, visually lossless compression achieves higher compression ratios than numerically lossless compression [2]. Third, combined with transmission protocols, visually lossless coding enhances the interactive image transmission by reducing response times [3].

Early attempts toward visually lossless coding employed the Gabor filter and the cortex transform. Currently, the discrete wavelet transform (DWT) is more commonly employed due to its suitability for both perceptual models and image coding schemes. In one of the first applications of the DWT to perceptual coding, Watson *et al.* measured the VTs for individual wavelet subbands based on the HVS contrast sensitivity function using randomly generated *uniform* noise as a substitute for quantization error [1]. These VTs were then employed to code the coefficients of each subband until the threshold for that subband was reached. The thresholds from [1] were introduced in the framework of JPEG2000 in [4], but the resulting images were not strictly visually lossless. This stems from the fact that JPEG2000 employs a deadzone quantizer, which introduces non-uniform quantization noise. Other approaches to obtain VTs such as [5] achieve more accurate thresholds, though they still assume uniform quantization, rather than deadzone quantization. A more suitable model for the quantization noise caused by the quantizer of JPEG2000 was proposed in [6]. When that model is applied to JPEG2000, the resulting compressed images are indistinguishable from the original ones at superior compression ratios.

Despite numerous studies on visually lossless codecs, the focus of most work has been on the encoder side. To the best of our knowledge, there are no methods to decode, or to parse and transmit, a visually lossless image from an already compressed (very high fidelity, or even numerically lossless) codestream. Since most methods are devised from the point of view of the encoder, an obvious approach would be to perform a full decoding and re-encoding. In situations where it is desirable to maintain the original (super-visually-lossless) quality, the re-encoded codestream could include side information to allow subsequent parsing of a visually lossless version. In a layered system such as JPEG2000, the re-encoded codestream could be constructed so that decoding or transmitting the first n layers would guarantee a visually lossless image. Nevertheless, there may exist large repositories of images encoded using numerically lossless or very high fidelity lossy methods. In such repositories, re-encoding may not be viable due to high computational costs. Thus, visually lossless decoding or parsing is of great interest.

Motivated by the discussion above, this work introduces strategies to decode or transmit the information necessary to reconstruct a visually lossless image from a codestream previously encoded using a *conventional* JPEG2000 encoder. Clearly, this is not possible unless the original codestream contains sufficient information to produce a visually lossless image in the first place. The goal pursued here is to provide visually lossless quality while decoding or transmitting the smallest subset possible from the original codestream. The proposed strategies employ the perceptual model of [6] to produce techniques that can be employed in a JPEG2000 decoder or in a JPIP server.

Manuscript received July 30, 2013; revised October 01, 2013; accepted October 29, 2013. Date of publication November 11, 2013; date of current version November 13, 2013. This work was supported by the Spanish Government (MINECO), by FEDER, and by the Catalan Government under Grants UAB-472-01-2/09, RYC-2010-05671, TIN2012-38102-C03-03, and 2009-SGR-1224 . The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yiannis Andreopoulos.

L. Jiménez-Rodríguez and F. Aulí-Llinàs are with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Barcelona, Spain (e-mail: ljimenez@deic.uab.cat; fauli@deic.uab.cat).

M. W. Marcellin is with the Department of Electrical and Computer Engineering, University of Arizona, Tucson AZ 85721 USA (e-mail: marcellin@ece.arizona.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2013.2290317

Section II of this paper overviews the model of [6] and describes the proposed strategies. Section III assesses the performance of the proposed methods through experimental results, while the last section concludes with some remarks.

II. Proposed Strategies

A. Visually Lossless Encoding

The model of distortion produced by the JPEG2000 deadzone quantizer [6] is employed to determine VTs for wavelet subbands. To do so, a stimulus image is generated by applying the inverse DWT to wavelet data that contain simulated quantization distortions for an assumed coefficient variance σ^2 and quantization step size Δ. The inverse DWT then produces an image with a distortion corresponding to quantization error for that subband, variance, and step size. To determine the VT for the assumed subband and variance, a two-alternative forced choice method is used. In this method, the stimulus and a mid-gray level image are displayed together and a human subject decides which is the stimulus. The experiment is iterated varying Δ to find the largest Δ for which the stimulus is not distinguished from the mid-gray level image, which is then the VT for that subband and variance, denoted as $\mathrm{VT}(\sigma^2)$.

