Universitat Autònoma de Barcelona

Departament d'Enginyeria de la Informació i de les Comunicacions

PREEMPTIVE STRATEGIES FOR DATA TRANSMISSION THROUGH JPEG2000 INTERACTIVE PROTOCOL

SUBMITTED TO UNIVERSITAT AUTÒNOMA DE BARCELONA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

by José Lino Monteagudo Pereira

Bellaterra, July 2013

Directed by
Dr. Francesc Aulí Llinàs
and Dr. Joan Serra Sagristà

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, July 2013

Dr. Francesc Aulí Llinàs and Dr. Joan Serra Sagristà
(Supervisors)

Committee:

Dr. Michael W. Marcellin
Dr. Ian Blanes García
Dr. Víctor Francisco Sánchez Silva
Dr. David Megías Jiménez (substitute)
Dr. Ferran Marqués Acosta (substitute)

To Emma and María

Abstract

Nowadays, with the advent of information technology and communications, images are widely used in many areas of our life. When sharing or transmitting images, the network bandwidth is a major concern, especially for large resolution images. In a client-server scenario, the bandwidth consumption increases along with the number of images requested by a user and with the number of users. Thus, efficient transmission strategies are needed to reduce the transmission cost and the client's response time. Efficiency can be achieved through compression and by using a suitable transmission protocol. JPEG2000 is a state-of-the-art image compression standard that excels for its coding performance, advanced features, and for its powerful interactive transmission capabilities.

The JPEG2000 Interactive Protocol (JPIP) is key to achieve fluent image browsing and to minimize the information exchanged in a client-server scenario. Furthermore, the efficienty of JPIP can be improved with: 1) appropriate coding parameters; 2) packet re-sequencing at the server; 3) prefetching at clients; and 4) proxy servers over the network. Prefetching strategies improve the responsiveness, but when clients are in a local area network, redundancies among clients are commonly not exploited and the Internet connection may become saturated.

This work proposes the deployment of prefetching mechanisms in JPIP proxy servers to enhance the overall system performance. The proposed JPIP proxy server takes advantage of idle times in the Internet connection to prefetch data that anticipate potential future requests from clients. Since the prefetching is performed in the proxy, redundancies among all the clients are considered, minimizing the network load. Three strategies are put forward to reduce the latency. The first strategy considers equal probability for next movements. The second strategy uses a user-navigation model. The third strategy predicts the regions of the images that are more likely to be requested employing a semantic map. All these strategies are implemented in our open source implementation of JPIP named CADI, which is also a contribution of this thesis.

Hoy en día, con el desarrollo de las tecnologías de la información y las comunicaciones, las imágenes son ampliamente utilizadas en muchos ámbitos de nuestra vida. Cuando se comparten o transmiten imágenes, el consumo de ancho de banda de la red es importante, especialmente para imágenes de gran resolución. En un escenario cliente-servidor, el consumo de ancho de banda aumenta con el número de imágenes solicitadas por el usuario y con el número de usuarios. Por lo tanto, se necesitan estrategias de transmisión eficientes para reducir los costes de transmisión y los tiempos de respuesta de los clientes. Se puede alcanzar una mayor eficiencia a través de la compresión y también mediante el uso de un protocolo de transmisión adecuado. JPEG2000 es un estándar de compresión de imagénes que está a la vanguardia y que destaca por su rendimiento en la codificación de imágenes, características avanzadas y por sus potentes capacidades para la transmisión interactiva de imágenes.

El protocolo *JPEG2000 Interactive Protocol (JPIP)* destaca porque es capaz de lograr una navegación fluida y porque minimiza la información intercambiada entre el cliente y el servidor. Además, la eficiencia de JPIP puede mejorarse mediante: 1) los parámetros de codificación apropiados; 2) la reorganización de paquetes en el servidor; 3) el prefetching en los clientes; y 4) el despliegue de servidores proxy en la red. Las estrategias de prefetching mejoran la capacidad de respuesta, sin embargo, cuando los clientes se encuentran en una red de área local, no se aprovechan las redundancias entre los clientes y la conexión a Internet puede llegar a saturarse.

Este trabajo propone el despliegue de mecanismos de prefetching en los proxy JPIP para mejorar el rendimiento global del sistema. El proxy JPIP aprovecha los instantes de inactividad de la conexión a Internet para precargar datos y anticipar posibles peticiones futuras de los clientes. Dado que el prefetching se realiza en el proxy, se tienen en cuenta las redundancias existentes entre todos los clientes, lo que minimiza la carga de la red. Tres son las estrategias de prefetching que se estudian en esta tesis para reducir la latencia. La primera estrategia considera probabilidades equiprobables para los futuros movimientos de los clientes. La segunda estrategia utiliza un modelo basado en el comportamiento de navegación del usuario. La tercera estrategia predice las regiones de las imágenes que tienen una mayor probabilidad de ser solicitadas basándose en el contenido semántico de la imagen. Todas estas estrategias están integradas en nuestra implementación de código libre de JPIP llamada CADI, que es también una contribución de esta tesis.

Acknowledgements

This dissertation is fruit of the work carried out in the last three years and it represents a milestone in my life. It seems an individual work, however, it would be meaningless without the collaboration, encouragement, feedback, and patience from a lot of people. Therefore, I am deeply appreciative of those who have helped me along this way.

First and foremost I wish to thank my advisors, Joan Serra and Francesc Aulí, for their support and constant encouragement during the development of this thesis. I thank Joan for the chance he gave me to join the Group on Interactive Coding of Images and from whom I have learnt a great deal on how to deal with scientific research. His desire for excellence and leadership are impressive. Thanks to Francesc that helped me to come up with this dissertation providing assistance with brilliant ideas and guiding me through the writing of papers. *Francesc, ara sí que ja està!*

I am grateful to all the members of the Department of Information and Communications Engineering, where I met amazing people and with whom I shared so many good experiences. In particular, María and Xoanet, my office mates with whom I spent many hours, Fernando for his help in the research and the talks in the train, Jorge for the supportive in my time of need and the good times shared at the EUI, and Juan for his unconditional friendship.

Going back to my days at the Universidade de Vigo, I would also acknowledge to all the members of the Group on Image Processing and Virtual Reality where I did my undergraduate degree project and I have started my post-graduate studies. It has been a pleasure having you not only as workmates but also as good friends. Thank you to Xulio, FMartin, Juancho, José Luis, Patri, Manuel, Javi, Paco, and Curro.

I am indebted to my loving family, those who are and also those who unfortunately

are gone. I am very fortunate to have a unique and amazing family whose support has been unconditional whenever I needed it.

Now, I need to express my gratitude to them in Spanish and Galician.

Pra min é realmente unha honra contar ca familia que teño e que sempre me apoiara cando o precisei. Somentes hai que ver a miña nai, unha auténtica loitadora que somentes pensa en traballar e tirar cara adiante. Tamén teño que agracederlle á miña irmá, ca que teño un especial vencello, que sempre estivera ahí pra o que fose. A Rosana que teña a paciencia que hai que ter pra aturarme todos os días, non é doado. E por último, ás peques porque sempre che alegran o día coas súas ocorrencias: Padrino, hoy no tengo ganas de hablar, hablamos mañana vale? , pues valrá, Emma; e María, padrino, yo sí que quiero hablar contigo, tengo que contarte una cosita... . Moitísimas grazas a toda a familia porque parte deste traballo perténcevos.

Non poido esquencerme de Álvaro, compañeiro da universidade e amigo dos de verdade. As charlas con el soen ser longas e moi productivas, sempre acabas aprendendo algo novo.

Mi agradecimiento para Jorgevich y Moni, grandes amigos que siempre me han apoyado no sólo en esta faceta de estudiante sino muy especialmente en la personal.

También quiero tener un pequeño recuerdo para los que fueron mis compañeros de piso durante algunos años. A Mary, Alberto, y Berta por los buenos momentos, que no han sido pocos.

Y por último, extender también el agradecimiento a mis compañeros de trabajo que con sus las charlas, debates, bromas, y chistes hacen más llevadero el día a día.

Once again, thank you so much.

Contents

Chapter 1

Introduction

Imagery has reached a significant role in a large variety of disciplines and is essential in some of them. The capturing devices can be as simple as a smart phone, or as sophisticated as a satellite or a medical scanner. The widespread interest of images has reached many applications in diverse areas, including remote sensing [1, 2, 3], medical diagnosis and treatment [4, 5], surveillance [6], astronomy [7], civil infrastructure, or military.

The increasing demand of new types of images and new fields of interest has led to the development of modern devices and new imaging technologies that are able to capture high resolution images in terms of spatial and spectral resolution. Such improvements have resulted in an increase in the size of the images, some of them even reaching terabytes in size. Consequently, an increasing demand on storage space and high-bandwidth network communications has arisen. Image compression systems are gaining significant attention because of the need for compression formats that can efficiently store large images and reduce transmission costs.

JPEG2000 standard, created by the Joint Photographic Experts Group (JPEG), stands out as a state-of-the-art image compression system for its excellent features and capabilities. Part 1 [8] of the standard defines the core coding system, which is the basis of the other parts of the standard. It is a wavelet based coding technology with a two-tiered coding scheme built on the EBCOT paradigm [9]. Among other features, it offers high compression ratios, four types of scalability, multi-component support, and a rich codestream syntax that allows random access to the codestream.

1.1 Interactive Browsing of Images

The first attempts to share and distribute images transmitted the whole image from a remote server to a client that processed and rendered the regions requested by the user. Due to the long transmission time and resources needed at the client, subsequent strategies were focused on a progressive transmission where clients received a coarse quality image that was progressively rendered as new data were received [10]. Although the browsing experience was enhanced, the images were still transmitted at full size and processed at the clients, which is inconvenient when dealing with large images. A better approach is the interactive browsing, where the users browse images that are stored in remote servers and are delivered to clients through a network. Interactive scenarios reduce response times, which is fundamental to achieve a fluent browsing experience. Also, response times may be decreased by increasing the bandwidth of the network communication and/or by using image compression techniques.

The first applications devised to transmit images interactively over Internet appeared in the 90s. They were based on standards such as JPEG [11] or GIF [12]. These applications were not efficient because these standards do not allow individual access and transmission of regions of interest. So the whole image needs to be delivered. This shortcoming was overcome through the use of three techniques: 1) compression of images at different resolutions and qualities, 2) indexing strategies and, 3) use of specific transmission protocols. One example of this approach is MiraMon [13]. Unfortunately, images at different resolutions and qualities have to be stored redundantly, and interactivity is strictly restricted to the image regions defined off-line.

Aimed to overcome these drawbacks, a consortium of several companies developed in 1996 the image format FlashPix [14] and the transmission protocol Internet Imaging Protocol (IPP) [15]. FlashPix permits image panning thanks to the use of a hierarchical structure to store images, whereas, IIP is a protocol to interactively transmit FlashPix images. Its drawback is low compression efficiency, since images are stored at different resolutions.

Later on, Lizardtech [16] and ER Mapper [17] developed MrSID and ECW, respectively. Both image formats are based on the wavelet technology, supporting access to a region of the image at different resolutions and interactive browsing. However, they are raster image formats without compression. Furthermore, they are not open standards.

JPEG2000 Interactive Protocol

The JPEG2000 coding scheme provides a set of features such as scalability by spatial location, by quality, by resolution, and by component, efficient transmission, and spatial and spectral random access to the compressed codestream. This high flexibility makes JPEG2000 one of the most suitable image coding systems in an interactive image transmission scenario. Nevertheless, the Part 1 of the standard defines only the codestream syntax, neither describing the architecture nor the protocol to interchange JPEG2000 codestreams efficiently.

The first approach addressed to exploit the JPEG2000 features for interactive transmission was introduced by Deshpande and Zeng [18] son after the publication of Part 1. Their proposal is focused on Web servers using the HTTP protocol [19]. More specifically, they use the byte-range capability of the HTTP to transmit segments of the compressed codestream. The main advantages of this solution are that it runs over a widely extended protocol, without requiring specific implementations, and that uses generic Web servers that support the byte-range capability. Nevertheless, its main drawback is that, before starting the interactive browsing, the clients need to download an index file containing a table of byte ranges for the different image resolutions and spatial locations within the codestream. So there exists an initial delay that depends on the index file size and the network bandwidth. Furthermore, the computational load is unbalanced against the client.

Two other approaches that take advantage of the Web server technology were proposed by Lian and Chan [20] and by Ortiz et al. [21, 22]. The former approach suggests a streaming mechanism through the HTTP protocol combined with the splitting of the codestream into layers to improve the throughput of the network. The main idea behind the later approach is to limit and fix the image file structure. Although the server is a conventional Web server, clients are more complex and the restrictions in the file structure reduces the JPEG2000 flexibility.

Another strategy that is conceptually different from the aforementioned approaches was devised by Li and Sun [23]. In that approach, applications access a virtual representation of the image built at the client side by means of the so-called virtual media access protocol (Vmedia). The Vmedia protocol allows the use of generic HTTP servers or specific implementations (Vmedia servers) that improve the transmission. In addition, an index file similar to the Deshpande's approach may be used to enhance the transmission efficiency. The main drawbacks of this strategy are similar to those found in the Deshpande and Zeng's approach.

Several other approaches were proposed in the literature to interactively transmit JPEG2000 images. Finally, an evolution of the JPIK protocol [24, 25] was standardized by ISO/IEC [26] as Part 9 of the JPEG2000 standard [27]. JPEG2000 Part 9 defines versatile syntax for the client requests and the server response along with methods for the interchange of imagery or imagery related data in an interactive browsing scenario. The protocol has been adopted in many fields such as remote sensing [28], medicine [29, 30], astronomy [7], or video-on-demand [31, 32, 33].

The main component of JPEG2000 Part 9 is a network protocol, referred to as JPEG2000 Interactive Protocol (JPIP), for the interactive and progressive transmission of JPEG2000 images. It has the architecture illustrated in Figure 1.1, which adopts a generic client-server model that provides a clean separation of functions: the server provides services, whereas the client consumes these services.

In the client side, the application may be a browser or any graphical user interface with which the user interacts. The flexibility of JPIP allows users to perform a selective access to a region of the image, called Window of Interest (WOI), and imagery metadata downloading on only those segments of the codestream related to the user request. Rather than interact directly with the codestream, the application sends the request to the JPIP Client and the Decompress/Render modules. The JPIP

Figure 1.1: JPIP protocol architecture. The server side is composed of a JPIP Server, the core of the server, a Cache Model to keep track of data sent to clients, and the images to be served. The client side is composed of the end-user application, the JPIP Client that carries out the communication with the server, a Client Cache where data received are stored, and the image decompressor.

Client manages communications with the JPIP server, sending the requested WOI to the server, receiving the segments of the codestream delivered by the server, which are then stored in the client cache. The Decompress/Render module renders the requested WOI from the cached data without having to wait for new data to arrive.

Regarding the JPIP server, it receives the requests from clients and transmits the response. The server reads the data associated with the codeblocks corresponding to the requested WOIs from the codestream, builds the appropriate packets, and delivers the response. An important feature of the server is the Cache Model, which is an optional module that maintains a model of the client's cache to avoid sending data previously delivered.

The JPIP protocol was designed to be independent of the underlying transport protocol, however, it usually works over HTTP/1.1 [19]. When using HTTP as the transport layer, requests are always encapsulated in a HTTP request and the response is delivered as a HTTP response message or transmitted over a TCP auxiliary channel.

JPIP defines a new level of abstraction for the partitioning scheme of a JPEG2000 image file. The new container, named *data-bin*, is constituted of segments from the JPEG2000 compressed image having a natural linear organization. It is considered the smallest data unit of a JPEG2000 image for JPIP. The standard defines a set of data-bin types depending on the JPEG2000 file box that encapsulates. They are precinct data-bin, tile header data-bin, tile data-bin, main header data-bin, and metadata-bin.

Although the basic partition scheme is the data-bin, the JPIP defines two new media types for sending the server responses: a JPP-stream used for precinct-based data-bins, and a JPT-stream for tile-based data-bins. Each stream consists of a consecutive sequence of messages, and each message is composed of a header and a segment from a single data-bin. Message headers are designed to be self-contained, allowing out-of-order transmission and not being affected if one of the messages is lost.

As previously stated, all data transmitted by the server are cached by clients and the server may maintain a model of the client's cache. The cache model depends on the capabilities defined by the clients and the statefulness of the JPIP session. JPIP distinguishes between two types of request: stateless requests and requests that belong to a session (stateful). The aim of stateful request is to avoid redundant transmission of data. When operating within the context of a session, the server keeps track of the data already sent to a client in response to a previous request. A session also allows that clients send a new request before receiving the current response. They may also stop the current response. Contrarily, stateless requests are not associated with any session and the client must explicitly indicate in each request the state of its cache so that the server does not retransmit data that the client already has. Evidently, stateful connections are more convenient to minimize the transmission of side information [34]

The flexibility offered by the JPIP protocol allows the server to disassemble and transcode the original codestream into a collection of data-bins, offering a new image partition strategy or modifying image attributes according to the server criterion. This freedom is extended to the point that client requests can even be modified.

1.2 Motivation

The architecture of systems for the interactive remote browsing of images commonly adopts a client-server model where images are stored in a remote server and applications allow users to perform panning and zooming operations on those images. JPIP fulfills the requirements of this scenario. Several JPIP compliant strategies that improve the efficiency of a conventional JPIP server have been proposed in the literature. First, efficiency may be improved if image compression parameters are well determined [29]. Second, a strategy carried out by servers is to dismantle and re-sequence the original codestream to maximize the quality of the delivered image [35, 36]. Third, the use of prefetching strategies at the clients achieves a faster displaying of the requested regions due to the prediction of the next user movements [37, 38]. Fourth, the deployment of proxy servers at different points of the communication network reduces the load of the server and the client's response time [39, 22].

Commonly, users of a company or an institution within the same local area network (LAN) browse images located in remote servers sharing the same Internet connection. In general, whichever is the Internet connection, the channel capacity of the LAN is many times larger and more reliable than that of the Internet. Remote browsing of images is characterized by high bandwidth demand with burst traffic patterns, so a fluent interactivity with acceptable responsiveness for end users may compromise the network load. In such a scenario, caching mechanisms are extensively discussed in the literature as a means to reduce client's response times as well as to reduce the server load and network bandwidth usage. Caching strategies can be carried out by proxy servers that are not only able to cache and reuse data transmitted from servers to clients, but also to increase availability and reduce network traffic [40, 41, 42]. In the context of JPEG2000, proxy servers were first introduced in [22]. This approach is focused on taking advantage of the conventional HTTP proxy infrastructure deployed in companies and over Internet to support the JPIP protocol. Nonetheless, it is not compliant with JPIP despite the fact that it inherits some features of the JPEG2000 standard. The first compliant JPIP proxy server was proposed in [39], focusing on re-formulating the client's requests to optimize the quality of the recovered image as data are being received by clients.

As previously stated, prefetching strategies at the clients reduce the perceived latency and henceforth improve the overall user experience. In interactive remote browsing systems, once a user has requested a region of the image, there exists a delay between the data transmission from the server and the next request performed by the user [37, 43, 44] during which the connection between the client and the server is idle. Such idle times may be employed by a prefetching mechanism to fetch data of potential future requests. Nevertheless, this approach may not be recommendable in the context of a company or institution due to the fact that 1) it might compromise the channel capacity of the Internet connection, and 2) it does not take into account the redundancies among data transmitted from the server. These shortcomings can be mitigated by means of a proxy server in the network infrastructure. Unfortunately, none of the JPEG2000-oriented proxy servers that appeared in the literature until 2008 were JPIP compliant or included prefetching mechanisms. The motivation behind this work is to develop a JPIP proxy server that supports prefetching mechanisms and that keeps compliance with the standard.

To achieve this purpose the proposed JPIP proxy server must be compliant with the JPIP protocol so that it can assume all the functionalities provided by a JPIP server such as understanding the JPIP syntax, managing JPEG2000 images in the image and compressed domains, and maintaining a model of the client's cache. Employing prefetching mechanisms in a JPIP proxy server, rather than in the clients, alleviates the amount of channel resources needed and permits to exploit the redundancies among clients. One important aspect of the prefetching is the WOI prediction, which is responsible for deciding what data should be pre-fetched. This approach provides a smart solution for users within a local area network that share the same Intenet connection for interactive browsing of JPEG2000 images that are located in a remote server.

1.3 Contributions and thesis organization

The contributions of this thesis are the evaluation of the interactive transmission of spectrally-transformed hyperspectral images, and the development of proxy servers with prefetching strategies within an interactive JPEG2000 context. To achieve this, we have implemented an open-source JPIP server and proxy server that includes prefetching strategies. This JPIP server is named CADI and is an ancillary contribution of this thesis.

The first scientific contribution of this thesis assesses the influence of spectral decorrelators, specifically those based on Discrete Wavelet Transform (DWT), for the interactive transmission of images. The DWT has been chosen as decorrelator because of its low computational complexity and excellent features such as component and resolution scalability. We employed a corpus from the remote sensing field because these images have a large number of spectral components. Detailed analysis and results are presented and discussed in Chapter 2 of this document, which corresponds to the paper

> J. Lino Monteagudo-Pereira, Joan Bartrina-Rapesta, Francesc Aulí-Llinàs, Joan Serra-Sagristà, Alaitz Zabala and Xavier Pons, Interactive transmission of spectrally wavelet-transformed hyperspectral images," In Proceedings of SPIE Conference on Satellite Data Compression, Communication, and Processing IV, Vol. 7084, August 2009, pp. 708 405, 1 12.

The second scientific contribution of this thesis is focused on the study and implementation of prefetching strategies in a JPIP proxy to improve the overall system's performance. The capabilities of a JPIP proxy server located within the local area network are extended by means of three prefetching strategies.

Our first approach proposes a simple prefetching strategy carried out in a JPIP proxy server that uses the idle times of the Internet connection to request potential future regions of the image. This approach considers that all possible movements of clients have the same probability to be executed, thus it carries out the prefetching with neither taking into account the most likely movement that may be requested by the users nor the nature of the images. This work is included in Chapter 3.2 and was published in

> J. L. Monteagudo-Pereira, F. Aulí-Llinàs, J. Serra-Sagristà, and J. Bartrina-Rapesta, "Smart JPIP proxy server with prefetching strategies," In Proceedings of the IEEE Data Compression Conference, March 2010, pp. 99 108.