In a JPEG2000 encoder, each subband of the DWT is quantized using an initial step size Δ_i. In this work, the initial step size for a given subband is set equal to the square root of the energy gain factor [4, Ch. 4.3.2] for that subband, although other choices are allowed by the standard. After quantization, the wavelet subbands are divided into small sets of coefficients called codeblocks. Each codeblock is coded employing three coding passes per bitplane called significance propagation (SPP), magnitude refinement (MRP), and cleanup (CP) [4]. A bitplane is defined as the collection of bits from all quantized coefficients corresponding to the same position of their binary representation. In the *en*coder of [6], the above perceptual model is applied in each codeblock as follows. First, $\mathrm{VT}(\sigma^2_{\mathcal{B}})$ is computed employing the variance of the coefficients within codeblock $\mathcal{B}$. At the end of each coding pass, the maximum absolute error produced by the partially transmitted coefficients is computed as $D = \max_{w \in \mathcal{B}}(|w - \hat{w}|)$, with w and $\hat{w}$ denoting the original and the reconstructed coefficient, respectively. When $D \leq \mathrm{VT}(\sigma^2_{\mathcal{B}})$, the encoding procedure is stopped.

B. Application to the Decoder

In a JPEG2000 decoder, the bitstream corresponding to a codeblock is decoded from the most significant bitplane of the codeblock to the least significant bitplane, until the last coding pass included in the bitstream for that codeblock is reached. The first difficulty that arises when attempting to apply the perceptual model in the decoder is that the variance for the codeblock is not available since the image is already encoded. So an estimate for $\sigma^2_{\mathcal{B}}$ is needed. One piece of information relevant to the variance of a codeblock is the bitplane number of the most significant bitplane of the codeblock, denoted as M, which is coded in the headers of the codestream. Empirical evidence indicates that variance estimates can be obtained via M. Fig. 1 depicts the average variance of codeblocks found in three different wavelet subbands. Results for other subbands are similar. Each point in the plots corresponds to the average variance of codeblocks in one wavelet subband that have the same value of M. The results indicate that the variance of codeblocks is strongly related to the wavelet subband and to M. Note, for instance, that the average variance of codeblocks with $M < 4$ is almost zero for all subbands, and then increases exponentially as M grows. The proposed strategy employs these average variances as estimates, denoted as $\hat{\sigma}^2_{\mathcal{B}}$.

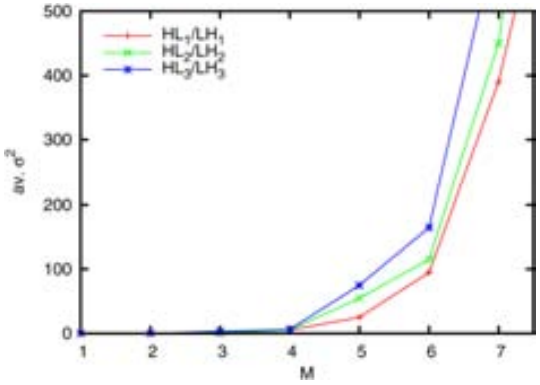


Fig. 1. Average variance of codeblocks having the same M in different subbands. Results are obtained for the images of Section III using the irreversible 9/7 DWT. Similar results are obtained for other subbands.

Another difficulty that arises is that D cannot be computed at the decoder because the original image is not available. The proposed strategy upper bounds the maximum absolute error at the end of a coding pass in bitplane P by noting that the effective (embedded) quantization step size of a coefficient, after bit P of its magnitude representation has been decoded, is $\Delta_i 2^P$. This fact, together with the knowledge of whether any coefficient from the codeblock is in the deadzone of the effective quantizer, can be used to upper bound the maximum absolute error as

$$D' = \begin{cases} \left.\begin{array}{ll} \Delta_i 2^P & \text{if pass} = \text{CP} \\ \Delta_i 2^{P+1} & \text{otherwise} \end{array}\right\} \text{if } \exists\ \hat{w} = 0 \\ \left.\begin{array}{ll} \Delta_i 2^P & \text{if pass = SPP} \\ \Delta_i 2^{P-1} & \text{otherwise} \end{array}\right\} \text{otherwise} \end{cases}. \quad (1)$$

Masking effects can also help to reduce the (de)coding rate without sacrificing visual quality. We adopt the strategy described in [6], in which the VT for a codeblock is multiplied by a masking factor α, $\alpha > 1$ when self- and/or texture-masking are present. Since the masking factor is computed from quantized coefficients, its implementation in the decoder presents no problems.

In summary, the proposed strategy for the decoder is as follows. First, the bitplane number of the most significant bitplane M for codeblock $\mathcal{B}$ is extracted from the codestream headers. Second, the variance of the codeblock $\hat{\sigma}^2_{\mathcal{B}}$ is estimated through a lookup table containing the average variances computed for a large corpus of images. Third, the VT for the codeblock is computed using the estimated variance $\hat{\sigma}^2_{\mathcal{B}}$. Fourth, the decoding process begins and, at the end of each coding pass, the maximum error D' and the masking factor α are computed.[1]. Decoding for codeblock B is stopped when $D' \leq \alpha \mathrm{VT}(\hat{\sigma}^2_{\mathcal{B}})$. Evidently, if the codestream does not contain enough coding passes to achieve $D' \leq \alpha \mathrm{VT}(\hat{\sigma}^2_{\mathcal{B}})$, the decoder stops the procedure after decoding

[1]A slight increment in coding performance can be achieved by re-estimating the codeblock variance at the end of each coding pass using partially reconstructed coefficients.

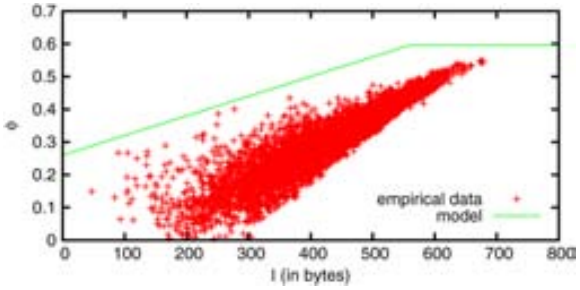


Fig. 2. Percentage of codeblock bitstream needed to reach the VT. Results are for the images of Section III when using the irreversible 9/7 DWT and 32×32 codeblocks.

TABLE I
PARAMETERS FOR THE UPPER BOUND TO ϕ AS A FUNCTION OF M

	$s = s_1 \cdot M + s_2$		$\phi_{\min} = n_1 \cdot M + n_2$		$\phi_{max} = m_1 \cdot M + m_2$	
	s_1	s_2	n_1	n_2	m_1	m_2
HH_1	-0.000105	0.00102	90	260	0.045	0.08
HL_1/LH_1	-0.00014	0.0012	80	260	0.055	0
HH_2	-0.000172	0.00155	90	260	0.072	0
HL_2/LH_2	-0.000191	0.00172	80	260	0.067	0
HH_3	-0.000155	0.00155	90	260	0.048	0
HL_3/LH_3	-0.00012	0.0012	80	260	0.06	0
HH_4	-0.000165	0.0018	90	260	0.048	0
HL_4/LH_4	-0.00013	0.00135	80	260	0.06	0
HH_5	-0.000165	0.0018	90	260	0.048	0
HL_5/LH_5	-0.00013	0.0014	80	260	0.06	0

the last available coding pass and then visually lossless quality cannot be guaranteed.

C. Application to JPIP Servers

The application of the visually lossless decoding procedure discussed above to a JPIP server is complicated by the fact that partial decoding of the file is required. It is preferable that the server not be required to decode any bitplane data, so that neither D' nor α can be computed. The only useful information about the codeblock that is then available is M, the number of coding passes, and the length of the bitstream generated for the codeblock, denoted as l.

Experiments indicate that M and l are good indicators of the amount of data that have to be transmitted to produce a visually lossless image. This can be seen as follows. Fig. 2 depicts the percentage of a codeblock bitstream required to reach its VT. The horizontal axis of the figure is l, whereas the vertical axis is the percentage of l, denoted as ϕ, that is required to reach the VT. Each point in the scatter plot corresponds to one codeblock in the HH_1 subband having $M = 4$. When l is small, ϕ is also small. As l increases, ϕ increases, until reaching a point at which ϕ does not grow more. Similar behavior holds for other subbands and Ms.