Our second prefetching strategy improves the model of probabilities. The main idea behind it is to analyze the behavior of users when browsing images and then elaborate an accurate user-navigation model. A detailed explanation of this approach can be found in Chapter 3.3, which was publised in

> J. Lino Monteagudo-Pereira, Francesc Auli-Llinas, Joan Serra-Sagristà, Alaitz Zabala, Joan Maso, and Xavier Pons, "Enhanced transmission of JPEG2000 imagery through JPIP proxy and user-navigation model," In Proceedings of the IEEE Data Compression Conference, April 2012, pp. 22 31.

Our last approach considers the content, or the semantics, of the images, elaborating a map with those regions of the image that have different probabilities to be visited. This process may be performed by means of an automatic classification or by the supervision of an expert in the field. These three approaches, along with a detailed comparison of their results, are described in Chapter 3.4, which was published in

> Jose Lino Monteagudo-Pereira, Francesc Auli-Llinas, Joan Serra-Sagrista, "JPIP Proxy Server with Prefetching Strategies Based on User-Navigation Model and Semantic Map," IEEE Transactions on Multimedia, 2013, in Press.

An ancillary contribution of this thesis is the development of the JPIP implementation CADI [45]. It is an open-source implementation programmed in Java and distributed under the General Public Licence (GPL). The implementation follows a client-server architecture and is composed of a graphical user interface, a JPIP client, a JPIP proxy, and a JPIP server. A description of the implementation, as well as documentation and manuals, are available on-line at

> http://gici.uab.es/CADI
>
> http://sourceforge.net/projects/cadi

CADI Software has been downloaded more than 1600 times from the SourceForge site [46] since the first stable version was released on January 16, 2008. CADI has been tested in different scenarios such as remote sensing, medical, surveillance, as well as by the research community.

Chapter 2

Interactive transmission of spectrally-transformed images

2.1 Introduction

This work evaluates the transmission of spectrally transformed images and describes the architecture of our JPIP implementation.

Hyperspectral images used in remote sensing commonly contain hundreds of spectral bands. Each component represents the same spatial location captured at different wavelengths. This type of images is characterized by the high degree of redundancy in the spectral domain. Several image coding techniques have been employed to reduce such redundancy [47, 48, 49, 1, 50, 51, 52].

This study employs the DWT as the spectral decorrelator. The influence of the DWT parameters (kernel and decomposition levels), as well as scalability issues are evaluated. Experimental results suggest that to apply a spectral transform achieves a superior performance when a group of components is transmitted. When only one component, or a reduced group, is transmitted, the best strategy is to do not apply any spectral transform. Moreover, the shorter the kernel and the lower number of decomposition levels applied, the better the coding performance when a group of contiguous components is requested.

2.2 Interactive transmission of spectrally wavelet-transformed hyperspectral images

```
@inproceedings {MBASZP09,
      AUTHOR = {J. Lino Monteagudo-Pereira, Joan Bartrina-Rapesta, Francesc Aulí-Llinàs,
                Joan Serra-Sagristà, Alaitz Zabala and Xavier Pons},
       TITLE = {Interactive transmission of spectrally wavelet-transformed hyperspectral images},
   BOOKTITLE = {Proceedings of SPIE, Conference on Satellite Data Compression, Communication,
                and Processing IV},
   PUBLISHER = {SPIE Press},
      VOLUME = {7084},
       MONTH = {Aug.},
        YEAR = {2009},
       PAGES = {708 405, 1-12},
        ISSN = {9780819473042},
         DOI = {10.1117/12.794749},
}
```

Interactive transmission of spectrally wavelet-transformed hyperspectral images

José Lino Monteagudo-Pereira[a], Joan Bartrina-Rapesta[a], Francesc Aulí-Llinàs[a],
Joan Serra-Sagristà[a*], Alaitz Zabala[b] and Xavier Pons[b,c]

[a] Department of Information and Communications Engineering
[b] Department of Geography
[c] Center for Ecological Research and Forestry Applications

Autonomous University of Barcelona
08290 Cerdanyola del Vallès, Spain

ABSTRACT

The size of images used in remote sensing scenarios has constantly increased in the last years. Remote sensing images are not only stored, but also processed and transmitted, raising the need for more resources and bandwidth. On another side, hyperspectral remote sensing images have a large number of components with a significant inter-component redundancy, which is usually taken into account by many image coding systems to improve the coding performance. The main approaches used to decorrelate the spectral dimension are the Karhunen Loêve-Transform and the Discrete Wavelet Transform (DWT).

This paper is focused on DWT decorrelators because they have a lower computational complexity, and because they provide interesting features such as component and resolution scalability and progressive transmission. Influence of the spectral transform is investigated, considering the DWT kernel applied and the number of decomposition levels.

In addition, a JPIP compliant application, CADI, is introduced. It may be useful to test new protocols, techniques, or coding systems, without requiring significant changes on the application. CADI can be run in most computer platforms and devices thanks to the use of JAVA and the configuration of a light-version, suitable for devices with constrained resources.

Keywords: Hyperspectral data coding, interactive hyperspectral data transmission, JPEG2000, JPIP, quality scalability.

1. INTRODUCTION

Images play a significant role in many fields of our society, notably in remote sensing applications like environmental monitoring, geographical information systems, disaster assessment and management, or land use classification in agriculture.[1] In the last years, advances of imaging sensors and satellites have implied larger amounts of captured data, which are then transmitted back to the earth for dissemination. New technology of imaging sensors captures images at very narrow wavelength, achieving hundreds of contiguous spectral bands, with high spatial and spectral resolutions. These images are subsequently processed, transmitted, and analyzed to retrieve important information. An example of this kind of sensor is the AVIRIS,[2] an airborne hyperspectral sensor that records 224 contiguous bands, from 400nm to 2500nm, of size 2048 rows × 614 columns × 2 bytes/sample per band (over 537 Megabytes per image).

Image compression has become increasingly of interest in both data storage and data transmission from remote acquisition platforms (satellites or airbornes) because, after compression, storage space and transmission time are reduced. Thus, several approaches have been employed for compression of hyperspectral images: vector-quantization-based techniques,[3,4] Karhunen-Loêve-Transform (KLT) applied to spectral pixel vectors,[5,6] extensions of the common 2-dimensional image-compression methods to 3-dimensional methods,[7] or wavelet-based compression systems (JPEG2000,[8,9] SPIHT,[10,11] SPIHT-3D,[12] SPECK-3D,[10] ICER,[13] ICER-3D,[14] etc.).

---

[*] Further author information: Send correspondence to Joan Serra-Sagristà.
Joan Serra-Sagristà: Department of Information and Communications Engineering, ETSE. Universitat Autònoma de Barcelona,
E-08290 Cerdanyola del Valles (Barcelona), Spain. E-mail: Joan.Serra@uab.es, Phone: +34-93-581 30 12, Fax: +34-93-581 44 77.

It is worth noting that our society trends towards remote displaying of images, that is, a scenario where images are stored in remote servers and clients browse on such images. Consequently, achieving acceptable response times when displaying remote images demands for communication networks with high bandwidth and also for image compression techniques. Several approaches have been developed to achieve an efficient image transmission. The first applications focused on interactive image transmission appeared in the 90s, and they were based on the JPEG[15] and GIF[16] standards. But the main drawback of these applications were that image region transmission compelled to deliver the whole image and cropping it at the client side. These weak points were overcome by means of 1) compression of images at different resolutions and qualities, and 2) the use of specific protocols to deliver images. Despite that some weak points were overcome, the server was forced to have images stored in several resolutions and qualities. Consequently, it was an overload for the server and the interactivity was restricted to off-line-defined image regions. In 1996, a consortium of several companies developed both and image format, FlashPix,[17] and a transmission protocol, Internet Imaging Protocol(IIP).[18] FlashPix is based on a hierarchical structure of the image at different resolutions, which allows to transmit regions of the image, and the IIP protocol defines a protocol to interactively transmit FlashPix images. The main disadvantage is also that images must be stored at different resolutions. On the other hand, MrSID, developed by Lizardtech,[19] and ECW, from ER Mapper,[20] are raster compression formats based on wavelet technology which offer multi-resolution, progressive transmission, and interactive zooming, but their main drawback is that they are not open standards.

The JPEG2000 standard, developed by the Join Photographic Experts Group, is composed by 12 parts including image coding (Part-1[21] and Part-2[22]), transmission (Part-9[23]), security and video. JPEG2000 is a powerful standard which supports multi-component, variable bit depth per component, flexible canvas coordinate system, different progression orders, regions of interest coding, etc., which make JPEG2000 an image coding system with many degrees of flexibility. The core coding system of JPEG2000 is constituted by four main stages: sample data transformations, sample data coding, rate-distortion optimization, and code-stream re-organization. The *sample data transformations* stage tiles the input images data, shifts unsigned image values to make them zero mean, compacts the energy of the image through the application of several decomposition levels of Discrete Wavelet Transform (DWT), and quantizes transformed coefficients. Then, the image is logically partitioned in code-blocks that are independently coded by the *sample data coding* stage, called Tier-1. Tier-1 generates one quality embedded code-stream for each code-block, hence the need of *rate-distortion optimization* techniques to select the best code-stream segments of code-blocks to construct the final code-stream for a given target bit-rate. The last stage of the core coding system is the *code-stream re-organization*, which allows the creation of successive layers of quality and, through the Tier-2, encodes the auxiliary data needed to identify layers' contents.

Part-9 of the JPEG2000 standard, JPEG2000 Interactive Protocol (JPIP), is a client-server architecture for interactive image transmission. It defines a request-response syntax that allows clients to select portions of an image, called Windows of Interest (WOI), which are delivered from the server to clients eliminating redundant representation of the image and redundant transmissions. This efficient transmission is achieved because the delivered data are segments of the code-stream. The aforementioned request syntax allows clients to include query fields describing many possibilities, for instance, image name, component, frame size, region, layers, session, status of client cache, etc.

This work has a twofold purpose. On the one hand, it is focused on evaluating transmission of spectrally wavelet transformed images against transmission of images without transformation across the spectral components. This evaluation is conducted within the framework of JPEG2000 standard. On the other hand, the development of a JPIP implementation is presented. It provides a good framework to test new protocols, algorithms, coding techniques, and even whole coding systems, because of its high and modular flexibility.

This paper is structured as follows. Section 2 contains an overview of hyperspectral image coding, mainly on image compression exploiting the spectral dimension by means of the wavelet technology, within JPEG2000. Different rate-allocation strategies to generate JPEG2000 compliant files from the compressed hyperspectral images are also explained in this section. Interactive image transmission is reviewed in section 3, centered on the JPIP protocol. It also tackles transmission of hyperspectral images with and without transform across the spectral domain. In addition, a JPIP compliant application, CADI, used to carry out the experiments, is introduced. Then, section 4 presents some experimental results. The last section contains a summary and draws some conclusions.

2. REVIEW OF HYPERSPECTRAL IMAGE CODING

Hyperspectral images usually have a similar global structure across components. However, different pixel intensities could exist among nearby spectral components or in the same component due to different absorption properties of the atmosphere or the material surface being imaged. This means that two kinds of correlations may be found in hyperspectral images: intraband correlation among nearby pixels in the same component, and interband correlation among pixels across adjacent components. Interband correlation should be taken into account because it allows a more compact representation of the image by packing the energy into fewer number of bands, enabling a higher compression performance.

There are many technologies which could be applied to remove correlation across the spectral dimension, but two of them are the main approaches for hyperspectral images: the KLT and the DWT. (The Karhunen-Loêve-Transform is also known as Principal Components Analysis). Although the KLT achieves the best results,[24,25] it is not always an attractive decorrelator due to its high computational complexity. Conversely, the DWT provides a reasonable computational complexity as well as some interesting features like component and resolution scalability and support for progressive transmission.

2.1 Compression of spectrally wavelet-transformed images with JPEG2000

As have been previously explained, the JPEG2000 standard is a powerful image coding system that allows many degrees of flexibility for compressing images. Although JPEG2000 was designed to allow compression of images with up to 16385 spectral components in a single code-stream, it does not indicate how spectral components should be encoded to achieve the best compression performance. While the Part-1 of the standard does not allow decorrelation across the spectral dimension (except for decorrelation of color components for RGB images), the Part-2 allows that any decorrelation might be applied across the spectral components, including both the KLT and the DWT.

As said, hyperspectral image compression performance is improved when interband correlation is taken into account.[8,26] In this case, and considering only wavelet-based spectral decorrelators, two possibilities appear, depending on how the hyperspectral image is considered, either the whole image as a cube, or considering separately the spectral and the spatial dimensions. The first possibility could be understood as a direct extension of the 2-D transform to 3-D. Table 1 below summarizes three kinds of wavelet-based decorrelators which could be applied to hyperspectral images, which are graphically presented in figure 1.

3D square transform	3D rectangular transform	3D hybrid
It is obtained by performing one decomposition level of a one-dimensional transform in each dimension. Then, this process is successively applied on the low-frequency cube. The resulting decomposition is a 3-D Mallat decomposition.	It is obtained by performing all decomposition levels of the wavelet transform along one dimension, then successively along the other desired dimensions.	It is obtained by performing all decomposition levels of the wavelet transform across the spectral dimension, and then a 2-D square transform is carried out independently for each spectrally transformed component.

Table 1: Types of three-dimensional DWT.

Penna *et al.*,[27] Fowler *et al.*[8] and Serra *et al.*[26] have shown that the 3D hybrid rectangular / square transform outperforms the 3D square for hyperspectral images. Consequently, the 3D hybrid rectangular / square has been selected here.

2.2 JPEG2000 rate-allocation strategies for multi-component images

As stated, JPEG2000 is divided in four stages. In the second stage, *sample data coding*, quantized coefficients are logically grouped into rectangular blocks, called code-blocks, following the scheme proposed in EBCOT,[28] and each code-block is independently coded by an entropy encoder obtaining one code-stream for each. A rate-distortion technique is needed to truncate and select the best code-stream segments for each code-block. In the last stage, *code-stream re-organization*, code-streams are merged and arranged to form the final code-stream. It is worth noting that the JPEG2000 standard does not specify how each code-stream must be truncated and arranged. However, the EBCOT algorithm produces the rate-distortion optimal segments of code-streams by means of optimal truncation points for each independent code-stream. Then segments of code-streams are concatenated to form the final code-stream following the order set by, for instance, the Post-Compression Rate-Distortion Optimization (PCRD-Opt) algorithm.

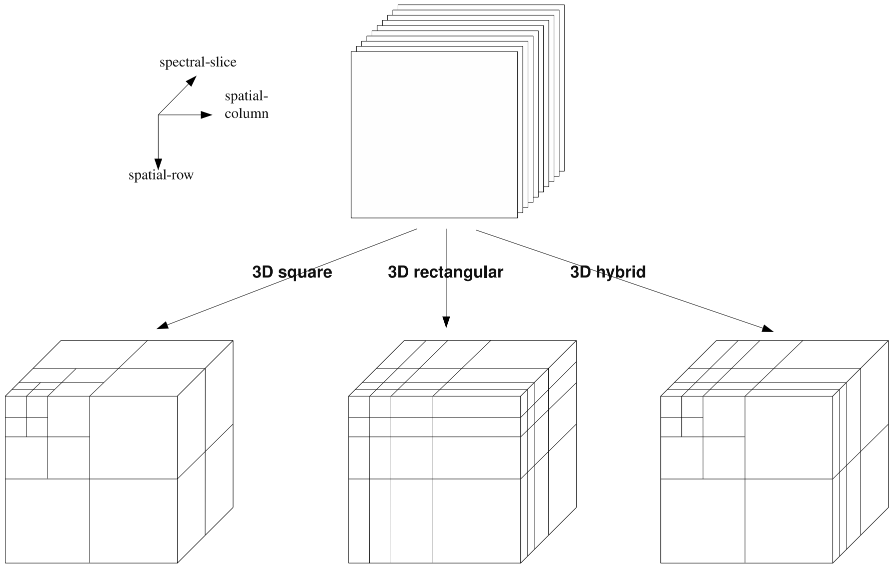


Figure 1: Types of three-dimensional DWT, with a 3 level spectral decomposition and a 3 level spatial decomposition.

An important issue that must be considered when a multi-component image is coded with JPEG2000 is the rate-allocation strategy. The PCRD-Opt achieves optimal rate-distortion code-streams for a single-component image, but its extension to multi-component images may be done according to three different strategies[8] (in all three cases, the generated final code-stream is JPEG2000-compliant):

Band-Independent Fixed-Rate (BIFR). This method is a band-independent rate-allocation strategy which assigns a fixed rate for each component. See figure 2a.

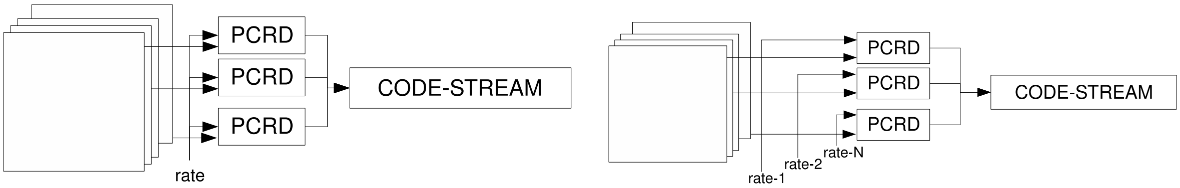
Band-Independent Rate Allocation (BIRA). This method is also a band-independent rate-allocation strategy which assigns a variable rate for each component. Different criteria could be used to allocate the rate for each component, such as energy, variance, or entropy. See figure 2b.

Multi Component (MC). This method performs the PCRD-Opt across the image components. The PCRD-Opt is applied simultaneously to all code-streams of the whole image, computing the truncation points in both spatial and spectral dimensions. This method achieves the best results.[8] See figure 2c.

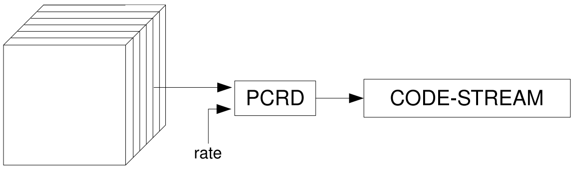
3. JPEG2000 INTERACTIVE TRANSMISSION OF IMAGES

The first approach to interactively transmit JPEG2000 images was proposed in 2001 by Deshpande and Zeng[29] . The main idea behind this first approach is to use the byte-range transmission capability of the HyperText Transfer Protocol (HTTP)[30] to transmit segments of JPEG2000 code-streams. Before the transmission of the code-stream segments, clients must download an index file containing a list of the code-stream byte ranges containing different image resolutions and spatial locations and, once this index file is downloaded, the client requests byte-ranges of the JPEG2000 code-stream through the HTTP protocol. The main advantage of this approach is that any HTTP server can be used and that it does not require the definition of any specific transmission protocol. However, the drawbacks are that the transmission of the index file widely increases the initial response time, and that the computational load is completely unbalanced against the client, which is precisely the one that might have constrained resources.

The second approach proposed for the transmission of JPEG2000 code-streams was introduced by Li and Sun.[31] Basically, this approach creates a virtual representation of the image at the client side that is accessed by applications through the so-called virtual media access protocol (Vmedia). This approach allows both the use of a HTTP server or a Vmedia server that improves the transmission performance. To enhance the transmission over the network, the Vmedia protocol may also use an index file similar to the previous approach. The drawbacks of this approach are similar to the drawbacks



(a) BIFR (b) BIRA



(c) MC

Figure 2: Rate-allocation strategies for hyperspectral images.

of the previous approach, because clients must also request specific chunks of the compressed image, and because the decoding takes place at the client-side.

The unfavourable points arising from these first approaches have been overcome by the JPIP protocol.[23] JPIP defines a simple syntax and several techniques to efficiently transmit JPEG2000 code-streams using very low computational resources. A key-feature of JPIP is that it gives several degrees of freedom to both client and server sides, allowing applications to decide whether to use stateful or stateless connections, cache or non-cache management, etc. This freedom is achieved by the definition of the data-bin container, which is the self-contained structure containing the actual data transmitted over the network. The negotiation between client and server allows the delivery of code-stream segments in different ways depending on the image, the network state, and the server load.

The tremendous flexibility of JPIP allows applications to fulfill a wide range of requirements and necessities. Recently, some studies have been proposed to improve the quality of the transmitted images through smart transcoding techniques[32,33] and a slight extension of the JPIP protocol has been also proposed in[34] to exploit the redundancy between requests of different clients through web proxy cache. An XML based model for the cache management is introduced in[35] aimed to improve the efficiency of applications.

3.1 Transmission of hyperspectral images without spectral transform

When hyperspectral images are compressed considering that each component is independent from the others, inter-component redundancy is not taken into account (thus not achieving the best compression performance). Conversely, each packet in the code-stream is only linked with one, and only one, component of the original image. Therefore, when several components of the image must be transmitted, delivered data belongs only to the requested components. Despite that the achieved coding performance is not the best, it could be of interest in an interactive scenario when users browse on reduced groups of image components because it gives the finest access accessibility to image components.

3.2 Transmission of hyperspectral images with wavelet transform as spectral decorrelator

When hyperspectral images are compressed performing a 1-D wavelet transform across the spectral dimension, and a 2-D wavelet transform across the spatial dimension, the inter-component redundancy is fully exploited. In this case, each spectrally-transformed component is related to several components of the original image, since there is a one to multiple relationship between components of the original image and spectrally transformed components (see an example in figure 3). Consequently, access flexibility to spectral components is decreased because decompressing an image component entails reading and transmitting the necessary components to invert the transform across the spectral domain. Note that the number of spectral components needed to invert the spectral transform depends on the length of the filter and the number of decomposition levels applied. The larger the length of the filter, or the larger the number of decomposition levels, greater is the number of spectral components needed.

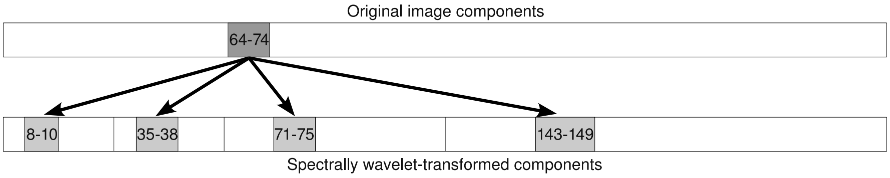


Figure 3: Relation between components of the original image and wavelet-transformed components. The DWT kernel is the 5/3 reversible and 3 decomposition levels have been applied.