Results corresponding to Fig. 2 are upper bounded for each wavelet subband (and each value of M) by the function

$$\phi' = \begin{cases} s \cdot l + \phi_{\min} & \text{if } l < l_{\max} \\ \phi_{\max} & \text{otherwise} \end{cases}. \qquad (2)$$

The parameters s, $\phi_{\min}$, and $\phi_{\max}$ employed in the upper bound (as functions of M) are reported in Table I. The solid line in Fig. 2 depicts the upper bound of (2) for the corresponding subband and value of M. This upper bound to the actual value of ϕ was computed over a wide corpus of images, being (2) an overly conservative estimate to assure visually lossless.

The results of Fig. 2 were generated using initial step sizes as discussed in Section II-A and by including all coding passes of each codeblock bitstream. Since all images are assumed to have been previously encoded by "non-aware" JPEG2000 encoders, different initial step sizes may have been employed, and codeblocks may have some missing coding passes (due to rate allocation procedures, etc.). In the case of missing coding passes (only), the resulting difference in l is approximated by noting that missing passes correspond to the least significant bitplanes, which are nearly incompressible. Thus, the length of such coding passes is well approximated by one bit per coefficient per bitplane. In the case of different initial step sizes, the resulting difference in l can be approximated by $\log_2$ of the ratio between the true and the assumed step size, in units of bits per coefficient. The true step size can be read from the codestream headers.

In summary, the proposed strategy for the JPIP server is as follows. First, M and l are extracted (or in the case of l, estimated as needed) from the codestream headers. Second, the percentage of each codeblock bitstream that needs to be transmitted to achieve a visually lossless image is computed via (2). Third, the server transmits the corresponding portions of the codeblock bitstreams to the client. Fourth, the client decodes data until reaching the end of each codeblock bitstream segment. The decoder must be aware that the end of a bitstream segment may not coincide with the end of a coding pass, so it must stop when all bytes are consumed (see [7]).

III. EXPERIMENTAL RESULTS

Experimental results are reported in Table II (all images are 8 bit, grayscale). The JPEG2000 coding parameters employed are: 5 levels of DWT, and codeblocks of size 32×32. The reversible 5/3 DWT is employed for numerically lossless results, otherwise the irreversible 9/7 DWT is used. A three-alternative forced-choice (3AFC) procedure is used to validate the results, using the same procedures and viewing conditions as those in [6]. The 3AFC test is performed with a HP ZR2440w monitor that has an IPS panel, contrast ratio of 1:1000, brightness of 350 cd/m^2, and a dot pitch of 0.27 mm. A total of 12 subjects participated in the validation test. When the images are visually lossless, the probability of correct response for the 3AFC test should be 1/3. The 95% confidence intervals for the mean frequency at which observers selected the correct image in this test are reported in the first row of the table. When the appropriate confidence interval contains 1/3, the images are visually lossless for these viewing conditions.

Table II includes compression results (in bps) for the strategy of Section II-B (labeled "decoder") and for the strategy of Section II-C (labeled "server"). Also included for comparison are results for numerically lossless encoding, and for the encoder based procedure of [6] (labeled "encoder"). The 3AFC results achieved by the "encoder," "decoder," and "server" strategies suggest that each produces visually lossless images. The rates achieved by the decoder are always only slightly larger than those of the encoder. These small differences are due to the use of estimates for the variance and maximum absolute distortion in each codeblock. On the other hand, due