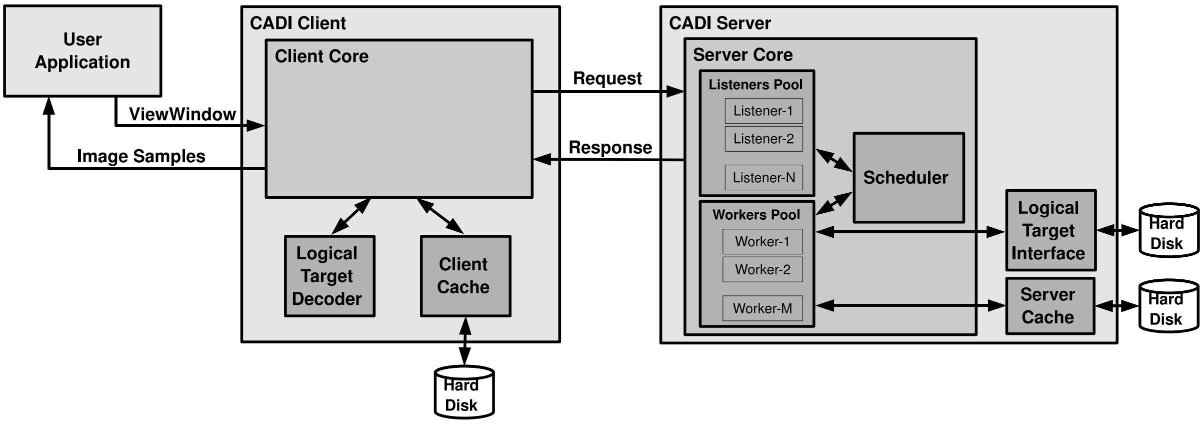


Figure 4: Modular scheme of the JPIP implementation CADI.

3.3 CADI: a JPIP implementation

The Group on Interactive Coding of Images (GICI) has developed an implementation of the JPIP protocol, named CADI. The development of CADI is focused on three main goals. The first one is that CADI has been designed with the aim to get a modular application where adding, modifying, or replacing modules was easy. Consequently, new transmission techniques and algorithms can be tested without requiring significant changes. The second purpose is to achieve a multi-platform application, thus Java has been the chosen programming language. And the last aim is that the client application has been designed to be run on personal computers or on constrained resources devices, so that both full- and light-versions may be enabled.

The CADI Software is composed by three applications. A JPIP server, *CADIServer*, a JPIP client *CADIClient*, and a graphical user interface, *CADIViewer*. Figure 4 shows the main modules of CADI's architecture and philosophy design.

CADIServer: It is composed by three main modules, a main module, the *server core*, and two auxiliary modules, the *logical target interface* and the *server cache*. The *server core* module receives requests of clients and it also sends responses to clients, therefore, each task is performed by two subsidiary and independent modules, *listeners pool* and *workers pool*, which are managed by the *scheduler* module. Both subsidiary modules have been implemented like a pool of threads in order to take advantage of multi-threading architectures and, as a consequence, multiple client requests can be simultaneously processed. The *scheduler* receives client requests form the *listeners pool* and it assigns them to a *worker*, where prioritization criteria could be applied. It is important to emphasize that when a JPEG2000 code-stream is requested by a client, it is indexed by a *worker* and it is kept in a shared memory among *workers* reducing the response time of new requests. Another important module is the *logical target interface*, which defines an interface between the *server core* and code-streams. It allows management of different image coding systems returning information about the image or segments of the code-stream. It is therefore very easy to add support for new image coding systems.

CADIClient: This application implements a JPIP client, a middleware between a graphical user interface (GUI) and a JPIP server. It has one interface to deal with the GUI and one interface to deal with the JPIP server. After receiving a


(a) Cuprite (b) Jasper Ridge (c) Low Altitude (d) Lunar Lake

Figure 5: AVIRIS hyperspectral images used in the experiments. Radiance data (band 25).

request for a particular window or region of the image from the GUI, the JPIP clients parses the request and forwards it to the server, which, in its turn, processes it and delivers segments of the compressed code-stream belonging to the requested WOI back to the client. The *CADIClient* is composed by: 1) a main module, *Client*, which manages the communication with the GUI and the server, 2) the *client cache* module, which manages the cache at the client side, and 3) the *logical target decoder* module, which performs the decoding of the compressed code-stream.

CADIViewer: It is a simple graphical user interface to display images. Users can request an image placed in a remote server and they can browse on the image doing a pan, a zoom, or improving the quality. The CADIViewer can be replaced by any other user application in an easy way because it only requests for a specific area of the image and receives the images samples for the requested area.

4. EXPERIMENTAL RESULTS

Experiments have been conducted to assess and compare the coding performance achieved when hyperspectral images are compressed and transmitted either with, or without, applying a wavelet transform across the spectral dimension. JPEG2000 standard has been taken as the baseline coding system because of its state-of-the-art competitive coding performance. Evaluation of the coding performance when applying an spectral wavelet transform has been carried out considering both the influence of the kind of wavelet kernel, and of the number of wavelet decomposition levels. With regard to the wavelet kernel for the spectral dimension, both the 9/7 irreversible and the 5/3 reversible filters are analysed. With regard to the number of decomposition levels for the spectral dimension, results are only reported for 5 and 3 decomposition levels, although more experiments have been performed. With regard to the 2-D wavelet spatial transform, results are reported for the 9/7 irreversible kernel, always with 5 decomposition levels.

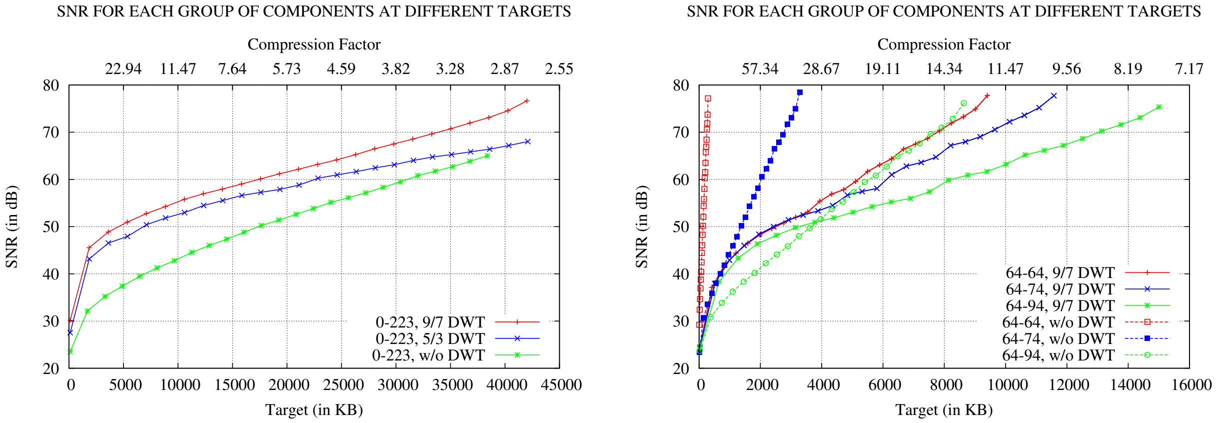
BOI software v1.2,[36] a JPEG2000 Part 1 and Part 2 implementation, has been used to encode the images, and CADI software v1.2[37] has been used to evaluate the transmission within a JPIP framework.

Experiments have been performed on an Airborne Visible/Infrared Imaging Spectrometer[2] radiance data set. AVIRIS sensor records the visible and the near infrared spectrum of the reflected light, and is capable of producing images of size 2048 rows × 614 columns × 224 bands × 2 bytes/sample per flight. For the results here, we crop the first scene in each data set to produce image cubes with dimensions 512 × 512 × 224 pixels, each pixel stored in signed 16 bits per pixel per band (bpppb). In particular, Cuprite, Jasper Ridge, Low Altitude, and Lunar Lake radiance images have been used; one original component of each image is shown in figure 5.

The distortion measure for evaluating the quality of the recovered images is the Signal to Noise Ratio (SNR), defined as

$$SNR_{Energy} = 10 \cdot log_{10} \frac{E(x)}{MSE} \quad (dB)$$

where $E(x) = \frac{1}{N_x N_y N_z} \sum_{i,j,k} x(i,j,k)^2$ is the energy of the input image (of size $N_x \cdot N_y \cdot N_z$), and the Mean Squared Error (MSE) (or L_2^2) is $MSE = \frac{1}{N_x N_y N_z} \sum_i \sum_j \sum_k [x(i,j,k) - \hat{x}(i,j,k)]^2$, with $x(i,j,k)$ denoting a pixel in the input image, and $\hat{x}(i,j,k)$ a pixel in the recovered image.



(a) Transmission of the whole image (224 components).

(b) Transmission of a contiguous set of components.

Figure 6: Performance comparison between two coding approaches: no transform across the spectral dimension versus an spectral wavelet transform approach. Cuprite radiance image.

The first experiment is focused on analysing the transmission of the whole image cube with and without applying a wavelet transform across the spectral domain. Figure 6a depicts the results for the Cuprite radiance image. The compression factor and the length of the received code-stream are reported in the horizontal axis; the SNR quality is reported in the vertical axis. Plots are provided for the 9/7 irreversible filter, the 5/3 reversible filter, and when no spectral wavelet transform has been applied. As reported in the literature, the best coding performance is achieved when the image has been spectrally transformed, because interband redundancy is removed by the wavelet decorrelator. Consistently, the 9/7 kernel provides a superior performance over the 5/3 kernel. The results for the other images of the AVIRIS data set are similar.

The second experiment is focused on analysing the transmission of only a group of components, both with (9/7 irreversible filter with 5 decomposition levels) and without applying a wavelet transform across the spectral domain. Figure 6b depicts the results for the Cuprite radiance image when only one, ten, or thirty contiguous components are transmitted. When transmitting a single component, transmission of images without spectral transform provides a superior performance, because the delivered data corresponds exactly to the requested component, meanwhile data delivered when the image has been spectrally transformed includes data belonging to components that have not been requested but which are needed to invert the wavelet transform. When transmitting thirty contiguous components, and for very low bit rate, the quality of the recovered image is higher when an spectral transform has been applied; for medium to high bit rate, it is better not to apply an spectral transform. As the number of requested components increases, the performance of the spectrally wavelet transform approach improves, because of the ratio between the number of requested components and the number of components needed to invert the DWT.

The third experiment assesses the influence of the wavelet kernel in the image transmission. Specifically, the 9/7 irreversible and the 5/3 reversible kernels have been evaluated. Figure 7 depicts the results for the Low Altitude radiance image when ten, twenty, or thirty contiguous components are transmitted, for both wavelet kernels. The 5/3 kernel yields a better coding performance than the 9/7 kernel because its filter length is shorter, needing a fewer number of spectral components to invert the DWT. However, as the number of requested components gets larger, the differences in coding performance between both wavelet kernels become shorter. Recall also that when the whole image cube is requested, the 9/7 kernel provides a better coding performance (as has been reported in figure 6a).

The fourth experiment assesses the influence of the number of wavelet decomposition levels. Figure 8a reports results for the Cuprite radiance image when the whole image is requested, using 5 and 3 decomposition levels (9/7 irreversible kernel). As expected, applying 5 wavelet decomposition levels yields a higher coding performance. However, when one, twenty, or thirty contiguous components are requested, figure 8b shows that the coding performance is higher when the number of decomposition levels is smaller, because again the number of spectral components needed to invert the spectral wavelet transform is smaller.

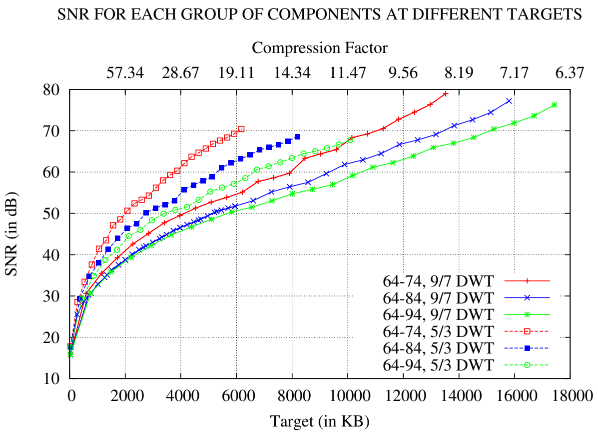
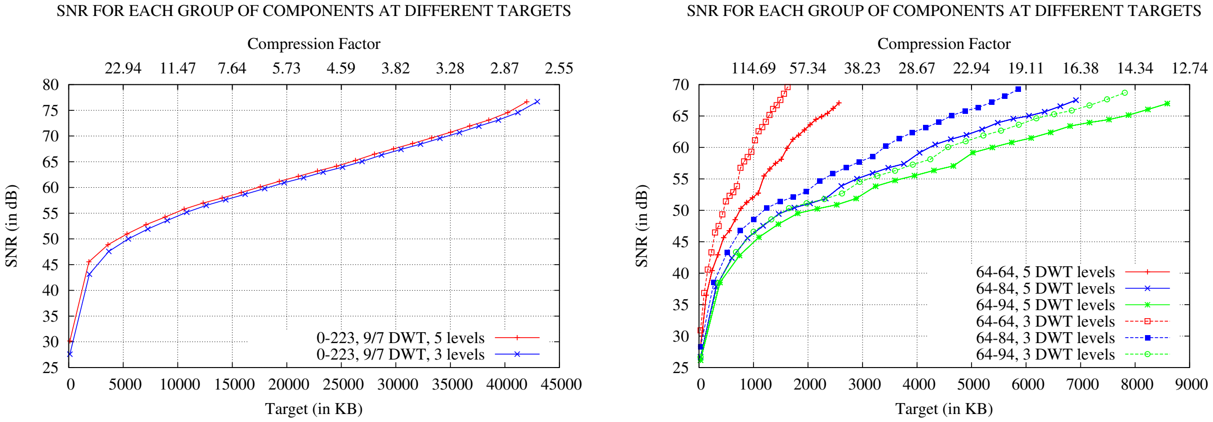


Figure 7: Coding performance comparison between the 9/7 irreversible and the 5/3 reversible filter applied across the spectral domain. Low Altitude radiance image.



(a) Transmission of the whole image (224 components).

(b) Transmission of a contiguous set of components. 5/3 reversible kernel.

Figure 8: Influence of the number of wavelet decomposition levels for spectral decorrelation. Cuprite radiance image.

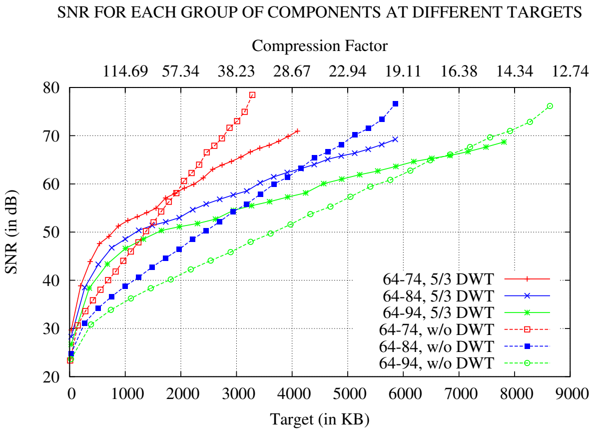


Figure 9: Performance comparison between two coding approaches: no transform across the spectral dimension versus an spectral wavelet transform approach (5/3 kernel with 3 decomposition levels). Cuprite radiance image.

According to the third and fourth experiments, and when only a group of contiguous components is requested, it seems that an spectral decorrelation performed by a 5/3 reversible filter with 3 decomposition levels wavelet transform provides a competitive coding performance. The fifth and last experiment compares, for the Cuprite radiance image, the coding performance of the two coding approaches, namely whether to apply or not a wavelet transform for spectral decorrelation. Figure 9 depicts the coding performance when the spectral wavelet transform is performed with the 5/3 kernel and with 3 decomposition levels. When transmitting ten contiguous components, and for very low bit rate, the quality of the recovered image is higher when an spectral transform has been applied; for medium to high bit rate, it is better not to apply an spectral transform. When transmitting thirty contiguous components, and for almost all bit rates, applying an spectral wavelet transform yields superior performance.

5. CONCLUSIONS

Previous studies have shown that the coding performance of hyperspectral image compression is improved when inter-band redundancy is exploited, for instance, by means of a Karhunen-Loêve Transform or a Discrete Wavelet Transform. Although the KLT helps to achieve a higher coding performance, because of its higher computational complexity and because of several attractive features of the DWT, the wavelet transform has been selected here. This paper deals with the interactive transmission of hyperspectral images that have been spectrally wavelet transformed.

When the whole hyperspectral image is to be recovered, applying such spectral transform proves to be beneficial. However, when hyperspectral images are transmitted, users may not need the whole image cube, but only a sub-set of components. If an spectral wavelet transform has been applied, in order to invert that transform, multiple components have to be delivered, even if they have not been requested by the user. The number of needed components to invert the DWT depends on the number of components requested, the wavelet kernel employed, and the number of wavelet decomposition levels. In an interactive transmission scenario, all these items should be carefully balanced.

Within the framework of JPEG2000 coding system, and in the setting of an interactive transmission scenario like JPEG2000 Internet Protocol (JPIP), we have investigated whether applying a wavelet transform as spectral decorrelator for hyperspectral image coding is appropriate. The influence of both the 9/7 irreversible and the 5/3 reversible kernels has been examined. The 9/7 filter yields a higher performance when the whole hyperspectral image is to be retrieved; however, when only a reduced sub-set of components is requested, the 5/3 filter is preferred. Similarly, applying a higher number of wavelet decomposition levels improves the coding performance when retrieving the whole image, but when requesting a reduced sub-set of components, it pays off to employ only a 3 level wavelet decomposition.

To summarize, the coding performance when hyperspectral images are interactively transmitted is a trade-off among the DWT filter, the number of DWT decomposition levels, and the number of contiguous requested components. In addition, it evolves along the bit rate.

This paper introduces also a JPIP implementation, CADI, which allows to evaluate and try new techniques, algorithms, and coding systems in the framework of the JPIP protocol. Testing is very easy due to its flexible design, where modules can be replaced or modified without affecting other modules. CADI's source code and documentation are under the General Public License (GPL), allowing its free use, modification, and distribution.

Acknowledgments

The authors would like to thank Fernando García-Vílchez for his help with the implementation.

This work has been partially supported by the Spanish and Catalan Governments, by FEDER, and by the Universitat Autònoma de Barcelona under Grants TSI2006-14005-C02, MEC_BPD-2007-1040, SGR2005-00319, and UAB2005-022.

REFERENCES

[1] Kerekes, J. P. and Scott, J. R., [*Hyperspectral Data Exploitation: Theory and Applications*], ch. "Hyperspectral Imaging Systems", 19–46, John Wiley & Sons Inc., Hoboken, NJ, USA (2007).

[2] Jet Propulsion Laboratory and NASA, "Airborne Visible InfraRed Imaging Spectrometer website." http://aviris.jpl.nasa.gov/html/aviris.overview.html.

[3] Gupta, S. and Gersho, A., "Feature predictive vector quantization of multispectral images," *IEEE Transactions on Geoscience and Remote Sensing* **30**(3), 491–501 (1992).

[4] Qian, S.-E., Hollinger, A., Williams, D., and Manak, D., "Vector quantization using spectral index-based multiple-subcodebooks for hyperspectral data compression," *IEEE Transactions on Geoscience and Remote Sensing* **38**(3), 1183–1190 (2000).

[5] Epstein, B., Hingorani, R., Shapiro, J., and Czigler, M., "Multispectral KLT-wavelet data compression for landsat thematicmapper images," 200–208, Proc. IEEE Data Compression Conference (1992).

[6] Saghri, J., Tescher, A., and Reagan, J., "Practical transform coding of multispectral imagery," *IEEE Signal Processing Magazine* **12**(1), 32–43 (1995).

[7] Abousleman, G., Marcellin, M., and Hunt, B., "Compression of hyperspectral imagery using the 3-D DCT and hybrid DPCM/DCT," *IEEE Transactions on Geoscience and Remote Sensing* **33**(1), 26–34 (1995).

[8] Fowler, J. E. and Rucker, J. T., [*Hyperspectral Data Exploitation: Theory and Applications*], ch. "3D wavelet-Based Compression of Hyperspectral Imagery", 379–407, John Wiley & Sons Inc., Hoboken, NJ, USA (2007).

[9] Penna, B., Tillo, T., Magli, E., and Olmo, G., "Embedded lossy to lossless compression of hyperspectral images using JPEG2000," *IEEE Transactions on Geoscience and Remote Sensing* **1**, 140–143 (2005).

[10] Tang, X. and Pearlman, W. A., [*Hyperspectral Data Compression*], ch. "Three-Dimensional Wavelet-Based Compression of hyperspectral Images", 273–308, Springer, USA (2006).

[11] Tang, X., Cho, S., and Pearlman, W., "3D set partitioning coding methods in hyperspectral image compression," **2**, 239–242, Proc. IEEE International Conference on Image Processing (2003).

[12] Kim, B.-J. and Pearlman, W., "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," 251–260, Proc. IEEE Data Compression Conference (1997).

[13] Kiely, A. and Klimesh, M., "The ICER Progressive Wavelet Image Compressor," tech. rep., IPN Progress Report (2003). http://ipnpr.jpl.nasa.gov/tmo/progress_report/42-155/155J.pdf.

[14] Klimesh, M., Kiely, A., Xie, H., and Aranki, N., "ICER-3D: A Progressive Wavelet-Based Compressor for Hyperspectral Images," tech. rep., IPN Progress Report (2006). http://ipnpr.jpl.nasa.gov/progress_report/42-164/164A.pdf.

[15] ISO/IEC and JTC1, "Digital compression and coding of continuous tone still images," (1991). Part 1: Tech. Rep. 10918-1; Part 2: Tech.Rep. 10918-2.

[16] CompuServe Incorporated, "Graphics interchange format (tm): A standard defining a mechanism for the storage and transmission of raster-based graphics information." http://www.w3.org/Graphics/GIF/spec-gif87.txt.

[17] I3A - The International Imaging Industry Association, "Flashpix file format." http://www.i3a.org/i_flashpix.html.

[18] I3A - The International Imaging Industry Association, "Internet imaging protocol." http://www.i3a.org/i_iip.html.

[19] "Lizardtech." http://www.lizardtech.com.

[20] "Earth resource mapper." http://www.ermapper.com.

[21] "JPEG 2000 image coding system - Part 1: Core coding system, Information technology 15444-1. International Organization for Standardization / International Electrotechnical Commission (ISO/IEC); International Telecommunication Union-Telecom Standardization (ITU-T)," (2000).

[22] "JPEG 2000 image coding system - Part 2: Extensions, Information technology 15444-2. International Organization for Standardization / International Electrotechnical Commission (ISO/IEC); International Telecommunication Union-Telecom Standardization (ITU-T)," (2004).

[23] "JPEG 2000 image coding system - Part 9: Interactivity tools, APIs, and protocols. Information technology 15444-9. International Organization for Sandardization / International Electrotechnical Commission (ISO/IEC); International Telecommunication Union-Telecom Standardization (ITU-T)," (2005).

[24] Du, Q. and Fowler, J. E., "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geoscience and Remote Sensing Letters* **4**(2), 201–205 (2007).

[25] Choi, E., Choi, H., and Lee, C., "Principal component analysis with pre-emphasis for compression of hyperspectral imagery," **2**, Proc. IEEE International Geoscience and Remote Sensing Symposium (2005).

[26] Serra-Sagrista, J. and Auli-Llinas, F., [*Computational Intelligence For Remote Sensing*], ch. "Remote Sensing Data Compression", Springer, USA (2008).

[27] Penna, B., Tillo, T., Magli, E., and Olmo, G., "Progressive 3-D coding of hyperspectral images based on JPEG2000," *IEEE Geoscience and Remote Sensing Letters* **3**(1), 125–129 (2006).

[28] Taubman, D. S., "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing* **9**(7), 1158–1170 (2000).

[29] Deshpande, S. and Zeng, W., "Scalable streaming of JPEG2000 images using hypertext transfer protocol," **9**, 372–381, International Multimedia Conference, ACM Press, New York, USA (2001).

[30] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T., "Hyper-Text Transfer Protocol – HTTP/1.1," tech. rep., World Wide Web Consortium and the Internet Engineering Task Force (1999). RFC 2616.

[31] Li, J. and Sun, H. H., "On interactive browsing of large images," *IEEE Transactions on Multimedia* **5**(4), 581–590 (2003).

[32] Taubman, D. and Prandolini, R., "Architecture, philosophy and performance of JPIP: Internet protocol standard for JPEG2000," **5150**, 791–805, Proc. SPIE Visual Communications and Image Processing (2003).

[33] Moshfeghi, M. and Ta, J., "Efficient image browsing with JPEG2000 internet protocol," **5371**, 31–42, Proc. of SPIE Medical Imaging 2004: PACS and Imaging Informatics (2004).

[34] Ortiz, J., Ruiz, V., and Garcia, I., "Improving web proxy caching on browsing JPEG2000 remote images with JPIP," **1**, 167–172, WSEAS Transactions on Information Science and Application (2004).

[35] Deshpande, S., "XML based JPIP client side cache model management," **5670**, 118–125, Proc. SPIE-IS&T (2005).

[36] Group on Interactive Coding of Images, "BOI software; Open Source JPEG2000 implementation." http://www.gici.uab.cat/BOI (October 2007).

[37] Group on Interactive Coding of Images, "CADI software; Open Source JPIP implementation." http://sourceforge.net/projects/CADI (June 2008).

Chapter 3

JPIP Proxy and Prefetching Strategies

3.1 Introduction

This chapter is composed of two conference papers and a journal paper that present three prefetching strategies to improve the interactive browsing of images over JPIP.

In a client-server framework, clients send requests and servers send back the responses. It is common in computer networks to deploy proxy servers, which are intermediate servers that clients use as a means to access remote servers. The main advantages of proxy servers are that they reduce the amount of bandwidth usage between the proxy and the server and that they reduce the response times. This is achieved by means of a proxy cache within the proxy. A proxy server with such a cache intercepts the requests from clients and sends the response to the clients if it is already present in the proxy's cache, or forwards the request to the server otherwise. Then, it updates its cache with the data received from the server. In such strategy, the user's perceived latency can be improved by increasing the caching hit ratio, which can be done anticipating the future data to be requested and fetching them in the cache.

The proposal of this research is to improve the browsing experience of the users by employing a JPIP proxy server with prefetching mechanisms. The proposed proxy pre-fetches image data from the server predicting future movements of the users. The main aspect when selecting a prefetching strategy is the ability to foresee what data need to be pre-fetched in order to reduce the responsiveness of the overall system. Three prefetching strategies are proposed:

1. The **simple prefetching** strategy employs a uniform model of probabilities to predict the next user movements. It was introducted in [53] and the results show an improvement of the transmission time and browsing experience when compared to a conventional JPIP proxy server without prefetching.

2. The prefetching strategy that uses a **user-navigation model** [54] improves the results of the previous strategy. The proposed prediction model considers the most likely future movements according to a general behavior extracted from user sessions. This analysis is carried out off-line and then the probabilities are applied to the model. Experimental results suggest that this more accurate model yields a notable improvement of the responsiveness of the system and a better navigating experience for the user.

3. The third approach is the prefetching based on the **semantic-map** [55]. The main idea behind it is to consider the content of the image as an indicator of the probability of an area to be visited. Images are segmented into regions with different priorities that are based on the interpretation of the semantic information extracted from the image. The process to obtain the priority map is carried out off-line as a previous pre-processing. Results indicate that this transmission strategy enhances the responsiveness of the overall system as compared to a conventional proxy.

3.2 JPIP Proxy Server with Prefetching

```
@inproceedings {MASB10,
      AUTHOR = {J. Lino Monteagudo-Pereira and Francesc Auli-Llinas and Joan Serra-Sagrista
                and Joan Bartrina-Rapesta},
       TITLE = {Smart JPIP Proxy Server with Prefetching Strategies},
   BOOKTITLE = {IEEE Data Compression Conference 2010 (DCC 2010)},
   PUBLISHER = {IEEE Press},
       MONTH = {Mar.},
        YEAR = {2010},
       PAGES = {99-108},
        ISSN = {1068-0314},
         DOI = {10.1109/DCC.2010.16},
}
```

Smart JPIP Proxy Server with Prefetching Strategies

José Lino Monteagudo-Pereira, Francesc Aulí-Llinàs,
Joan Serra-Sagristà and Joan Bartrina-Rapesta

Department of Information and Communications Engineering
Universitat Autònoma de Barcelona, Spain

Abstract

Remote browsing of images is receiving much attention lately, mostly in niche applications like geographical information systems or in the telemedicine scenery. Interactive transmission of compressed images has been identified as the most competitive approach, being JPIP, JPEG2000 Interactive Protocol, a key-enabler for these situations. Also, it has been reported that JPIP Proxy Servers help to increase the transmission performance, and that prefetching strategies help to lower responsiveness time. In this paper we contribute a smart JPIP Proxy Server that, thanks to a prefetching strategy undertaken during idle transmission times, largely improves the viewing experience of the final user because of its anticipation of future navigation requests. Experimental results are reported for a remote sensing and a medical environment, respectively performing panning and zoom in, showing enhanced performance in both cases.

I. INTRODUCTION

JPEG2000 is a prominent standard for the compression, transmission, security, and manipulation of images and video. Part 1 of the standard [1] defines the core coding system, which is wavelet-based with a two tiered coding strategy. Among other features, the JPEG2000 coding scheme supplies scalability by spatial location, by quality, by resolution, and by component. This high degree of scalability is combined with a rich codestream syntax that allows random access to the file. The capability to work in the codestream domain, *i.e.*, to identify –and potentially transmit–, any image portion without needing to decode any part of the codestream, makes JPEG2000 one of the most suitable coding standards for image transmission.

The potential of JPEG2000 to interactively share images and video was explored soon after the publication of Part 1 [2]–[5]. In November 2005, JPEG2000 Part 9 [6], JPEG2000 Interactive Protocol (JPIP), was published. It is a versatile client-server protocol for the interchange of imagery among different implementations. JPIP facilitates a wide range of applications, such as remote browsing of high resolution images, video-on-demand scenarios, or surveillance systems, among others.

When applications have to deliver images interactively, the fundamental asset of JPIP is the rapid recovery of the areas requested by the client. This issue has been extensively discussed in the literature and several mechanisms have been proposed. First, the server

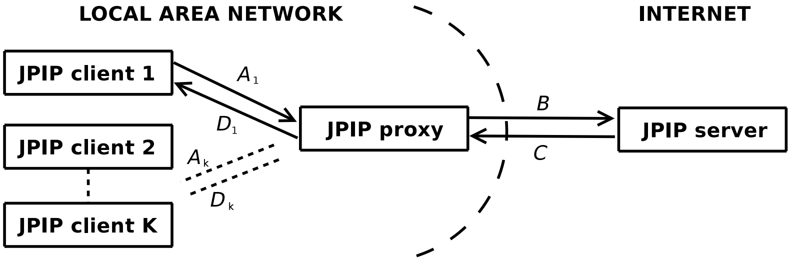


Fig. 1. The framework envisioned in this work considers a LAN with several JPIP clients browsing images from a JPIP server through Internet.

must consider the re-sequencing of the original codestream to maximize the quality of the received images [7]. Second, efficiency is also improved when image data is compressed by way of proper coding parameters [8]. Third, prefetching strategies at the client side enhance the browsing experience due to a faster visualization of the requested areas [9]. Last but not least, the deployment of proxy servers at different points of a local network, or distributed over the Internet, reduces both the load of the server and the client's response time [10], [11].

Commonly, all clients in a center share the same Internet connection, which may have a bandwidth ranging from the most modest ADSL lines at 256 Kbps to the fast T3 connections at up to 45 Mbps. Whichever is the Internet connection, the bandwidth of the local area network (LAN) is generally several times higher than that of the Internet. Therefore, to improve the efficiency of JPIP clients browsing the same image or video sequence, it is beneficial for such centers to feature some of the above mechanisms. In such scenery, the anticipation of user movements [9] might help to fully exploit the network bandwidth; however, the application of prefetching strategies simultaneously to all clients might saturate the Internet connection, reducing the overall system's performance. On the other hand, the deployment of a proxy server like [11] is particularly suitable, since it caches all information retrieved by clients, avoiding the transmission of the same information more than once. Nonetheless, typical proxy servers leave the Internet connection idle when users have retrieved the requested area.

The framework envisioned in this work considers several JPIP clients within a LAN that share the same connection and which request data from a remote JPIP server through Internet (see Figure 1). Our purpose is to evaluate prefetching strategies applied to a JPIP proxy server, rather than to each client, to enhance the browsing experience for clients within the LAN. The proposed JPIP proxy server employs a rate-distortion optimized algorithm that maximizes the responsiveness of the whole system taking into account the current and (possibly) future regions requested by clients. Prefetching is carried out only by the proxy server when the Internet connection is idle. Experimental results report significant gains with respect to typical JPIP proxy servers without prefetching strategies.

This paper is structured as follows: Section II serves as a short overview of the JPIP protocol; Section III introduces the algorithm employed in our JPIP proxy server;

Section IV assesses the performance of the proposed method in a remote sensing and a medical scenario; and last section summarizes this work.

II. JPEG2000 INTERACTIVE PROTOCOL

The JPEG2000 core coding system is constituted by four main coding stages: sample data transformation, sample data coding (tier-1), rate-distortion optimization, and codestream re-organization (tier-2). The first stage applies a wavelet transform that compacts the image energy and decomposes the image in successive levels of resolution. After the wavelet transform and quantization, the image is conceptually partitioned in small sets of wavelet coefficients, called codeblocks, that are coded by the bitplane coding engine in the tier-1 coding stage. Tier-1 produces a quality embedded bitstream for each codeblock that can be truncated at the several points. Let d_n, l_n denote, respectively, the distortion and length achieved at truncation points for a codeblock of the image. The subsequent stage commonly employs Lagrange optimization to minimize the image distortion given a target bitrate for the final codestream, or to construct quality layers. First, the convex hull of individual codeblocks is established by identifying those truncation points with strictly decreasing distortion-length slope s_n, with $s_n = \frac{d_{n-1} - d_n}{l_n - l_{n-1}}$. Then, quality layers are formed selecting bitstream segments from the union of all codeblocks with the highest distortion-length slopes. Through this process, each layer contains bitstream segments with equal or higher distortion-length slope than the threshold achieved for that layer, which is referred to as S_q with $1 \leq q \leq Q$, Q denoting the number of layers of the codestream. The last stage of the coding pipeline is the tier-2, which codes auxiliary information to allow the identification of quality layers within the final codestream.

As Figure 2 depicts, the JPEG2000 codestream is structured in containers. The smallest one is the *packet*, which encapsulates data from codeblocks belonging to the same component, resolution level, and spatial area of the image, the so-called *precinct*. Data produced for each precinct are distributed in as many packets (some of them possibly null) as quality layers has the final codestream. When the image is interactively transmitted, JPIP servers generally dismantle the codestream into a collection of *data-bins* that contain all packets belonging to one precinct. When the client requests an image region, called window of interest (WOI) from now onward, the server identifies those data-bins that contain encoded data of that WOI, and transmits segments of these data-bins to the client. Rather than to interact directly with the codestream, JPIP defines a versatile dialog between client and server in which the client requests a WOI specifying spatial location, resolution, and components on the image domain, and the server identifies that WOI in the compressed domain and transmits back selected segments of the corresponding data-bins. Messages containing one or several segments of data-bins are referred to as *JPP-streams*[1].

---

[1] In the discussion above, and in the rest of this paper, we assume that the image is not partitioned in tiles [12, Ch. 11.2], since interactive transmission is commonly not benefited from such partitions [3].

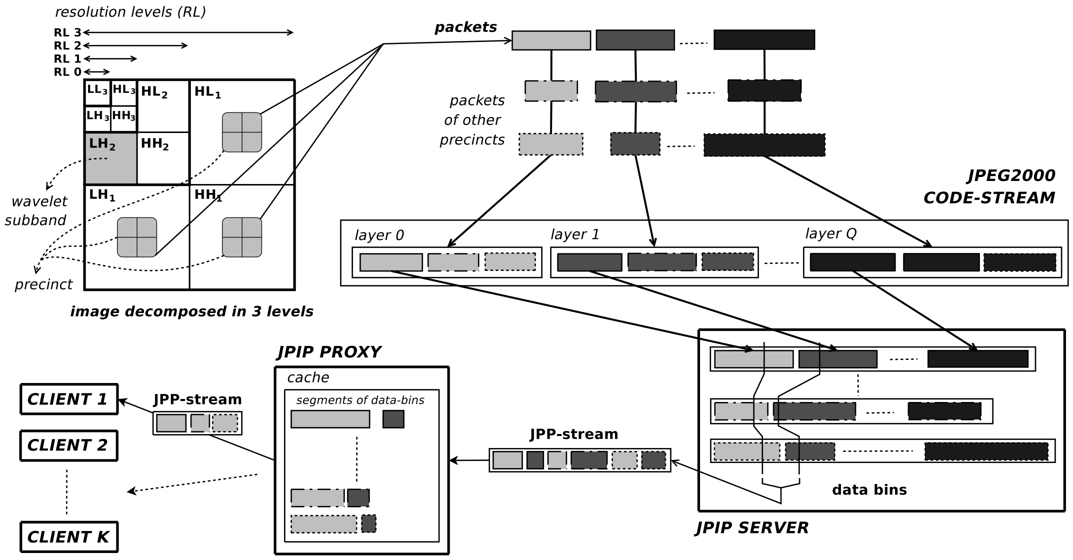


Fig. 2. Simplified overview of the JPEG2000 partitioning system, the codestream organization, and the data transmitted by the JPIP server and proxy.

An important issue that JPIP servers contemplate is the delivery of data in an optimized fashion: to be more precise, let us denote packets of the codestream as $\mathcal{T}_{c,r,s,q}$, where subindexes c, r, s, q identify the component, resolution level, spatial location, and quality layer of the packet; when the client requests a WOI, say $\mathcal{W}$, the server determines those $\mathcal{T}_{c,r,s,q}$ related to $\mathcal{W}$, and determines adequate rates for JPP-streams transmitted to the client. With an abuse of notation, we write $\mathcal{T}_{c,r,s,q} \in \mathcal{W}$ when $\mathcal{T}_{c,r,s,q}$ are related to $\mathcal{W}$.

To minimize the distortion of transmitted WOIs, the server commonly selects packets according to the the layer's order, *i.e.*, packets in first layers are transmitted first. Since $\mathcal{W}$ does not typically match exactly with image precincts, [7] suggested the application of a window scaling factor $F_{c,r,s}$ that accounts for the percentage of coefficients within that precinct that are relevant to $\mathcal{W}$. Through $F_{c,r,s}$, data is re-sequenced and transmitted to the client according to artificial distortion-length slopes computed for individual packets as $S'_{c,r,s,q} = S_q \cdot F_{c,r,s}$. This re-sequenced packet ordering does not assure the minimization of distortion per transmitted unit, since S_q is only an approximation of the real distortion-length slope for packet $\mathcal{T}_{c,r,s,q}$, however, such strategy is near-optimal in practice [7], [9]. As common, we assume that values for S_q are recorded in the codestream.

All transmitted data is cached by the client. When the server maintains a model of that cache, the connection is named stateful; otherwise, it is named stateless.

III. SMART JPIP PROXY SERVER

A. Preliminaries

The deployment of proxy servers is a common practice to improve the network infrastructure of LANs and the Internet. Currently, most proxy servers are based on the

HTTP protocol to cache the information contained in web sites. The main difficulty to take advantage of the already deployed HTTP proxy infrastructure is that the rich dialog between JPIP clients and servers is not understood by HTTP proxy servers. The reason that for is because petitions carried out by JPIP clients are understood by proxies as URLs, thus, only exactly the same requests are matched. This issue is studied in [10], where a modification to JPIP to take advantage of HTTP proxy servers is proposed.

To maintain the full set of JPIP features and keep compliance with the standard, another option is to introduce a specialized JPIP proxy server, able to understand the JPIP protocol. The bottom line approach is to dismantle JPP-streams in the proxy, and cache data as a collection of data-bins (see Figure 2). To replace the server efficiently, the proxy must know how to associate packets $\mathcal{T}_{c,r,s,q}$ to the image domain, and also to maintain a model of the clients' cache to allow stateful connections.

The typical functions carried out by a JPIP proxy server (see Figure 1) are: first, a client k within the LAN requests a WOI (link A_k) to the JPIP proxy server, which lays also within the LAN; once the proxy receives the request, it checks whether that WOI is contained in cached data or not, in case the WOI is *totally* contained in cache, the proxy responses to the client transmitting JPP-streams (link D_k), in case the WOI is *partially* contained in cache, the proxy transmits cached data to the client and, simultaneously, re-formulates the petition to the server requesting non-cached data (connection $\{B,C\}$). When there are no data-bins belonging to the requested WOI in cache, the proxy sends the petition to the server and caches transmitted data. This model of JPIP proxy server, jointly with an algorithm to re-formulate client's requests, is proposed in [11], significantly improving the interactive transmission of multiple clients browsing the same image.

However, the JPIP proxy server in [11] only considers the *current* WOIs requested by users. It has been shown [9] that user-navigation models can enhance the responsiveness of the application by anticipating the request of future WOIs. The main incentive behind user-navigation models is that there exists a delay between the emission of data by the server and the feedback that it causes to the user in terms of WOI definition, during which potential future WOIs can be requested. This delay includes the transmission time, decoding latency, and the time the user needs to interpret newly received visual information to decide next navigation commands. Nevertheless, prefetching strategies should not be directly applied as they are formulated in [9] to all clients within a LAN. This is justified since the joint transmission of WOIs that are in current use with WOIs that *might* be possibly required in the future, could cause that the available bandwidth is assigned to useless queries, penalizing the overall system performance.

B. Rate-distortion optimized transmission

To overcome the aforementioned limitations, the smart JPIP proxy server proposed in this paper introduces a strategy that combines techniques of rate-distortion optimization and a model of probabilities derived from user-navigation models. The union of these two technologies in the proxy provides prefetching to all clients within the LAN simultaneously, maximizing the overall system's performance.

Our JPIP proxy server has two modes of operation: normal and prefetching. The *normal mode* is active when at least one client requests WOIs that are not fully contained in the proxy's cache. The *prefetching mode* is active when all client's petitions can be responded with the proxy's cache, and connection with the server ($\{B,C\}$ in Figure 1) is idle.

The first insight employed by the JPIP proxy server is the re-formulation of WOIs requested to the server. Rather than to re-formulate the WOI including several areas demanded by clients [11], our proxy maximizes the overall system's performance requesting specific WOIs one by one through the use of JPIP parameters (see below). Let us extend the notation of the previous section to denote the currently requested WOIs by clients as $\mathcal{W}_k$, with $1 \leq k \leq K$, K being the number of active clients.

In *normal mode*, the proxy computes artificial distortion-length slopes $S''_{c,r,s,q}$ for packets considering all clients browsing the same image according to

$$S''_{c,r,s,q} = S_q \cdot F_{c,r,s} \cdot \frac{1}{K} \sum_k I(\mathcal{T}_{c,r,s,q}) \text{ , with } I(\mathcal{T}_{c,r,s,q}) = \begin{cases} 1 & \text{if } \mathcal{T}_{c,r,s,q} \in \mathcal{W}_k \\ 0 & \text{otherwise} \end{cases} .$$

The proxy identifies those $\mathcal{T}_{c,r,s,q}$ with highest values of $S''_{c,r,s,q}$, and requests the corresponding WOI to the server. Through this strategy, the overall system's performance is maximized since the highest $S''_{c,r,s,q}$ determines the packet that minimizes the clients' distortion per transmitted unit in link C. Once the packet arrives to the proxy, it is transmitted to the clients using artificial slopes $S'_{c,r,s,q}$ to form JPP-streams, thus maximizing the performance in D_k links.