TABLE II
RESULTS ACHIEVED BY THE PROPOSED STRATEGIES. IMAGES WITH $*$ ARE THOSE USED IN THE VALIDATION TEST

	validation test				.346 ± .05			.350 ± .05			.488 ± .15			.431 ± .09		
	lossless	encoder [6]			decoder			server			server -40%			server -2BP		
image (size)	bps	bps	dB	ssim	bps	dB	ssim	bps	dB	ssim	bps	dB	ssim	bps	dB	ssim
barbara (512 × 512)	4.79	1.69	39.68	.9988	1.76	40.25	.9990	3.09	48.05	.9998	1.91	41.34	.9992	2.08	43.61	.9952
boats (512 × 512)	4.42	1.48	41.11	.9991	1.52	41.40	.9991	2.76	48.09	.9998	1.71	42.09	.9993	1.74	43.68	.9960
frog* (621 × 498)	6.28	3.17	38.42	.9965	3.53	40.38	.9978	4.70	47.86	.9996	2.87	37.52	.9957	3.96	44.51	.9787
goldhill* (512 × 512)	4.85	1.92	40.66	.9988	2.01	41.21	.9990	3.33	48.38	.9998	2.05	41.11	.9990	2.25	43.30	.9951
horse* (512 × 512)	5.26	2.24	39.73	.9992	2.32	40.16	.9993	3.67	48.19	.9999	2.26	39.58	.9992	2.75	43.98	.9953
lena* (512 × 512)	4.33	1.42	41.62	.9990	1.46	41.83	.9991	2.66	47.92	.9998	1.65	42.48	.9992	1.56	43.15	.9967
baboon* (512 × 512)	6.12	2.78	37.51	.9968	2.93	37.99	.9971	4.58	48.29	.9997	2.80	37.65	.9969	3.71	44.04	.9845
mountain* (640 × 480)	6.71	2.77	33.99	.9980	2.92	34.41	.9982	4.93	46.30	.9999	3.02	34.90	.9984	4.41	44.68	.9921
onthepad (512 × 512)	6.52	3.03	36.51	.9986	3.12	36.65	.9986	4.95	48.27	.9999	3.03	37.20	.9988	4.18	44.52	.9939
peppers* (512 × 512)	4.63	1.62	40.34	.9989	1.66	40.60	.9991	3.04	48.05	.9998	1.88	41.95	.9994	1.94	42.98	.9975
thecook* (512 × 512)	5.49	2.59	39.65	.9991	2.63	39.78	.9992	4.01	48.60	.9999	2.46	39.52	.9991	3.07	43.78	.9958
zelda* (512 × 512)	4.01	1.16	42.45	.9994	1.18	42.53	.9989	2.32	48.00	.9997	1.44	43.25	.9991	1.20	43.14	.9968
man (1024 × 1024)	4.84	1.85	40.46	.9992	1.94	40.90	.9992	3.26	48.21	.9999	2.02	41.28	.9993	2.18	43.13	.9968
woman* (600 × 800)	3.12	0.86	44.89	.9993	0.90	45.21	.9995	1.32	48.46	.9998	0.84	43.82	.9993	0.88	46.21	.9964
portrait (2048 × 2560)	4.41	1.57	41.25	.9975	1.64	41.67	.9993	2.73	47.97	.9998	1.68	41.23	.9993	1.78	43.71	.9963
flowers* (600 × 800)	3.36	1.04	44.00	.9993	1.10	44.56	.9994	1.61	49.02	.9998	1.01	43.27	.9992	1.16	46.20	.9953
cafeteria* (600 × 750)	6.11	2.42	35.14	.9993	2.54	35.45	.9977	4.31	46.63	.9998	2.65	36.50	.9982	3.73	44.64	.9903
fishing* (600 × 800)	4.73	1.83	40.67	.9991	1.90	41.04	.9994	2.92	47.15	.9998	1.81	40.32	.9993	2.18	44.11	.9962
fruit* (600 × 750)	4.49	1.68	41.31	.9991	1.74	41.66	.9994	2.76	48.12	.9997	1.71	41.15	.9993	1.95	44.38	.9964
japanese* (600 × 800)	5.07	2.18	40.03	.9952	2.26	40.37	.9991	3.35	47.00	.9998	2.07	39.23	.9989	2.68	44.36	.9946
tableware (600 × 750)	4.51	1.33	39.32	.9993	1.38	39.49	.9992	2.69	47.39	.9999	1.66	41.30	.9994	1.81	44.02	.9960
fieldfire* (600 × 800)	4.55	1.75	41.89	.9986	1.86	42.23	.9956	2.87	46.98	.9985	1.77	40.87	.9940	1.90	43.21	.9757
bicycle (2048 × 2560)	4.40	1.48	40.41	.9996	1.54	40.87	.9994	2.70	48.00	.9999	1.67	41.77	.9995	1.80	43.91	.9969
pier* (600 × 800)	4.80	1.88	39.29	.9993	1.97	39.88	.9988	3.09	47.56	.9998	1.91	38.81	.9985	2.53	45.19	.9922
orchid* (600 × 750)	3.55	0.82	43.38	.9997	0.86	43.65	.9997	1.71	48.13	.9999	1.07	44.31	.9997	0.91	45.01	.9980
threads* (600 × 800)	4.14	1.48	41.69	.9993	1.54	42.13	.9993	2.34	47.88	.9998	1.46	41.05	.9992	1.74	45.13	.9956
musicians (600 × 750)	5.53	2.19	37.52	.9982	2.24	37.68	.9982	3.78	46.92	.9998	2.33	39.05	.9987	2.99	43.64	.9937
silver (600 × 800)	3.67	1.19	42.75	.9993	1.25	43.40	.9994	1.80	48.24	.9998	1.13	41.86	.9991	1.38	46.06	.9947
candle* (600 × 750)	6.16	2.52	35.66	.9973	2.59	35.79	.9973	4.38	46.71	.9998	2.69	37.08	.9980	3.78	44.39	.9897
average	*4.86*	*1.86*	*40.05*	.9986	*1.94*	*40.45*	*.9988*	*3.16*	*47.81*	*.9998*	*1.95*	*40.40*	*.9987*	*2.35*	*44.23*	*.9935*