In *prefetching mode*, the JPIP proxy server requests potential future WOIs. The next *actual* WOI requested by clients is denoted as $\widetilde{\mathcal{W}_k}$ and is defined as a function of possible user movements, given the current $\mathcal{W}_k$, as $\widetilde{\mathcal{W}_k} = f(\mathcal{W}_k, X)$, where X stands for the user movements, *i.e.*, $X = \{$up, down, left, right, upLeft, upRight, downLeft, downRight, zoomIn, zoomOut$\}$. We assume that the probability of future movements is equally distributed as a first approximation and simplicity, though the proposed framework might also include other strategies like those proposed in [9] in order to achieve a better assumption of the client's movements. When the server is in prefetching mode, the proxy requests data using the artificial distortion-length slopes $S'''_{c,r,s,q}$ defined as

$$S'''_{c,r,s,q} = S_q \cdot F_{c,r,s} \cdot \frac{1}{K} \sum_k P(\mathcal{T}_{c,r,s,q}), \text{ with } P(\mathcal{T}_{c,r,s,q}) = \sum_X \begin{cases} \frac{1}{10} & \text{if } \mathcal{T}_{c,r,s,q} \in f(\mathcal{W}_k, X) \\ 0 & \text{otherwise} \end{cases} .$$

Again, the proxy requests that WOI corresponding to the packet with highest $S'''_{c,r,s,q}$, and caches the response hoping that next user movements will request that area.

To summarize, delivery of data in links D_k is conducted through $S'_{c,r,s,q}$, and WOIs are requested in link B through $S''_{c,r,s,q}$ when the proxy is in normal mode, and through $S'''_{c,r,s,q}$ when the proxy is in prefetching mode. Again, strict optimal performance is not guaranteed due to the use of artificial distortion-length slope for packets.

C. Implementation considerations

Several considerations must be taken into account to implement the proposed JPIP proxy server:

- The connection $\{B,C\}$ between proxy and server must use JPIP options ALIGN and AUX. ALIGN forces the JPIP server to deliver non-segmented packets, which is required by the proxy to identify individual packets $\mathcal{T}_{c,r,s,q}$ without needing to decode data. AUX forces the JPIP server to include a field that specifies the layers to which packets belong to. These options are not required in links A_k, D_k.
- The JPIP proxy server requests WOIs (link B) specifying the number of layers through the JPIP option LAYERS. This is mandatory to restrict the server to transmit a specific number of layers for the requested precinct. Note that this blocks any rate control mechanism used in the server to transmit data, intentionally leaving the control to the proxy. Although this causes requests of multiple WOIs in a short period of time, our experience indicates that this strategy works efficiently in practice. To force the server to complete the transmission of requested WOIs, the proxy must include the parameter WAIT in requests. This assures that the delivery of packets is carried out as intended by the proxy, and that transmissions are not interrupted as more requests arrive to the server.
- We assume that values for S_q are available by the proxy through their transmission in COM markers of the codestream, or through rough estimations as described in [13].
- It is worth noting that the server sees the proxy as a single client, thus the use of a stateful connection is highly recommended in connection $\{B,C\}$. Connections between the proxy and clients can be stateless.
- As a final remark, it is recommended that each connection is conducted by a different execution thread in the proxy, such that requests and responses can be treated simultaneously.

IV. EXPERIMENTAL RESULTS

All the infrastructure used to evaluate the proposed method, including the client, the server, and the JPIP proxy server are implemented in our JPEG2000 Part 9 implementation CADI [2]. Experiments are carried out in a LAN with a network bandwidth of 100 Mbps, and a 10 Mbps connection to the Internet shared by all clients. Images used in these experiments belong to the remote sensing and the medical communities, namely, a Landsat image of size 4096×4096 (8-bit gray-scale), and a Computed Radiography (CR) of size 2048×2495 (8-bit gray-scale). Coding parameters are: 5 levels of 5/3 IWT, codeblock size of 32×32, and restart coding variation. The server adjusts on-the-fly the precinct size in all resolution levels according to the codeblock size to enhance interactivity options.

First we evaluate the results achieved when two clients browse the Landsat image, selecting their WOIs through panning movements. Our browsing application allows panning to any direction, and also changes on size of the displaying window. Table I (top)

---

[2] http://www.gici.uab.cat/CADI

TABLE I
User navigation commands launched by clients 1 and 2 during the browsing session. WOIs are specified as {x origin, y origin, width, height, layers, resolution level}.

REMOTE SENSING IMAGE	
time (in secs)	*event*
0	Client 1 requests WOI {512, 2048, 768, 704, 10, 5}
2	Client 2 requests WOI {2560, 1472, 448, 448, 25, 5}
18	Client 1 requests WOI {1088, 1856, 1088, 704, 34, 5}
21	Client 2 requests WOI {1408, 1536, 704, 896, 34, 5}

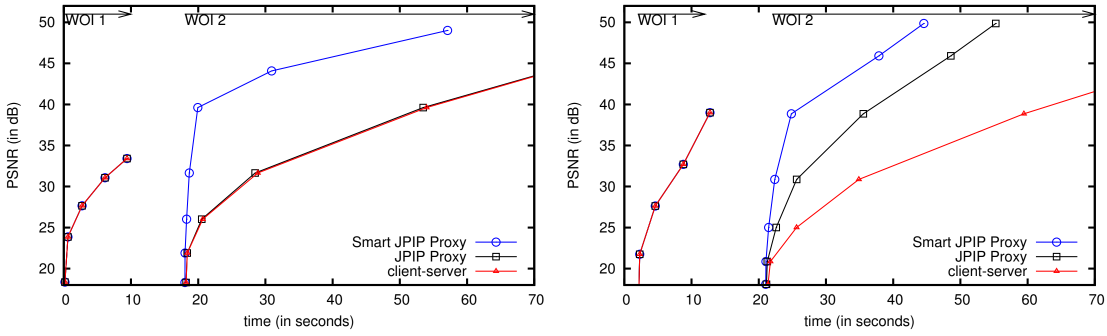
MEDICAL IMAGE	
time (in secs)	*event*
0	Client 1 requests WOI {144, 128, 288, 128, 10, 3}
5	Client 1 requests WOI {288, 256, 256, 288, 25, 4}
50	Client 1 requests WOI {576, 512, 512, 576, 34, 5}

relates the user navigation commands of a representative browsing session. Recall that our JPIP proxy server activates the prefetching mode when requested windows are all completely transmitted, *i.e.*, from second 13 to 18. Results achieved by the proposed JPIP proxy server with prefetching strategies (labeled as "Smart JPIP Proxy" in the figures) are compared to results achieved by a JPIP proxy server without prefetching (labeled as "JPIP Proxy"), which is similar to the FIFO proxy server introduced in [11], and to the results achieved when clients are directly connected to the server (labeled as "client-server").

Figure 3(a) depicts the results achieved by Client 1. The first WOI is retrieved by the three strategies with the same performance, since there is no data in the proxy's cache. The second WOI requested by Client 1 corresponds to a spatial area between the first WOI requested by Client 1 and the first WOI requested by Client 2, thus, that area is partially cached by the Smart JPIP Proxy during prefetching mode, since it has the highest probability in the proposed user-navigation model. This allows the Smart JPIP Proxy to transmit cached data to the client very rapidly while it requests more data to the server. The high increment in quality achieved from second 18 to 20 corresponds to cached data, whereas increments from second 20 onwards correspond to data transmitted at the rate imposed by the Internet connection. A typical JPIP Proxy, on the other hand, does not benefit from the idle time, and the performance when retrieving the second WOI is the same as that achieved for a client-server connection.

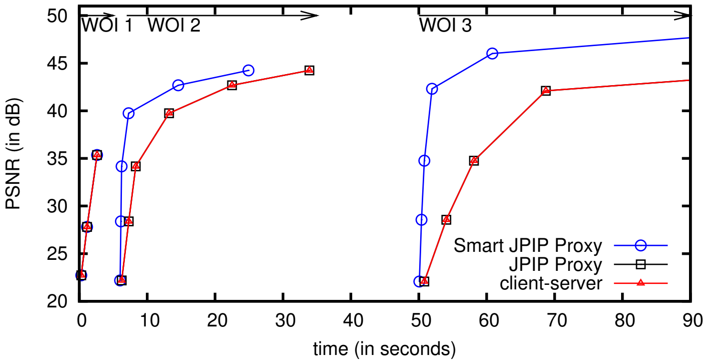
Figure 3(b) depicts the results achieved by Client 2. Similarly, the first requested WOI is retrieved by the three strategies with the same performance, since there is no data for that WOI in the cache of the proxy. The second WOI requested by the client is spatially close to the second WOI requested by Client 1. Therefore, a typical JPIP Proxy server enhances the performance with respect to the client-server strategy since data retrieved for Client 1 can be reused for Client 2. As above, the Smart JPIP Proxy yields the best performance since it cached more data belonging to that WOI during prefetching mode.

Our second experiment evaluates the performance of the proposed proxy when a single client browses the CR image carrying out three consecutive zoom in. This experiment considers a single client to illustrate the gains that can be achieved by the Smart JPIP



(a) Client 1 (b) Client 2

Fig. 3. Evaluation of the performance achieved by the proposed JPIP proxy server, a typical JPIP proxy server, and client-server connections when browsing a Landsat image.



(a) Client 1

(b) Smart JPIP Proxy

(c) JPIP Proxy

Fig. 4. (a) Evaluation of the performance achieved by the proposed JPIP proxy server, a typical JPIP proxy server, and client-server connections when browsing a CT image. (b) WOI recovered at second 51 by the proposed JPIP proxy server. (c) WOI recovered at second 51 by a typical JPIP proxy server.

proxy even when only one client is active. Table I (bottom) relates the WOIs requested by the client. Our JPIP proxy server activates the prefetching mode from second 2.6 to 6, and from second 25 to 50. Figure 4(a) depicts the results achieved by the three aforementioned strategies. The performance achieved by a typical JPIP proxy and the client directly connected to the server is the same since the proxy begins with no data in the cache. Contrarily, the Smart JPIP Proxy provides enhanced performance because it cached potential future WOIs during idle time. For the third WOI, for example, the Smart JPIP Proxy needs less than 2 seconds to achieve a PSNR of 44 dB, whereas the remaining strategies require more than 28 seconds. Figures 4(b), and 4(c) respectively provide the recovered WOIs at second 51 by the typical and the Smart JPIP Proxy servers. Similar results are obtained when several clients browse these and other images, suggesting that the proposed JPIP proxy server significantly improves the browsing experience for the end-user.

V. CONCLUSIONS

This paper considers remote interactive browsing of large JPEG200 images through JPIP, JPEG2000 Interactive Protocol, when a collection of clients in a given local area network share the same Internet connection and request windows of interest (WOIs) from a given remote server. For this framework, JPIP Proxy Servers have been proposed as a means to reduce the transmission time, and prefetching strategies have been proposed to decrease the responsiveness time.

In our work, a smart JPIP Proxy Server is presented. It has two operational modes, a *normal mode* which serves local requests when it has them in cache and which asks for requests to the remote server otherwise, and a *prefetching mode* that benefits from prefetching during idle transmission times, anticipating future navigation requests and sending petitions for potential WOIs to the remote server. Experimental results suggest that the *prefetching mode* yields significantly better transmission time and viewing experience, both for panning and zoom in movements, as compared to a typical JPIP Proxy Server, and even better when compared to a plain client-server scenario.

ACKNOWLEDGMENT

This work has been partially supported by Universitat Autonoma de Barcelona, Spanish Government, FEDER, and Catalan Government, under Grants UAB-BI3INT2006-08, 2008-BPB-0010, TIN2009-14426-C02-01, TIN2009-05737-E/TIN, and 2009-SGR-1224.

REFERENCES

[1] *Information technology - JPEG 2000 image coding system - Part 1: Core coding system*, ISO/IEC Std. 15 444-1, Dec. 2000.

[2] S. Deshpande and W. Zeng, “Scalable streaming of JPEG2000 images using hypertext transfer protocol,” in *Proc. ACM International Conference on Multimedia*, vol. 9, Oct. 2001, pp. 372–381.

[3] D. Taubman, “Remote browsing of JPEG2000 images,” in *Proc. IEEE International Conference on Image Processing*, Sep. 2002, pp. 229–232.

[4] J. Li and H.-H. Sun, “On interactive browsing of large images,” *IEEE Trans. Multimedia*, vol. 5, no. 4, pp. 581–590, Dec. 2003.

[5] E. A. Politou, G. P. Pavlidis, and C. Chamzas, “JPEG2000 and dissemination of cultural heritage over the internet,” *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 293–301, Mar. 2004.

[6] *Information technology - JPEG 2000 image coding system: Interactivity tools, APIs and protocols*, ISO/IEC Std. 15 444-9, Dec. 2005.

[7] D. Taubman and R. Rosenbaum, “Rate-distortion optimized interactive browsing of JPEG2000 images,” in *Proc. IEEE International Conference on Image Processing*, vol. 3, Sep. 2003, pp. 765–768.

[8] K. Krishnan, M. W. Marcellin, A. Bilgin, and M. S. Nadar, “Efficient transmission of compressed data for remote volume visualization,” *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1189–1199, Sep. 2006.

[9] A. Descampe, C. D. Vleeschouwer, M. Iregui, B. Macq, and F. Marques, “Prefetching and caching strategies for remote and interactive browsing of JPEG2000 images,” *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1339–1354, May 2007.

[10] J. P. G. Ortiz, V. G. Ruiz, M. F. Lopez, and I. Garcia, “Interactive transmission of JPEG2000 images using web proxy caching,” *IEEE Trans. Multimedia*, vol. 10, no. 4, pp. 629–636, Jun. 2008.

[11] L. Lima, D. Taubman, and R. Leonardi, “JPIP proxy server for remote browsing of JPEG2000 images,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, Oct. 2008, pp. 844–849.

[12] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image compression fundamentals, standards and practice*. Norwell, Massachusetts 02061 USA: Kluwer Academic Publishers, 2002.

[13] D. S. Taubman, “Localized distortion estimation from already compressed JPEG2000 images,” in *Proc. IEEE International Conference on Image Processing*, Oct. 2006, pp. 3089–3092.

3.3 Enhanced Prefetching Strategy

```
@inproceedings {MASZMP12,
      AUTHOR = {J. Lino Monteagudo-Pereira and Francesc Auli-Llinas and Joan Serra-Sagrista
               and Alaitz Zabala and Joan Maso and Xavier Pons},
       TITLE = {Enhanced Transmission of JPEG2000 Imagery through JPIP Proxy and User-Navigation Model},
   BOOKTITLE = {IEEE Data Compression Conference 2012 (DCC 2012)},
   PUBLISHER = {IEEE Press},
       MONTH = {Apr.},
        YEAR = {2012},
       PAGES = {22-31},
        ISSN = {1068-0314},
         DOI = {10.1109/DCC.2012.10},
}
```

Enhanced transmission of JPEG2000 imagery through JPIP proxy and user-navigation model

J. Lino Monteagudo-Pereira[†], Francesc Aulí-Llinàs[†], Joan Serra-Sagristà[†],
Alaitz Zabala[‡], Joan Masó[⋆], and Xavier Pons[‡]

[†]Department of Information and Communications Engineering
[‡]Department of Geography
[⋆]Center for Ecological Research and Forestry Applications (CREAF)

Universitat Autònoma de Barcelona, Barcelona, Spain

Abstract

The efficient transmission of large resolution images is a key aspect in many applications to minimize the transmission costs and to enhance the browsing experience. Among the currently available standards for the coding and transmission of imagery, JPEG2000 excels for its superior coding performance and advanced capabilities. The JPEG2000 Interactive Protocol (JPIP) minimizes the amount of information transmitted in a client-server scenario. Nonetheless, JPIP does not provide mechanisms to re-use data already delivered to clients browsing the same image within a local network. Common HTTP proxy servers are not able to understand the syntax of JPIP, thus specialized JPIP proxy servers are put in practice. This work improves the capabilities of traditional JPIP proxy servers by means of a user-navigation model that, together with prefetching strategies, allows the server to anticipate (potential) future requests of clients. Experimental evidence indicates that the introduction of the navigational model into a JPIP proxy server enhances the browsing experience notably.

I. INTRODUCTION

The transmission of large resolution images is a requirement of many applications that are related to fields like territory management, oil spill control, or natural disaster monitoring, among others. GEO-PICTURES [1] is an European-funded project whose aim is to develop an ubiquitous tool that aids in situations of emergency such as natural disasters or civil defense. Among its multi-purpose functionalities, a key-piece of the GEO-PICTURES suite of applications is the storage and distribution of earth observation images to control centers and rescue teams working in the field. Important to the application is that only low or limited Internet bandwidth may be available during emergency situations. Therefore, the image transmission infrastructure must make the best use of the available communication links to provide a pleasant experience to the end-users browsing the images.

Among the currently available standards to code and transmit images, JPEG2000 excels for its superior coding efficiency and transmission capabilities. Part 1 of the standard [2] defines the core coding system, which is wavelet-based with a two tiered coding strategy. Fundamental features of the JPEG2000 coding scheme are scalability by spatial location, by quality, by resolution, and by component. This high degree of scalability is partially provided by a rich codestream syntax that allows random access to the file. The ability to work in the codestream domain, *i.e.*, to identify –and potentially transmit–, any image portion without needing to decode any part of the codestream, makes JPEG2000 one of the most suitable coding standards for interactive image transmission.

The potential of JPEG2000 to interactively share images and video was explored soon after the publication of Part 1 [3]–[6]. In November 2005, JPEG2000 Part 9 [7], JPEG2000 Interactive Protocol (JPIP), was published providing a versatile client-server protocol for the interchange of imagery among different implementations. When applications have to deliver images interactively, the fundamental asset of JPIP is the rapid recovery of the areas requested by the client. This issue has been extensively discussed in the literature and several JPIP-compliant techniques to achieve maximum efficiency have been proposed. First, the server must consider the re-sequencing of the original codestream to maximize the quality of the received images [8]. Second, efficiency is also improved when image data is compressed by way of proper coding parameters [9]. Third, prefetching strategies at the client side enhance the browsing experience due to a faster visualization of the requested areas [10]. Last but not least, the deployment of proxy servers at different points of a local network, or distributed over the Internet, reduces both the load of the server and the client's response time [11], [12].

Commonly, all clients in a center (or in a camp in the context of GEO-PICTURES) share the same Internet connection. The communication channel employed in each occasion is different depending on the situation, ranging from satellite links using BGAN terminals at 32 kilobits per second (kbps), wireless terrestrial 3G networks at 384 kbps, or wired terrestrial ADSL lines at up to 10 megabits per second (Mbps). Whichever is the Internet connection, the bandwidth of the local area network (LAN) is generally much higher than that of the Internet. Therefore, JPIP clients browsing an image can benefit from the mechanisms above to enhance the experience of the users.

The purpose of this work is to provide efficient mechanisms for the transmission of earth observation imagery in the context of GEO-PICTURES. JPEG2000 and JPIP are respectively adopted as the compression standard and the transmission protocol to interactively transmit images. This paper describes the research carried out to enhance the browsing experience of multiple users sharing the same Internet link in a camp, which is a common situation that occurs during emergency missions. The research presented herein lies on the top of our previous work [13], which introduced a JPIP proxy server that caches all information transmitted to JPIP clients within a LAN. The main contribution of that work is the deployment of a simple prefetching strategy in the JPIP proxy server that utilizes idle moments of the Internet connection to retrieve more data from the server. Prefetching is carried out with*out* considering the most likely movements that may be requested by the users. The current work continues that research by establishing a user-navigation model that predicts with more exactitude the future movements of the users. This model is introduced in the JPIP proxy server to aid the prefetching of image data, resulting in enhanced responsiveness for the overall system.

The paper is structured as follows. Section II serves as a short overview of the JPIP protocol. Section III briefly reviews the algorithm employed in our JPIP proxy server, and introduces the user-navigation model. Section IV assesses the performance of the proposed method in a GEO-PICTURES realistic scenario, and the last section concludes this work.

II. JPEG2000 INTERACTIVE PROTOCOL

The JPEG2000 core coding system is constituted by four main coding stages: sample data transformation, sample data coding (tier-1), rate-distortion optimization, and codestream re-organization (tier-2). The first stage applies a wavelet transform that compacts the image energy and decomposes the image in successive levels of resolution. After the wavelet transform and quantization, the image is conceptually partitioned in small sets of wavelet coefficients, called codeblocks, that are coded by the bitplane coding engine in tier-1. Tier-1 produces a quality embedded bitstream for each codeblock that can be truncated at different points. Let d_n, and l_n denote, respectively, the distortion and the length achieved at truncation points for a codeblock of the image. The subsequent stage commonly employs Lagrange optimization to form progressive layers of quality. Each quality layer gathers some bitstream segments of codeblocks minimizing the image distortion without exceeding a bitrate. To do so, the Lagrange optimization procedure first establishes the convex hull of individual codeblocks by identifying those truncation points with strictly decreasing distortion-length slope s_n, with $s_n = \frac{d_{n-1} - d_n}{l_n - l_{n-1}}$. Then, quality layers are formed selecting bitstream segments from the union of all codeblocks with the highest distortion-length slopes until the target bitrate for that layer is achieved. Through this process, each layer contains bitstream segments with equal or higher distortion-length slope than the threshold achieved for that layer, which is referred to as S_q with $1 \leq q \leq Q$, Q denoting the number of layers of the codestream. The last stage of the coding pipeline is the tier-2, which codes auxiliary information to allow the identification of quality layers within the final codestream.

The JPEG2000 codestream is structured in containers. The smallest container is the *packet*, which encapsulates data from codeblocks belonging to the same component, resolution level, and spatial area of the image, the so-called *precinct*. Data produced for each precinct are distributed in as many packets (some of them possibly null) as quality layers has the final codestream. When the image is interactively transmitted, JPIP servers generally dismantle the codestream into a collection of *data-bins* that contain all packets belonging to one precinct. When the client requests an image region, called window of interest (WOI) in what follows, the server identifies those data-bins that contain encoded data of that WOI, and transmits segments of these data-bins to the client. Rather than to interact directly with the codestream, JPIP defines a versatile dialog between client and server in which the client requests a WOI specifying spatial location, resolution, and components on the image domain, and the server identifies that WOI in the compressed domain and transmits back selected segments of the corresponding data-bins. Messages containing one or several segments of data-bins are referred to as *JPP-streams*.

An important issue that JPIP servers contemplate is the delivery of data in an optimized fashion. To be more precise, let us denote packets of the codestream as $\mathcal{T}_{c,r,p,q}$, where subindexes c, r, p, q identify the component, resolution level, spatial location, and quality layer of the packet. When the client requests a WOI, say $\mathcal{W}$, the server determines those $\mathcal{T}_{c,r,p,q}$ related to $\mathcal{W}$ and determines adequate rates for JPP-streams transmitted to the client. With an abuse of notation, we write $\mathcal{T}_{c,r,p,q} \in \mathcal{W}$ when $\mathcal{T}_{c,r,p,q}$ are related to $\mathcal{W}$.

As defined by JPEG2000, the server transmits packets according to the the layer's order, *i.e.*, packets in first layers are transmitted first. This minimizes the distortion of transmitted WOIs except when $\mathcal{W}$ does not match exactly with image precincts, which occurs often. To addres this issue, [8] suggested the application of a window scaling factor $F_{c,r,s}$ that accounts for the percentage of coefficients within that precinct that are relevant to $\mathcal{W}$. Through $F_{c,r,s}$, data is re-sequenced and transmitted to the client according to artificial distortion-length slopes [8] computed for individual packets as $S'_{c,r,s,q} = S_q \cdot F_{c,r,s}$. This

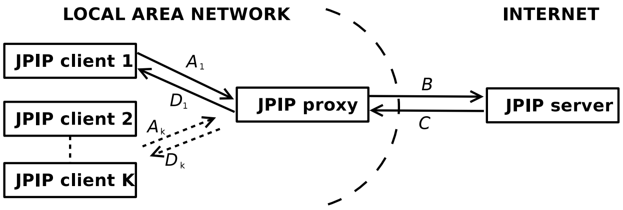


Fig. 1: The situation considered in this work has many JPIP clients sharing the same Internet link that request an image from a remote JPIP server in Internet.

procedure re-defines on-the-fly the quality layers of the codestream, which is a low-cost operation. The re-sequenced packet ordering neither assures the minimization of distortion per transmitted unit, since S_q is only an approximation of the real distortion-length slope for packet $\mathcal{T}_{c,r,s,q}$, however, such strategy is near-optimal in practice [8], [10]. As common, we assume that values for S_q are recorded in the codestream.

All transmitted data is cached by the client. When the server maintains a model of that cache, the connection is named stateful; otherwise, it is named stateless.

III. PROPOSED JPIP PROXY SERVER

A. *Rate-distortion optimized re-sequencing*

The main difficulty to take advantage of traditional HTTP proxy infrastructure is that the rich dialog between JPIP clients and servers is not understood by HTTP proxy servers. The reason that for is because petitions carried out by JPIP clients are understood by proxies as URLs, thus, only exactly the same requests are matched [11]. Hence, a specialized JPIP proxy server able to understand the JPIP protocol is necessary. The bottom line approach is to dismantle JPP-streams in the proxy, and cache data as a collection of data-bins. To replace the server efficiently, the proxy must know how to associate packets $\mathcal{T}_{c,r,p,q}$ to the image domain, and also to maintain a model of the clients' cache to allow stateful connections.

As depicted in Figure 1, client k within the LAN requests a WOI (link A_k) to the JPIP proxy server, which also lays within the LAN. Once the proxy receives the request, it checks whether that WOI is contained in cached data or not: in case the WOI is *totally* contained in cache, the proxy responses to the client transmitting JPP-streams (link D_k); in case the WOI is *partially* contained in cache, the proxy transmits cached data to the client and, simultaneously, re-formulates the petition to the server requesting non-cached data (connection $\{B,C\}$). When there are no data-bins belonging to the requested WOI in cache, the proxy sends the petition to the server and caches transmitted data. This model of JPIP proxy server, jointly with an algorithm to re-formulate client requests was first proposed in [12].

In this first approach to a JPIP proxy server, the proxy only considers the *current* WOIs requested by users. It has been shown [10] that enhanced browsing experience is achieved by anticipating the request of future WOIs. The main incentive is that there exists a delay between the emission of data by the server and the feedback that it causes to the user in terms of WOI definition, during which potential future WOIs can be requested. This delay includes the transmission time, decoding latency, and the time the user needs to interpret newly received visual information to decide next navigation commands. [10] introduced prefetching strategies at the client. When multiple clients within a LAN browse the same image, prefetching has to be carried out in the proxy to evaluate globally the responsiveness of the overall system, rather than considering only individual clients.

The JPIP proxy server proposed in our previous work [13] introduces a strategy that combines techniques of rate-distortion optimization to prefetch data from the server when the connection is idle. The proxy has two modes of operation: normal and prefetching. The *normal mode* is active when at least one client requests WOIs that are not fully contained in the proxy's cache. The *prefetching mode* is active when all client's petitions can be responded with the proxy's cache, and connection with the server ($\{B,C\}$ in Figure 1 is idle.

Let us extend the notation of the previous section to denote the currently requested WOIs by clients as $\mathcal{W}_k$, with $1 \leq k \leq K$, K being the number of active clients. In *normal mode*, the proxy computes artificial distortion-length slopes $S''_{c,r,p,q}$ for packets considering all clients browsing the same image according to

$$S''_{c,r,p,q} = S_q \cdot F_{c,r,p} \cdot \frac{1}{K} \sum_k I(\mathcal{T}_{c,r,p,q}) \ ,$$

$$\text{with } I(\mathcal{T}_{c,r,p,q}) = \begin{cases} 1 & \text{if } \mathcal{T}_{c,r,p,q} \in \mathcal{W}_k \\ 0 & \text{otherwise} \end{cases} \ .$$

The proxy identifies those $\mathcal{T}_{c,r,p,q}$ with highest values of $S''_{c,r,p,q}$, and requests the corresponding WOI to the server. Through this strategy, the overall system's performance is maximized since the highest $S''_{c,r,p,q}$ determines the packet that minimizes the clients' distortion per transmitted unit in link C. Once the packet arrives to the proxy, it is transmitted to the clients using artificial slopes $S'_{c,r,p,q}$ to form JPP-streams, thus maximizing the performance in D_k links.

B. *Prefetching with user-navigation model*

The JPIP proxy server enters in *prefetching mode* when link C is idle. The goal of the prefetching mode is to maximize the overall system responsiveness. To this end, it retrieves data of (potential) future requests anticipately. Key to maximize performance is to foresee those WOIs that are more likely to be requested by the clients. Our objective is therefore to obtain a user-navigation model that can conduct efficiently the prefetching of data in the JPIP proxy.

First, the procedure followed to obtain the user-navigation model is described. The main insight behind our model is to observe the chain of movements carried out by a user when navigating on an image. Most browsing applications allow the user to request any spatial area of the image at any of the available resolutions, regardless of the current or previous requests. Nonetheless, empirical observation indicates that aimless movements are performed only in rare occasions (see below), being more usual those movements that displace the active view to areas adjacent to the current WOI. Furthermore, the resolution of the active view – not the resolution of the image browsed – is seldom changed by the user. These observations are used to define the coordinate system employed to obtain the user-navigation model.

The idea is that once the user has requested the first WOI, the whole image –including all available resolutions– can be partitioned in equally sized regions with origin on, and dimensions of, the active view. This partitioning system is depicted in Figure 2. If the

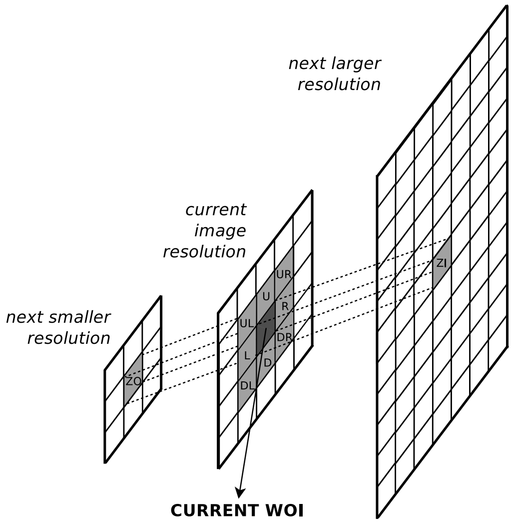


Fig. 2: Partitioning system of a multi-resolution image originating from the current WOI. The possible movements of a user in future requests are labeled as: U (up), UR (up right), R (right), DR (down right), D (down), DL (down left), L (left), UL (up left), ZI (zoom in), and ZO (zoom out).

size of the requested WOI changes, the image needs to be re-partitioned. But this does not happen often in practice and, even if it does occur, it does not affect the proposed model since only individual movements from one WOI to the next one are considered. The partitioning system of Figure 2 eases the identification of the areas adjacent to the current WOI that are susceptible to be requested in next commands. These areas are depicted in gray in the figure. There are ten next possible movements, corresponding to eight panning operations (vertical and horizontal displacements in the same resolution level), and two zooming operations (changes on the image resolution).

The individual probabilities of the 10 movements defined in the proposed model are obtained empirically. To this end, WOIs requested by clients when browsing three large resolution images have been recorded during a period of approximately 3 years. Images employed in this experiment are referred to as “Catalunya”, “Girona”, and “Lleida”, and are available with a large variety of resolution sizes in the server. These images cover large areas of Catalunya (a region of Spain) and are browsed through MiraMon, which is a popular GIS and remote sensing application developed by CREAF [14].

To analyze only browsing commands relevant to our purposes, the so-called *sessions* are identified. One session collects all commands executed by one client considering a maximum time interval of 30 minutes between consecutive requests. If two consecutive commands are delayed in more than 30 minutes, they are considered as two different sessions. This time window gives spare time to the user to receive and observe the requested WOI. Only sessions with more than three movements are considered; the remaining are considered irrelevant to our purposes and are discarded. 1629, 1203, and 923 client requests are analyzed for “Catalunya”, “Girona”, and “Lleida”, respectively.

The relevant WOI requests are translated into the ten aforementioned movements, denoted as X, with $X = \{$up, up right, right, down right, down, down left, left, up left, zoom in, zoom out$\}$. Movements that do not correspond to any of these categories

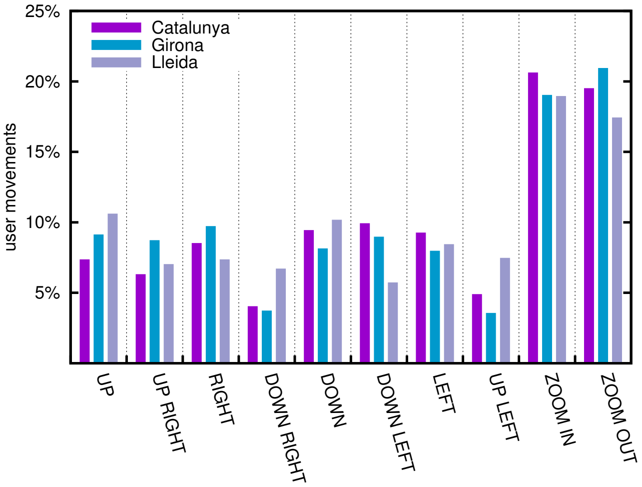


Fig. 3: Analysis of movements executed by users when navigating three different images.

represent less than 8% of all requests analyzed and are discarded. On average, the number of movements in each session is approximately 13. The probability of user movements (i.e., $P(X)$) is determined for each image considering all commands of all sessions. Figure 3 reports individual probabilities of movements for the three images analyzed. Results suggest that users navigate similarly on all images. The most used navigational commands are zooming operations, which have a probability of 40% to be executed. There is no significant difference between “zoom in” and “zoom out”. The remaining movements (corresponding to panning) have all similar probabilities.

Next, the user-navigation model determined above is introduced into the JPIP proxy server. When the JPIP proxy server is in prefetching mode, the navigational model is employed to generate artificial distortion-length slopes that aid the prefetching of data. Let us define WOIs that a client can request in next petitions as a function of the current WOI $\mathcal{W}_k$ and a user movement X as $f(\mathcal{W}_k, X)$. Evidently, $f(\cdot)$ always returns a WOI side by side $\mathcal{W}_k$ (including different resolutions) since X defined above only permits adjacent movements. When the JPIP proxy server is in prefetching mode, packets are re-sequenced using the artificial distortion-length slopes $S'''_{c,r,p,q}$ according to

$$S'''_{c,r,p,q} = S_q \cdot F_{c,r,p} \cdot \frac{1}{K} \sum_k J(\mathcal{T}_{c,r,p,q}) \text{ , with}$$

$$J(\mathcal{T}_{c,r,p,q}) = \sum_X \begin{cases} P(X) & \text{if } \mathcal{T}_{c,r,p,q} \in f(\mathcal{W}_k, X) \\ 0 & \text{otherwise} \end{cases} .$$

Again, the proxy requests that WOI corresponding to the packet with highest $S'''_{c,r,p,q}$, and caches the response hoping that future user movements will request that area. As indicated by the previous analysis, $P(X) = 20\%$ when $X = \{\text{zoom in} \mid \text{zoom out}\}$, and $P(X) = 7.5\%$ for the eight panning movements. Finer adjustment of these probabilities does not seem to improve performance significantly.

To summarize, delivery of data in links D_k is conducted through $S'_{c,r,p,q}$, and WOIs are requested in link B through $S''_{c,r,p,q}$ when the proxy is in normal mode, and through

$S'''_{c,r,p,q}$ when the proxy is in prefetching mode. Again, strict optimal performance is not guaranteed due to the use of artificial distortion-length slope for packets.

IV. EXPERIMENTAL RESULTS

All the infrastructure used to evaluate the proposed method, including the client, the server, and the JPIP proxy server are implemented in our JPEG2000 Part 9 implementation CADI [15]. Experiments are carried out in a LAN with a network bandwidth of 100 Mbps, and a 1 Mbps connection to the Internet shared by all clients. This set up is similar to that found in emergency missions. The image used in these experiments is a Landsat image of size 7200×5000 (24-bit color) covering an area of Barcelona. JPEG2000 coding parameters are: 5 levels of 9/7 DWT, codeblock size of 64×64, restart coding variation, and 25 quality layers logarithmically spaced in terms of bitrate. The server adjusts on-the-fly the precinct size in all resolution levels according to the codeblock size to enhance interactivity options.

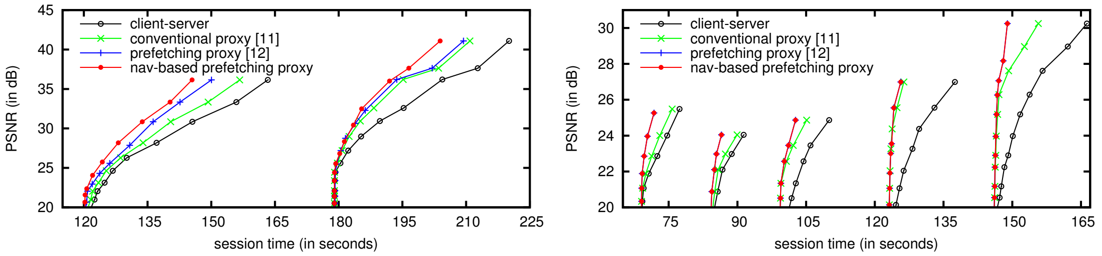
Results are evaluated when four users browse the image simultaneously. This may correspond, for example, to an assessment period during an emergency situation when four members of a team are in a camp evaluating regions that may require urgent help. Four transmission strategies are evaluated:

1) *Client-server*: the image transmission is carried out without using any proxy infrastructure.
2) *Conventional JPIP proxy*: all data transmitted between the server and the four clients is cached by a JPIP proxy that re-uses already transmitted packets. This is the proxy server introduced in [12].
3) *JPIP proxy with simple prefetching*: this strategy is the same as the previous one but, in addition to cache all transmitted data, the proxy uses idle time to prefetch more data from the server without using any user-navigation model. This proxy was introduced in our previous work [13].
4) *JPIP proxy with navigational-based prefetching*: this strategy extends our previous work by introducing the user-navigation model in the proxy, where prefetched data have a higher probability to be requested by users.

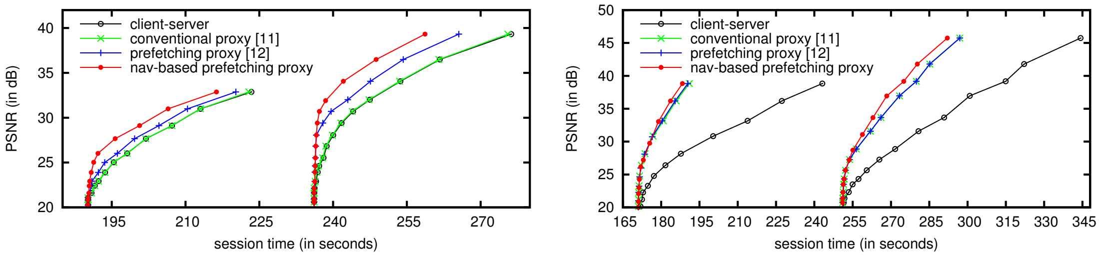
The results are reported for the four transmission strategies when the same browsing sessions are executed by the four clients. In other words, the exact chain of movements is exactly executed by all clients to obtain results for the four transmission strategies. The chain of movements selected for each user corresponds to typical browsing sessions exploring areas of an image. We allow users to perform 10 movements during a session that lasts 8 minutes approximately. This is a common behavior when members of an emergency team explore areas of a territory using earth observation images.

Figure 4 reports the results achieved by the four clients. The vertical axis of the figures is the quality of the retrieved image, in terms of Peak Signal to Noise Ratio (PSNR). The horizontal axis is the browsing session time. Only selected WOIs for each client are depicted in these figures for the sake of clarity. All plots starting from the same point, approximately, correspond to the same WOI retrieved by the client. WOIs are decoded at the end of each quality layer, hence, the quality achieved by all strategies when decoding the same WOI is equal, changing only the instant of time in which the layer is completely available at the client.

In general, the first WOIs retrieved by all clients are recovered similarly since no data are in the proxy cache. Thus, these first WOIs are not shown in the plots. This



(a) Client 1 - WOI 7 and 8 (b) Client 2 - WOI 1 to 5



(c) Client 3 - WOI 8 and 9 (d) Client 4 - WOI 8 and 9

Fig. 4: Coding performance evaluation achieved when four clients browse a Landsat image using four transmission strategies.

observation holds for all strategies except for the first one, which recovers all WOIs (including also those of Figure 4) slower than the remaining strategies due to the lack of a proxy infrastructure. The differences between the conventional proxy (strategy 2) and the proxies that prefetch data (strategies 3 and 4) become evident after the second minute of the session, approximately, since there have been already enough idle moments during consecutive user commands to prefetch more data from the server. The differences are relevant for some WOIs. For example, WOI 9 retrieved by Client 3 is recovered 4.64 dB better than a conventional proxy server at second 240.

It is noteworthy that the proxy that uses the user-navigation model (strategy 4) is able to enhance the responsiveness of the system since the first idle periods. Note, for instance, that the navigational-based prefetching proxy delivers all layers of WOI 7 requested by Client 1 after 25 seconds of receiving the request. The simple prefetching strategy requires 30.5 seconds to deliver the same layers, whereas the conventional proxy without prefetching requires 36.2 seconds. Similar performance is achieved by other WOIs and clients. These results suggest that, under these conditions, the JPIP proxy server using the user-navigation model improves the responsiveness of the overall system significantly, which results in a better browsing experience for the end-users. We remark that the reported results are obtained when browsing Earth Observation images. Navigation models for other types of images may achieve a different structure.

V. CONCLUSIONS

The efficient transmission of earth observation images is an important aspect of emergency missions to help teams to explore and assess the damaged areas of a territory. This work is concerned with the browsing of large resolution images by clients in a camp sharing a connection link. Since the Internet connection may be of low capacity, it is important to set up an infrastructure that assures a satisfactory experience to end-users. For this purpose, the research presented in this paper analyzes the behavior of users when navigating on an image. This analysis employs data collected during three years by applications of interactive image transmission. The analysis results in a user-navigation model that suggests that ten movements are the most frequently performed. From these ten navigational commands, two correspond to zooming in and zooming out, which have 40% probability of being executed, whereas the remaining ones are panning movements, with a 60% probability of being executed. This model is introduced in a JPIP proxy server that, in addition to cache all data transmitted from the server to clients, prefetchs more data when the connection link to the server is not utilized by the clients. Experimental results suggests that the use of the navigational-model in the proxy server enhances the browsing experience notably compared to conventional proxy servers.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Government (MICINN), by the European Union, by FEDER, and by the Catalan Government, under Grants RYC-2010-05671, TIN2009-14426-C02-01, FP7-SPACE-2009_1-242390-GEOPICTURES, 2009-SGR-1224, and 2009-SGR-1511. Xavier Pons is recipient of an ICREA Acadèmia Excellence in Research grant (2011-2015).

REFERENCES

[1] "GMES and earth observation combined with position based image and sensor communications technology for universal rescue, emergency and surveillance (GEO-PICTURES)," European Union Framework Program VII, Space Project FP7- 242390, 2010. [Online]. Available: http://www.geo-pictures.eu/

[2] *Information technology - JPEG 2000 image coding system - Part 1: Core coding system*, ISO/IEC Std. 15 444-1, Dec. 2000.

[3] S. Deshpande and W. Zeng, "Scalable streaming of JPEG2000 images using hypertext transfer protocol," in *Proc. ACM International Conference on Multimedia*, vol. 9, Oct. 2001, pp. 372–381.

[4] D. Taubman, "Remote browsing of JPEG2000 images," in *Proc. IEEE International Conference on Image Processing*, Sep. 2002, pp. 229–232.

[5] J. Li and H.-H. Sun, "On interactive browsing of large images," *IEEE Trans. Multimedia*, vol. 5, no. 4, pp. 581–590, Dec. 2003.

[6] E. A. Politou, G. P. Pavlidis, and C. Chamzas, "JPEG2000 and dissemination of cultural heritage over the internet," *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 293–301, Mar. 2004.