to the conservative upper bounds employed for ϕ in the server strategy, its rates are larger than those of the encoder strategy, though still substantially lower than for numerically lossless. Thus, it is of interest to consider less conservative strategies.

As mentioned previously, the upper bounds employed above were computed from a very large corpus of imagery containing images of different types. As the upper bounds apply to every image in this corpus, the proposed system is very robust to images with different statistical properties. A less conservative strategy that might lead to lower encoding rates for certain image types would be to compute different upper bounds for different classes of imagery. We do not pursue this strategy here due to space constraints, as well as our preference for a universal scheme that does not rely on prior knowledge of image types. Rather, the final two columns in the table represent alternate strategies, which decrease the file size significantly, but do not guarantee visually lossless quality. In particular, the strategy labeled "server -40%" is the same strategy as "server" but reduces ϕ' by 40%, resulting in an average rate similar to that achieved by the "decoder" strategy. Observers found that most images are visually lossless (and all have very high quality) for this strategy, so it may be good enough when strictly visually lossless is not required. The strategy labeled "server -2BP" omits the coding passes from the two least significant bitplanes of codestreams produced by the "server" strategy, which also produces slightly visible distortion in some images. For completeness, the PSNR and SSIM achieved for each image is also reported in Table II. Visually lossless images are achieved from 30 to 45 dB, all with SSIM values higher than 0.99.

IV. CONCLUSIONS

In general, visually lossless coding methods are done from the perspective of the encoder, and assume that the original image is available. If the image is already coded, most methods cannot identify the visually relevant information within the codestream without fully re-encoding the image. We propose strategies for the decoding and transmission of JPEG2000 codestreams that produce visually lossless images. The proposed strategies can be employed in a decoder, transcoder, or JPIP server to reduce the decoding or transmission rate without penalizing the visual quality of the images.

REFERENCES

[1] A. Watson, G. Yang, J. Solomon, and J. Villasenor, "Visibility of wavelet quantization noise," *IEEE Trans. Image Process.*, vol. 6, no. 8, pp. 1164–1175, Aug. 1997.

[2] Z. Liu, L. Karam, and A. Watson, "JPEG2000 encoding with perceptual distortion control," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1763–1778, Jul. 2006.

[3] H. Oh, A. Bilgin, and M. W. Marcellin, "Visually lossless JPEG2000 at fractional resolutions," in *IEEE Int. Conf. Image Processing*, Sep. 2011, pp. 309–312.

[4] D. S. Taubman and M. W. Marcellin, "JPEG2000 Image compression fundamentals, standards and practice," in . Norwell, MA, USA: Kluwer, 2002.

[5] D. Chandler and S. Hemami, "Dynamic contrast-based quantization for lossy wavelet image compression," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 397–410, Apr. 2005.

[6] H. Oh, A. Bilgin, and M. W. Marcellin, "Visually lossless encoding for JPEG2000," *IEEE Trans. Image Process.*, vol. 22, no. 3, pp. 189–201, Jan. 2013.

[7] F. Auli-Llinas, J. Bartrina-Rapesta, and J. Serra-Sagrista, "Enhanced JPEG2000 quality scalability through block-wise layer truncation," *EURASIP J. Adv. Signal Process.*, vol. 2010, pp. 1–11, May 2010, article ID 803542.

Chapter 4

Conclusions

4.1 Summary

Compression is commonly employed to transmit images and video over the Internet. In most Internet connections the channel capacity is not constant due congestion or infrastructure. In video-on-demand scenarios this may become a problem, leading to pauses while playing a movie. This problem has been considered in this thesis in transmission schemes that use intraframe coding.