[7] *Information technology - JPEG 2000 image coding system: Interactivity tools, APIs and protocols*, ISO/IEC Std. 15 444-9, Dec. 2005.

[8] D. Taubman and R. Rosenbaum, "Rate-distortion optimized interactive browsing of JPEG2000 images," in *Proc. IEEE International Conference on Image Processing*, vol. 3, Sep. 2003, pp. 765–768.

[9] K. Krishnan, M. W. Marcellin, A. Bilgin, and M. S. Nadar, "Efficient transmission of compressed data for remote volume visualization," *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1189–1199, Sep. 2006.

[10] A. Descampe, C. D. Vleeschouwer, M. Iregui, B. Macq, and F. Marques, "Prefetching and caching strategies for remote and interactive browsing of JPEG2000 images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1339–1354, May 2007.

[11] J. P. G. Ortiz, V. G. Ruiz, M. F. Lopez, and I. Garcia, "Interactive transmission of JPEG2000 images using web proxy caching," *IEEE Trans. Multimedia*, vol. 10, no. 4, pp. 629–636, Jun. 2008.

[12] L. Lima, D. Taubman, and R. Leonardi, "JPIP proxy server for remote browsing of JPEG2000 images," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Oct. 2008, pp. 844–849.

[13] J. L. Monteagudo-Pereira, F. Auli-Llinas, J. Serra-Sagrista, and J. Bartrina-Rapesta, "Smart JPIP proxy server with prefetching strategies," in *Proc. IEEE Data Compression Conference*, Mar. 2010, pp. 99–108.

[14] X. Pons. (2004) MiraMon. Geographic Information System and Remote Sensing software. Centre de Recerca Ecolgica i Aplicacions Forestals, CREAF. Bellaterra. ISBN: 84-931323-5-7. [Online]. Available: http://opengis.uab.cat/wms/satcat/

[15] J. L. Monteagudo-Pereira. (2011) CADI. Group on Interactive Coding of Images. GICI. [Online]. Available: http://www.gici.uab.cat/CADI

3.4 Prefetching Based on User-Navigation Model and Semantic Map

```
@article {MAS12,
    AUTHOR = {J. Lino Monteagudo-Pereira and Francesc Auli-Llinas and Joan Serra-Sagrista},
   JOURNAL = {IEEE Transactions on Multimedia},
     TITLE = {JPIP Proxy Server with Prefetching Strategies Based on User-Navigation Model and
              Semantic Map},
    VOLUME = {},
    NUMBER = {},
     PAGES = {},
     MONTH = {},
      YEAR = {2013},
      ISSN = {1520-9210},
       DOI = {},
}
```

JPIP Proxy Server with Prefetching Strategies Based on User-Navigation Model and Semantic Map

J. Lino Monteagudo-Pereira, Francesc Aulí-Llinàs, *Member, IEEE*, and Joan Serra-Sagristà, *Senior Member, IEEE*

***Abstract*—The efficient transmission of large resolution images and, in particular, the interactive transmission of images in a client-server scenario, is an important aspect for many applications. Among the current image compression standards, JPEG2000 excels for its interactive transmission capabilities. In general, three mechanisms are employed to optimize the transmission of images when using the JPEG2000 Interactive Protocol (JPIP): 1) packet re-sequencing at the server; 2) prefetching at the client; and 3) proxy servers along the network infrastructure. To avoid the congestion of the network, prefetching mechanisms are *not* commonly employed when many clients within a local area network (LAN) browse images from a remote server. Aimed to maximize the responsiveness of all the clients within a LAN, this work proposes the use of prefetching strategies at the proxy server –rather than at the clients. The main insight behind the proposed prefetching strategies is a user-navigation model and a semantic map that predict the future requests of the clients. Experimental results indicate that the introduction of these strategies into a JPIP proxy server enhances the browsing experience of the end-users notably.**

***Index Terms*—Interactive image transmission, JPEG2000, JPIP, prefetching strategies, user-navigation model, semantic map.**

I. INTRODUCTION

THE efficient transmission of large resolution images is a requirement in many applications related to territory management, telemedicine, disaster monitoring, or map navigation, among others. To provide a pleasant browsing experience, the transmission of the image areas requested by the users has to be optimized. Compression and scalability are fundamental aspects of the employed coding system to do so.

Among the current standards to code and transmit images, JPEG2000 excels for its coding performance and transmission capabilities. Part 1 of the standard [3] defines the core coding system, which is wavelet-based with a two tiered coding strategy. Fundamental features of JPEG2000 are support for high resolution images (in terms of spatial, spectral, or bit-depth), and scalability by spatial location, by quality, by resolution, and by component. This high degree of scalability is partially provided by a rich codestream syntax that allows random access to the file. The ability to work in the codestream domain, i.e., to identify and potentially transmit any portion of the image without needing to decode the codestream, makes JPEG2000 one of the most suitable coding standards for interactive image transmission.

The potential of JPEG2000 to interactively transmit images was explored soon after the publication of Part 1 [4]–[8]. In November 2005, JPEG2000 Part 9 [9] was published providing a versatile client-server syntax for the interchange of imagery that is named JPEG2000 Interactive Protocol (JPIP). Since then, JPIP has been adopted in fields such as medicine [10], [11], remote sensing [12], [13], or video-on-demand [14]–[16], among others.

The main asset of JPIP is the rapid recovery of the image areas requested by the client. Several mechanisms have been proposed in the literature to improve this aspect. An efficient technique employed in the server is to dismantle and re-sequence the original codestream [17]. Another mechanism is to utilize prefetching at the client. Prefetching refers to the ability to anticipate the future movements of the user so that the corresponding data can be retrieved during the moments in which the connection between the server and the client is idle [18]. The use of a proxy server is yet another mechanism that improves the interactive transmission capabilities of JPIP in some scenarios. Let us explain further. Commonly, in centers or institutions users browse images located in a remote server sharing the same Internet connection. In general, the channel capacity of the local area network (LAN) is much larger than that of the Internet. In such a scenario, a proxy server caches and reuses the data transmitted from the server to any of the clients, avoiding the transmission of the same information more than once. Fig. 1 illustrates this scenario. In the framework of JPEG2000, proxy mechanisms were first explored in [19] re-defining the JPIP syntax so that the conventional HTTP proxy infrastructure deployed in centers and in the Internet were able to understand the dialog between a JPIP server and a client. Unfortunately, that approach is not compliant with the standard. The first compliant JPIP proxy server was introduced in [20] employing re-sequencing techniques similar to those of [17] to optimize the delivery of data to the clients in the LAN.

As seen in the experiments reported in Section V, it is not recommendable that the clients use prefetching individually in the scenario of Fig. 1 because that might saturate the Internet connection. None of the JPIP proxy servers proposed in the literature employs prefetching strategies, either. So, typically, the Internet connection $\{B,C\}$ depicted in Fig. 1

J. Lino Monteagudo-Pereira, Dr. Francesc Aulí-Llinàs, and Dr. Joan Serra-Sagristà are with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Spain (phone: +34 935813571; fax: +34 935814477; e-mail: jlino@deic.uab.cat, fauli@deic.uab.cat, joan.serra@uab.cat). A preliminary version of this work was presented at the IEEE Data Compression Conference 2010 and 2012 [1], [2]. This work has been partially supported by the Spanish Government (MINECO), by FEDER, and by the Catalan Government, under Grants RYC-2010-05671, TIN2009-14426-C02-01, TIN2012-38102-C03-03, and 2009-SGR-1224.

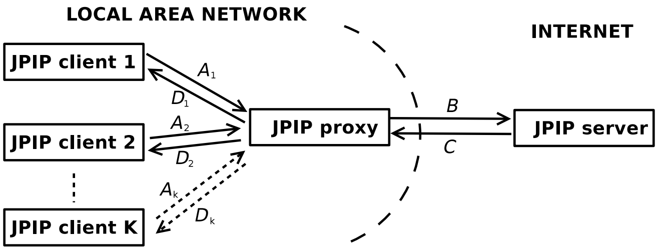


Fig. 1: JPIP clients within a LAN share the same Internet connection to browse images located in a remote JPIP server. Data transmitted from the server to the clients is cached and reused by a JPIP proxy server.

is left idle when users have retrieved the requested areas. The purpose of this work is to introduce prefetching strategies in the JPIP proxy server –rather than to each client– to enhance the browsing experience of the clients within the LAN. The JPIP proxy server proposed herein employs a rate-distortion optimized algorithm that maximizes the responsiveness of the whole system taking into account the current and (possible) future areas requested by the clients. The functionality of the proposed algorithm is the same if the proxy is introduced in scenarios with a more complex topology than that of Fig. 1. Prefetching is carried out only by the proxy when the Internet connection is idle. Two strategies are put in practice to predict the areas that the clients may request in the future. The first employs a user-navigation model that has been extracted from the logs collected by several image servers belonging to the remote sensing community. The second strategy utilizes a semantic map that prioritizes areas of the image depending on their content. Experimental results report significant gains with respect to classical client-server approaches and with respect to conventional JPIP proxy servers without prefetching strategies. The research of this paper extends our previous works [1], [2] by describing a more accurate user-navigation model, the novel semantic map-based prefetching strategy, and by providing extended experimental results.

The paper is structured as follows. Section II reviews fundamentals concepts of the JPEG2000 core coding system, the codestream organization, and the JPIP protocol. The architecture and functionalities of the proposed JPIP proxy server are described in Section III, whereas Section IV introduces the two prefetching strategies proposed. The performance achieved by our JPIP proxy server and other conventional transmission strategies is assessed in Section V through experimental results carried out for large remote sensing images. The last section concludes this work with a brief summary and some remarks.

II. OVERVIEW OF JPEG2000 AND JPIP

The JPEG2000 core coding system is constituted by four main coding stages [21]: sample data transformation, sample data coding (tier-1), rate-distortion optimization, and codestream re-organization (tier-2). The first stage applies a wavelet transform that decorrelates and decomposes the image in successive levels of resolution. Then, the image is conceptually partitioned in small sets of wavelet coefficients, called codeblocks, that are coded by the tier-1 coding stage by means of a bitplane coding engine. Tier-1 produces a quality embedded bitstream for each codeblock that can be truncated at increasing rates. The third stage of the coding system forms quality layers using rate-distortion optimization techniques. A quality layer is defined as a collection of bitstream segments from different codeblocks. Its transmission and/or decoding represents an increment on the quality of the image. Commonly, the rate-distortion optimization stage employs Lagrange optimization to minimize the distortion at the target rates selected for the quality layers and for the final codestream. To do so, first the convex hull of individual codeblocks is established by identifying those truncation points with strictly decreasing distortion-rate slope. If d_n and r_n respectively denote the distortion and the rate achieved at the truncation points of the bitstream generated for a codeblock, the distortion-rate slope is defined as $s_n = \dfrac{d_{n-1} - d_n}{r_n - r_{n-1}}$. Quality layers are then formed selecting bitstream segments from the union of all codeblocks with the highest distortion-rate slopes. Through this process, each layer contains bitstream segments with equal or higher distortion-rate slope than the threshold achieved for that layer, which is referred to as S_q with $1 \leq q \leq Q$, Q denoting the number of layers of the codestream. The last stage of the coding pipeline is the tier-2, which codes auxiliary information. In this and following discussions we assume that the image is not partitioned in tiles [21, Ch. 11.2] since interactive transmission is commonly not benefited from such partitions [5].

As depicted in Fig. 2, the JPEG2000 codestream is structured in containers. The smallest container is the *packet*, which encapsulates segments of the bitstreams generated from codeblocks that belong to the same component, resolution level, and spatial area of the image, the so-called *precinct*. Data produced for each precinct are distributed in as many packets (some of them possibly null) as quality layers has the final codestream. When the image is interactively transmitted, JPIP servers generally dismantle the codestream into a collection of *data-bins* that contain all packets belonging to one precinct. When the client requests an image region, called window of interest (WOI), the server identifies those data-bins containing encoded data from that WOI and transmits segments of these data-bins to the client. Rather than to interact directly with the codestream, JPIP defines a versatile dialog between client and server in which the client requests a WOI specifying spatial location, resolution, and components on the image domain, and the server identifies that WOI in the compressed domain. Messages containing one or several segments of data-bins are referred to as *JPP-streams*.

As mentioned above, an important issue that JPIP servers contemplate is the delivery of data in an optimized fashion. Let us denote the packets of the codestream as $\mathcal{T}_{c,r,p,q}$, where subindexes c, r, p, q identify the component, resolution level, spatial location, and quality layer of the packet, respectively. When the client requests a WOI, say $\mathcal{W}$, the server identifies the packets related to the WOI, which we denote as $\mathcal{T}_{c,r,p,q} \succ \mathcal{W}$, and determines adequate rates for JPP-streams transmitted to the client.

To optimize the rate-distortion efficiency, the server com-

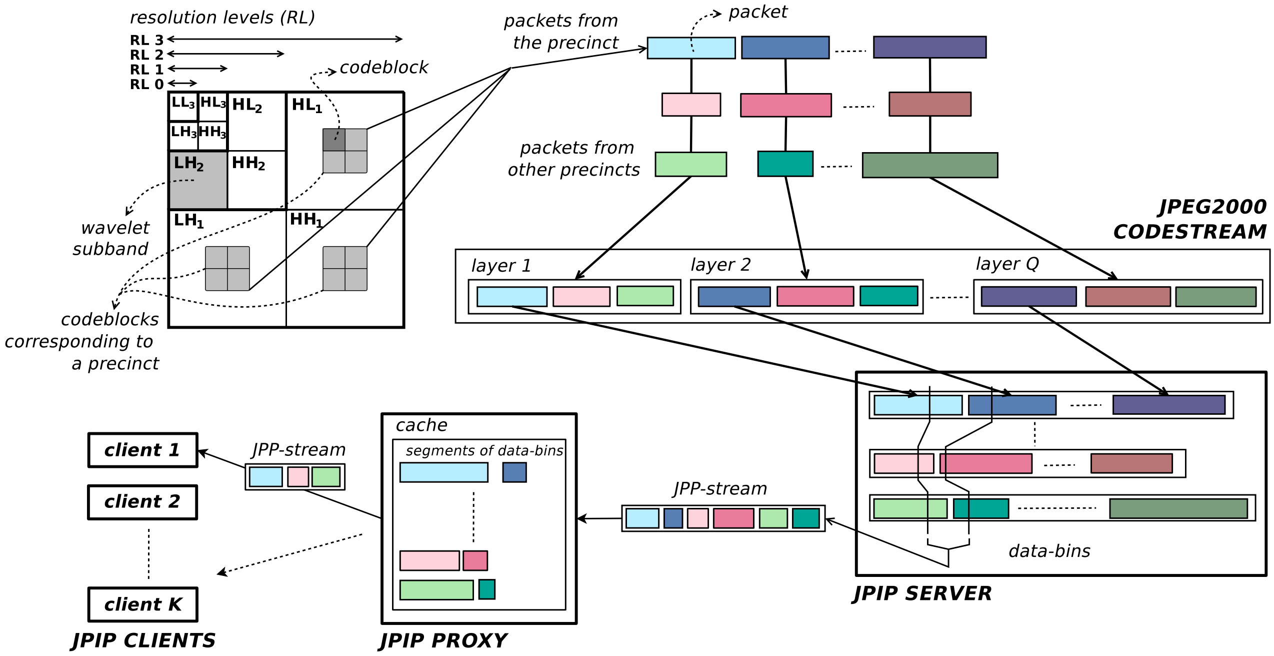


Fig. 2: Simplified overview of the JPEG2000 partitioning system, the organization of the codestream, and the data transmitted by the JPIP server, proxy, and client.

monly selects packets according to the layer's order, i.e., packets in first quality layers are transmitted first. Since $\mathcal{W}$ does not typically match exactly with image precincts, [17] suggested the application of a window scaling factor $F_{c,r,p} \in (0, 1]$ that accounts for the percentage of coefficients within that precinct that are relevant to $\mathcal{W}$. Through $F_{c,r,p}$, data is re-sequenced and transmitted to the client according to artificial distortion-rate slopes computed for individual packets as

$$S'_{c,r,p,q} = S_q \cdot F_{c,r,p}. \qquad (1)$$

This re-sequenced packet ordering does not assure the minimization of distortion per transmitted unit since S_q is only an approximation of the real distortion-rate slope for packet $\mathcal{T}_{c,r,p,q}$. The real distortion-rate slope of $\mathcal{T}_{c,r,p,q}$ is not stored in the codestream because it requires a significant rate and is not necessary to decode the image. This poses an issue that has been approached in the literature from different points of view [14], [17], [18], [22], [23]. As reported in these works, the use of S_q in Equation 1 provides near optimal performance in practice. As common, we assume that values for S_q are recorded in the codestream.

All data transmitted by the server is cached by the client. The server can maintain a model of the client's cache to avoid the retransmission of the same data. In this case the connection is named stateful. When the server does not maintain the model of the client's cache, the client may include a description of its cache when requesting WOIs. In this case, the connection is named stateless. Of the two, stateful connections are more convenient to minimize the transmission of side information [24].

III. PROPOSED JPIP PROXY SERVER

A. *System architecture*

The proposed JPIP proxy server maintains the full set of JPIP features, keeping compliance with the standard. The main ability of the proxy is to understand the JPIP protocol so that it can dismantle and cache the JPP-streams transmitted by the server as a collection of data-bins (see Fig. 2). These data-bins are associated to packets in the codestream and to precincts in the image domain, which allows their optimized re-sequencing to respond the clients' requests. To replace the server efficiently, the JPIP proxy also maintains a model of the clients' cache to allow stateful connections.

As observed by many authors [18], [25], [26], there exists a delay between the emission of data by the server and the feedback provided by the user before requesting a new WOI. This delay includes the transmission time, decoding latency, and the time that the user needs to interpret newly received visual information. During this time, the connection between the client and the server is idle. As stated before, idle moments can be employed to prefetch data that may be requested in the future by the client(s). In the scenario of Fig. 1, prefetching should not be directly applied to all clients within a LAN as formulated in [18] because the joint transmission of WOIs that are in current use, with WOIs that *might* possibly be required, could cause the assignment of the available channel resources to useless queries, providing negligible gains on the overall system performance, as is experimentally demonstrated in Section V.

Our approach introduces prefetching strategies in the JPIP proxy server combining techniques of rate-distortion optimization with a model of probabilities that predicts the next users' movements. The proxy provides prefetching to all clients within the LAN simultaneously. The proposed proxy server has two modes of operation: normal and prefetching. The *normal mode* is active when at least one client requests a WOI that is not fully contained in the proxy's cache. The *prefetching mode* is active when all client's petitions can be responded with the cached data, and so the connection between the proxy and the server is idle.

The functions that are carried out by the proxy server when it is in normal mode are similar to those of a conventional proxy [20]. When a client within the LAN requests a WOI to the JPIP proxy server (link A_k in Fig. 1), the proxy checks whether that WOI is contained in its cache or not. In the case that there are no data-bins belonging to the requested WOI in the cache, the proxy sends the petition to the server (connection $\{B,C\}$), and responds to the client caching all data transmitted from the server. In the case that the WOI is *partially* contained in the cache, the proxy transmits cached data to the client and, simultaneously, re-formulates the petition to the server requesting non-cached data. In the case that the WOI is *totally* contained in the cache, the proxy responds to the client transmitting JPP-streams (link D_k). Evidently, the proxy can handle more than one request at the same time, so this procedure is executed in parallel for all concurrent requests.

To minimize the interchange of image data between the proxy and the server when the proxy is in normal mode, WOIs requested to the server are re-formulated. Rather than including several areas demanded by clients as in [20], our proxy requests specific WOIs one by one in the order that maximizes the overall system's performance. Let us extend the notation of the previous section to denote the WOIs currently requested by the clients as $\mathcal{W}_k$, with $1 \leq k \leq K$, K being the number of active clients. In normal mode, the proxy computes artificial distortion-rate slopes $S''_{c,r,p,q}$ for packets considering all clients browsing the same image according to

$$S''_{c,r,p,q} = S_q \cdot F'_{c,r,p} \cdot \frac{1}{K} \sum_k \delta(\mathcal{T}_{c,r,p,q}, k) \ \forall \ c, r, p, q \ , \quad (2)$$

with

$$\delta(\mathcal{T}_{c,r,p,q}, k) = \begin{cases} 1 & \text{if } \mathcal{T}_{c,r,p,q} \succ \mathcal{W}_k \\ 0 & \text{otherwise} \end{cases} . \quad (3)$$

$\delta(\mathcal{T}_{c,r,p,q}, k)$ is a binary function that ascertains whether packet $\mathcal{T}_{c,r,p,q}$ is necessary to serve the current WOI $\mathcal{W}_k$ requested by client k. When a packet is necessary to serve the WOIs from *all* the clients, $\frac{1}{K} \sum_k \delta(\mathcal{T}_{c,r,p,q}, k)$ in (2) results in 1. Otherwise, it results in the fraction of clients requesting that packet. $F'_{c,r,p} \in (0, 1]$ in Equation (2) accounts for the percentage of coefficients in the precinct that are relevant to the WOIs of all clients. We note that $S''_{c,r,p,q}$ is computed for all the packets of the codestream.

Packets are requested by the proxy in decreasing order of $S''_{c,r,p,q}$. This strategy maximizes the overall system's performance since the highest $S''_{c,r,p,q}$ corresponds to that packet that minimizes the clients' distortion per transmitted unit in link C of Fig. 1. Once the packet arrives to the proxy, it is transmitted to the clients using artificial slopes $S'_{c,r,p,q}$ to form JPP-streams, thus maximizing the performance in D_k links.

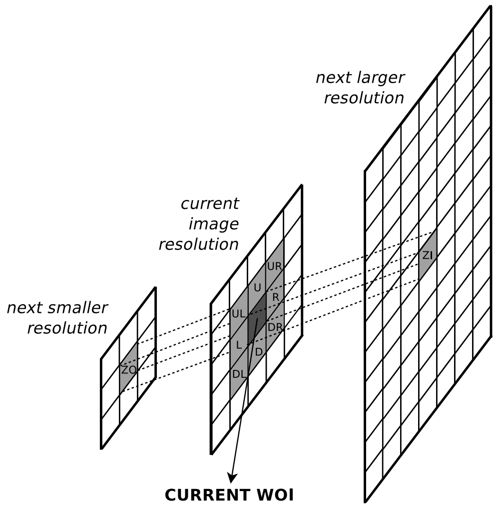


Fig. 3: Partitioning system of a multi-resolution image originating from the current WOI. The possible movements of a user in future requests are labeled as: U (up), UR (up right), R (right), DR (down right), D (down), DL (down left), L (left), UL (up left), ZI (zoom in), and ZO (zoom out).