FAst rate allocation through STeepest descent (FAST) is an algorithm which provides near optimal transmission performance assigning a specific rate for every frame of a JPEG2000 compressed video sequence. This thesis proposes and adaptation of the FAST algorithm that considers that the channel capacity may vary at any time of the transmission. Results obtained with the proposed method suggest that the modified algorithm provides near optimal performance without causing under-/over-flow to the clients buffer.

The most common image coding schemes are lossless and lossy compression. Lossless compression has no quality losses but its compression ratios aren't high. On the other hand, lossy compression ratios are high but the expense of losing image quality. In some scenarios higher compression ratios than those achieved with lossless compression are required without allowing losses in the image quality. In these scenarios, another coding scheme could be used: visually lossless. Visually lossless encodes only

the visually relevant data, obtaining high compression ratios without any perceptible quality loss. When the images have not been encoded in visually lossless regimes, they may have to be re-encoded to obtain a visually lossless compression. This may be a problem for large image repositories.

This thesis proposes a method that allows to decode or transmit JPEG2000 compressed images in visually lossless regime. The proposed method is based on a well-known visually lossless scheme, but adapting it to the circumstances of the decoder. For transmitting visually losslessly JPEG2000 images, this thesis also proposes a model that allows to discard almost all the non-visually relevant data of the codestream. This model does not need to decode the image. Results obtained for both the decoder and the transmission scheme suggest that the images decoded and transmitted using these methods are visually lossless while reducing significantly the rate decoded/transmitted..

4.2 Future work

The research presented in this thesis has used JPEG2000 gray scale images. One line of future work that has been started at the time of writing this text is the implementation of the visually lossless methods for color images.

A second line of future work is to implement a visually lossy compression method. The main idea behind such a coding regime is to use visual metrics to code the image instead of using numerically-based metrics such as the mean square error. The insight acquired with the visually lossless schemes proposed in this thesis may be a good basis to start from.

Appendix A

Acronyms

3AFC three-Alternative Forced-Choice

CBR Constant Bit Rate

CSF Contrast Sensitivity Function

CP Cleanup Pass

DCT Discrete Cosine Transform

DWT Discrete Wavelet Transform

FAST FAst rate allocation through STeepest descent

FAST-TVC FAST for time-varying channels

HVS Human Visual System

JPIP JPEG2000 Interactive Protocol

MMAX Minimization of the MAXimum MSE

MMSE Minimization of the average MSE

MRP Magnitude Refinement Pass

NL Numerically Lossless

SPP Significance Propagation Pass

V1 primary Visual cortex reception fields

VBR Variable Bit Rate

VT Visibility Threshold

Every day, videos and images are transmitted over the Internet. Image compression allows to reduce the total amount of data transmitted and accelerates the delivery of such data. In video-on-demand scenarios, the video has to be transmitted as fast as possible employing the available channel capacity. In such scenarios, image compression is mandatory for faster transmission. Commonly, videos are coded allowing quality loss in every frame, which is referred as lossy compression. Lossy coding schemes are the most used regime for Internet transmission due its high compression ratios. Another key feature in video-on-demand scenarios is the channel capacity. Depending on the capacity a rate allocation method decides the amount of data that is transmitted for every frame. Most rate allocation methods aim to achieve the best quality for a given channel capacity. In practice, the channel bandwidth may suffer variations on its capacity due traffic congestion or problems in its infrastructure. This variations may cause buffer under-/over-flows in the client that causes pauses while playing a video. The first contribution of this thesis is a JPEG2000 rate allocation method for time-varying channels. Its main advantage is that allows fast processing achieving transmission quality close to the optimal. Although lossy compression is the most used to transmit images and videos in Internet, when image quality loss is not allowed, lossless compression schemes must be used. Lossless compression may not be suitable in scenarios due its lower compression ratios. To overcome this drawback, visually lossless coding regimes can be used. Visually lossless compression is a technique based in the human visual system to encode only the visually relevant data of an image. It allows higher compression ratios than lossless compression achieving losses that are not perceptible to the human eye. The second contribution of this thesis is a visually lossless coding scheme aimed at JPEG2000 imagery that is already coded. The proposed method permits the decoding and/or transmission of images in a visually lossless regime.