In prefetching mode, the JPIP proxy server requests WOIs that may be demanded in the future by the clients. Although most browsing applications allow the user to request any spatial area of the image at any of the available resolutions, the evaluation carried out in Section IV-A indicates that aimless movements are only 5% of all movements executed in a normal browsing session, being more usual those movements that displace the active view to areas adjacent to the current WOI. Furthermore, the resolution of the active view –not the resolution of the image browsed– is seldom changed by the user, being approximately only 3% of all movements executed in a browsing session as indicated below. These observations are employed to define the coordinate system of the potential future WOIs requested by the clients.

The idea is that once the user has requested the first WOI, the whole image –including all available resolutions– can be partitioned in equally sized regions with origin on, and dimensions of, that first WOI. This partitioning system, which is depicted in Fig. 3, eases the identification of the areas adjacent to the current WOI that are susceptible to be requested in next commands. Evidently, the partitioning is carried out particularly for each user. If the size of the requested WOI changes, the image needs to be re-partitioned. But this does not happen often in practice and, even if it does occur, it does not affect the proposed model since only individual movements from one WOI to the next are considered. These areas are depicted in gray in the figure. There are ten next possible movements corresponding to eight panning operations (vertical and horizontal displacements in the same resolution level), and two zooming operations (changes on the image resolution). We note that the "zoom in" operation may consider not only the central region of the current WOI in the next larger resolution (as it is depicted in Fig. 3) but all regions at the larger resolution level that correspond to the current WOI. Our partitioning system considers only the central region due to the empirical observations reported in Section IV-A indicate that that is the most frequent "zoom in" operation carried out by users.

The movements carried out by the user are denoted as X, with $X = \{$ up | up right | right | down right | down | down left | left | up left | zoom in | zoom out $\}$. The WOIs that the clients can request in the future are defined as a function of the current WOI $\mathcal{W}_k$ and a user movement X as $f(\mathcal{W}_k, X)$. Evidently, $f(\cdot)$ always returns a WOI side by side $\mathcal{W}_k$ (including different resolutions) since X is defined only as adjacent movements. When the JPIP proxy server is in prefetching mode, artificial distortion-rate slopes $S'''_{c,r,p,q}$ are computed as

$$S'''_{c,r,p,q} = S_q \cdot F'_{c,r,p} \cdot \frac{1}{K} \sum_k \delta'(\mathcal{T}_{c,r,p,q}, k) \; \forall \; c, r, p, q \; , \quad (4)$$

with

$$\delta'(\mathcal{T}_{c,r,p,q}, k) = \sum_X \begin{cases} P(X) & \text{if } \mathcal{T}_{c,r,p,q} \succ f(\mathcal{W}_k, X) \\ 0 & \text{otherwise} \end{cases} . \quad (5)$$

Equation (4) is equal to (2) except for the use of $\delta'(\cdot)$. $\delta'(\cdot)$ is a function that returns the probability of packet $\mathcal{T}_{c,r,p,q}$ to be requested by client k in future movements. The summation in Equation (5) accounts for the probabilities of all possible future movements of the client with regard to that packet. $P(X)$ in (5) is the probability that the user executes the movement in future requests. It is determined through the strategies described in the next section. When the proxy is in prefetching mode, it requests WOIs in decreasing order of $S'''_{c,r,p,q}$ and caches the response hoping that future user movements will request prefetched data. Prefetched data are kept in cache even when the prediction fails because other users may browse that area of the image in future requests.

Briefly summarized, the functions carried out by the JPIP proxy server are the delivery of data to the clients through links D_k using $S'_{c,r,p,q}$, and the requesting of WOIs to the server through link B using $S''_{c,r,p,q}$ when the proxy is in normal mode, and using $S'''_{c,r,p,q}$ when the proxy is in prefetching mode. As in [17], [20], strict optimal performance is not guaranteed due to the use of artificial distortion-rate slope for packets.

B. Implementation considerations

Some considerations have to be taken into account to implement the proposed JPIP proxy server:

- The connection $\{B,C\}$ between proxy and server must use the JPIP options ALIGN and AUX. ALIGN forces the JPIP server to deliver non-segmented packets, which is required by the proxy to identify individual packets $\mathcal{T}_{c,r,p,q}$ without needing to decode data. AUX forces the JPIP server to include a field that specifies the layers to which packets belong to. These options are not required in links A_k, D_k.
- The JPIP proxy server requests WOIs (link B) specifying the number of layers through the JPIP option LAYERS. This is mandatory to force the server to transmit a specific number of layers for the requested precinct. Note that this blocks any rate control mechanism used in the server to transmit data, intentionally leaving the control to the proxy. Although this causes requests of multiple WOIs in a short period of time, the results reported in Section V indicate that this strategy does not saturate the network link to the server, working efficiently in practice. To force the server to complete the transmission of requested WOIs, the proxy must include the parameter WAIT in requests. This assures that the delivery of packets is carried out as intended by the proxy and that transmissions are not interrupted as more requests arrive to the server.
- We assume that values for S_q are available to the proxy through their transmission in COM markers of the codestream, or through rough estimations as described in [27].
- As seen in Fig. 1, the server sees the proxy as a single client, thus the use of a stateful connection is highly recommended in connection $\{B,C\}$. Connections between the proxy and clients can be stateless.
- The operations carried out by the proxy do not entail the re-encoding of any part of the codestream, so the system can be escalated with the proxy handling many clients simultaneously [20].
- The administration of the LAN must consider that a failure in the proxy server is critic since it leaves all clients without connection. This may be solved using a high availability cluster, for example.

IV. PREFETCHING STRATEGIES

The goal of the JPIP proxy server when it is in prefetching mode is to optimize the responsiveness of all clients. Key to achieve this goal is to foresee the WOIs that are more likely to be requested by the clients. The objective is then to obtain a precise model for probabilities $P(X)$ employed in Equation (5). Two approaches are used. The first is based on a user-navigation model that captures the common behavior of users when navigating on an image. The second approach utilizes the content of the image to predict the probability of an image area to be visited more or less frequently. The former approach can be employed when the content of the image is not known or when it is too arduous to determine, whereas the latter can be employed otherwise.

A. User-navigation model

The user-navigation model is described first. The main insight behind this approach is to observe the chain of movements carried out by users when navigating on an image. The individual probabilities of the 10 movements defined in Fig. 3 are obtained as follows. The first step was to record WOIs requested by clients when browsing four large resolution

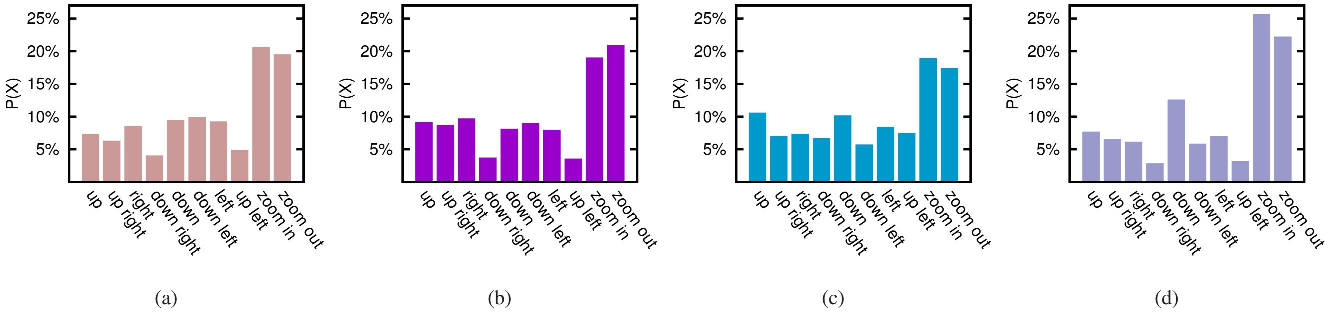


Fig. 4: Evaluation of the movements carried out by users when navigating on the images of the remote sensing corpus (a) “Barcelona” 1629 sessions, (b) “Girona” 1203 sessions, (c) “Lleida” 923 sessions, and (d) “Catalunya” 228016 sessions.

images during a period of approximately 3 years. Images employed in this experiment are referred to as “Barcelona”, “Girona”, “Lleida”, and “Catalunya”, and are available with a large variety of resolution sizes in the server. The size of the largest resolution is approximately 13000×13000 pixels. The images belong to the remote sensing community and cover large areas of Catalunya (a region of Spain). They are browsed through MiraMon, which is a popular remote sensing application developed by CREAF [28]. In general, the size of the user’s view is set to 800×600 pixels in this application.

To analyze only browsing commands relevant to our purposes, the second step of this analysis is to identify the so-called *sessions* in the logs gathered by the server. One session collects all commands executed by one client considering a maximum time interval of 30 minutes between consecutive requests. If two consecutive commands are delayed in more than 30 minutes, they are considered as two different sessions. This time window gives spare time to the user to receive and observe the requested WOI. Only sessions with more than three movements are considered; the remaining are irrelevant to our purposes and are discarded. In total, the selected sessions accumulate 1629, 1203, 923, and 228016 client requests for “Barcelona”, “Girona”, “Lleida”, and “Catalunya”, respectively.

The third step of the analysis is to translate the relevant WOI requests into the ten movements defined by X. Movements that do not correspond to any of these categories represent less than 8% of all requests analyzed and are discarded. The discarded movements correspond to changes on the resolution of the active view, or aimless movements. On average, the number of movements in each session is approximately 13. The average time that the users spend on a browsing session is 7 minutes, approximately. The probability of user movements (i.e., $P(X)$) is determined for each image considering all commands of all sessions. Fig. 4 reports the individual probabilities achieved for $P(X)$, for the four images. Results suggest that users navigate similarly on images “Barcelona”, “Girona”, and “Lleida”. For these images, the navigational commands that are most frequent correspond to the zooming operations, which have a probability of 40% to be executed, approximately. There is no significant difference between “zoom in” and “zoom out”. The remaining movements corresponding to panning operations present slight differences among the three images, though they are not significant. Our experience indicates that $P'(X) = 20\%$ when $X = \{\text{zoom in} \mid \text{zoom out}\}$ and $P'(X) = 7.5\%$ when $X = \{$ up | up right | right | down right | down | down left | left | up left $\}$ is a model that, in this context, captures well the behavior of most users browsing an image. Finer adjustment of these probabilities does not seem to improve performance significantly. The use of conditional probabilities considering the last movement of the user to predict the next one (not shown in the figures) does not provide any significant gain either.

The experimental results reported in Fig. 4(d) for “Catalunya” suggest that the users have a slightly different behavior when navigating on this image. The main difference compared with the previous images is that the users move more frequently downward than upward. More precisely, the probability of panning movements $X = \{$ down right | down | down left $\}$ is 22% against the 17% achieved for $X = \{$ up left | up | up right $\}$. Such a difference is only found in this image. It is caused due to the positioning of the first WOI, which is fixed by the server at the top of the image. This compels most users to reach their areas of interest moving downward. Also, the zooming operations carried out on this image have a slightly higher probability of being executed than that found in the previous three images. This may be caused due to the high spatial resolution available for this image, which reaches 10 meters per pixel at the highest resolution level. This may trigger the curiosity of the users to zoom in and out more frequently to explore the requested areas in more detail. As seen in Section V, a slight increment on performance is achieved for this image when probabilities $P(X)$ correspond to those reported in Fig. 4(d) (i.e., $P''(X) = \{7.7\% \mid 6.6\% \mid 6.2\% \mid 2.9\% \mid 12.6\% \mid 5.9\% \mid 7\% \mid 3.3\% \mid 25.7\% \mid 22.3\%\}$ in the movement order reported in the figure). We remark that these observations result in a slight change in the model that has little impact on the performance of the system. The user-navigation models employing $P'(X)$ and $P''(X)$ mainly describe the same underlying user behavior.

The user-navigation model determined in this analysis is valid for applications that transmit interactively remote sensing

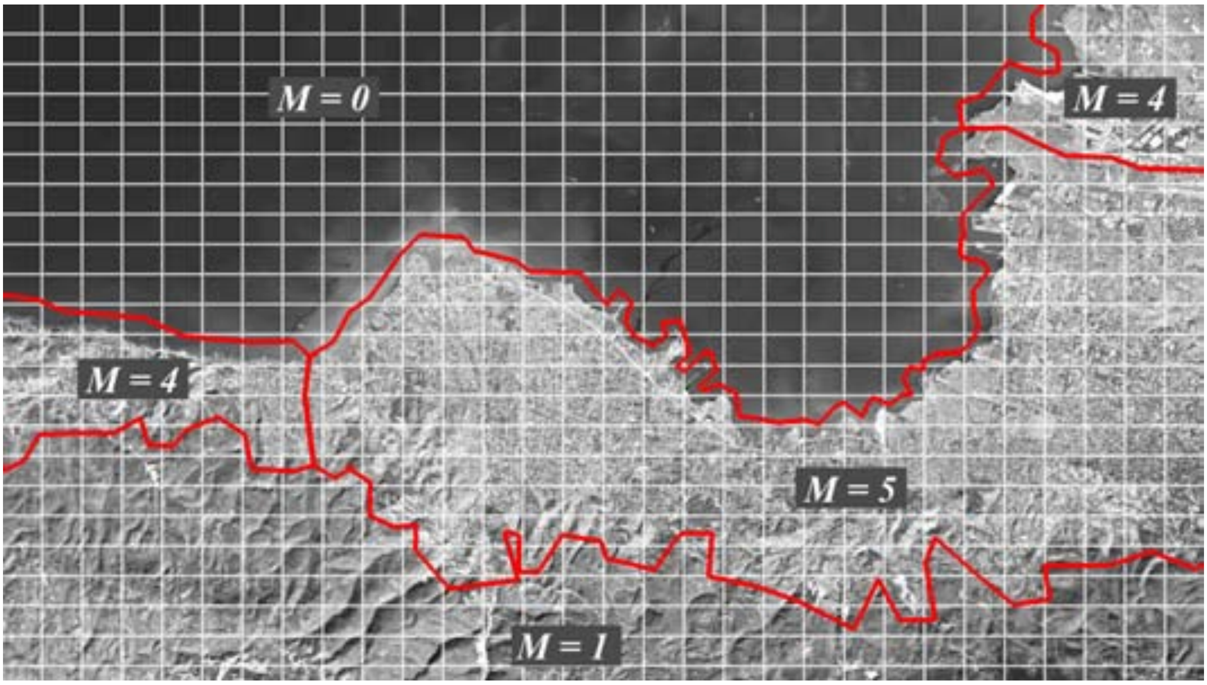


Fig. 5: Semantic map and precincts employed for the “Port-au-Prince city” image (see the details of this image in Section V). The semantic regions are delimited in red. The label on each region indicates its priority, which is denoted as M and is determined depending on the content of the image. The areas corresponding to the city center and its surroundings have the highest priorities, whereas the forest and the sea have lower priorities. The white lines forming the grid correspond to the JPEG2000 precincts of resolution level 3 (resolution level 0 and 5 are the smallest and the largest, respectively).

images of high resolution. Other applications that transmit images with different particularities and/or users behaving differently may require different models. The methodology described above can be employed in other scenarios to determine suitable user-navigation models for them.

B. Semantic map

The content of the image is also a good indicator for the probability of an area to be visited. This observation has been used recently by the remote sensing community with the aim to optimize the computational resources of a web map server [26]. To the best of our knowledge, such an approach has never been employed in the framework of JPIP. The main idea is to assign high probabilities to those movements that displace the WOI to image areas whose contents are more appealing to the user. This strategy is implemented in the JPIP proxy server as follows. First, the image is partitioned in regions containing different types of content. In a remote sensing image, these regions may correspond to cities, roads, seas, or deserted areas, for instance. As stated in [26], each type of region has a different chance to be visited. Cities are more commonly browsed by users than deserted areas, for example. So selected priorities are assigned to each region type. The higher the priority, the higher the chance that that image area is visited. Depending on how this operation is implemented, it may require the supervision of an expert in the field, or the use of other techniques such as classification [29]. It is important to employ accurate methods of supervision/classification since otherwise the performance of the system may be penalized.

The selected regions together with their priorities form the so-called *semantic map*. The semantic map is employed to obtain individual priorities for packets. As described above, packets $\mathcal{T}_{c,r,p,q}$ are quality increments corresponding to the precinct of component c, resolution level r, and spatial location p. Fig. 5 illustrates the relation between the semantic map and the precincts of one resolution level of an image employed in the experimental tests of the next section. As seen in the figure, each precinct may be contained in one or more semantic regions. When a precinct is fully contained in one region, the priority of that region is assigned to all packets of that precinct. When a precinct is partially contained in more than one region, the priority for packets of that precinct is computed as a weighted average among the regions to which it belongs. More precisely, probabilities $P(X)$ employed in Equation (5) are determined according to

$$P'''(X) = \frac{\delta''(\mathcal{T}_{c,r,p,q}, X)}{\sum_{X'} \delta''(\mathcal{T}_{c,r,p,q}, X')} \ , \tag{6}$$

with

$$\delta''(\mathcal{T}_{c,r,p,q}, X) = \begin{cases} \sum_{s} F''_{c,r,p,e} \cdot M_{c,r,p,e} & \text{if } \mathcal{T}_{c,r,p,q} \succ f(\mathcal{W}_k, X) \\ 0 & \text{otherwise} \end{cases} . \tag{7}$$

$\delta''(\cdot)$ is a function that returns the priority assigned through the semantic map to packet $\mathcal{T}_{c,r,p,q}$ when the user executes movement X. When the packet is related to WOI $\mathcal{W}_k$, this priority is computed in Equation (7) as $\sum_s F''_{c,r,p,e} \cdot M_{c,r,p,e}$. $M_{c,r,p,e}$, with $1 \leq e \leq E$, denotes the priorities of the E semantic regions that correspond to the precinct located at c, r, p. $F''_{c,r,p,e}$ accounts for the percentage of coefficients within the precinct located at c, r, p that are relevant to the semantic region e. Equation (6) computes $P(X)$ as the priority of movement X divided by the summation of priorities of all possible movements.

V. EXPERIMENTAL RESULTS

All the infrastructure used to evaluate the proposed method, including the client, the server, and the JPIP proxy server are implemented in our JPEG2000 Part 9 implementation CADI [30]. Experiments are carried out in a LAN with a channel capacity of 100 Mbps and a connection to the Internet shared by all clients with a channel capacity of 10 Mbps. The images employed in the experiments are two satellite images from GeoEye Inc. that were provided to help missions responding to the earthquake that struck Haiti in 2010. Our aim is that rescue teams in similar situations employ the proposed mechanisms to enhance their transmission infrastructure. The images cover areas of Port-au-Prince and its surroundings, are referred to as “Port-au-Prince airport” and “Port-au-Prince city”, and are 8-bit gray scale with size 32768×19456 and 31744×20480, respectively. The 8-bit gray scale Landsat image “Catalunya” utilized in Section IV-A is also employed in the experiments. It is provided by CREAF and has a size of 13561×13161. These images are compressed with JPEG2000 using the following coding parameters: 5 levels of irreversible CDF 9/7 wavelet transform, codeblock size of 64×64, restart coding variation, and 25 quality layers logarithmically spaced in terms of bitrate. The server adjusts on-the-fly the precinct size in all resolution levels according to the codeblock size to enhance interactivity options.

Results are obtained when four or five users browse an image simultaneously. Eight transmission strategies are evaluated along the following experiments:

1) *Client-server*: the image transmission is carried out without using any proxy infrastructure.
2) *Client-server with prefetching at the clients*: the image transmission is carried out without using any proxy infrastructure, and prefetching is employed individually at each client.
3) *Conventional JPIP proxy*: all data transmitted between the server and the clients are cached by a proxy that reuses already transmitted packets. This strategy is (almost) equivalent to that introduced in [20].
4) *Conventional JPIP proxy with prefetching at the clients*: prefetching is employed individually at each client and all data transmitted between the server and the clients are cached by a proxy.
5) *JPIP proxy with simple prefetching*: this strategy uses the proxy described in Section III but employing a uniform model of probabilities for the prefetching mode, i.e., $P(X) = 1/10 \; \forall \; X$. This strategy was introduced in our previous work [1]. Here, it serves to appraise the gain that is achieved when prefetching is carried out employing the probability models described in Section IV.
6) *JPIP proxy with prefetching based on the user-navigation model*: this strategy uses the proxy described in Section III with prefetching based on the user-navigation model described in Section IV-A. Employed probabilities are $P'(X)$ or $P''(X)$ as indicated.
7) *JPIP proxy with prefetching based on the user-navigation model and Quality of Service (QoS) considerations*: this strategy is as the previous one except for the consideration of clients that may have a higher priority during the prefetching mode of the proxy (see below).
8) *JPIP proxy with prefetching based on the semantic map*: this strategy uses the proxy described in Section III with prefetching based on the semantic map described in Section IV-B. Employed probabilities are $P'''(X)$.

The results are reported for these strategies when the clients execute their browsing sessions identically. Evidently, each client browse different areas of the image, but the same chain of movements is reproduced by each client when evaluating different strategies. The chain of movements selected for each user corresponds to typical browsing sessions exploring areas of an image. We allow users to perform up to 13 movements during a session that lasts 7 minutes approximately. This is a behavior typically observed in browsing sessions, as described in Section IV-A.

Fig. 6 reports the results achieved when four clients browse the “Port-au-Prince airport” image. The vertical axis of these and following figures is the quality of the retrieved WOI, in terms of Peak Signal to Noise Ratio (PSNR). The horizontal axis is the browsing session time. Only three representative WOIs selected for each client are reported in these figures for the sake of clarity. The WOIs correspond to three consecutive movements of the user and are selected to illustrate the behavior of each strategy. Similar results hold for the remaining –not reported– WOIs. To enhance the presentation, the horizontal axes of the figures are broken between WOIs. All plots starting from the same point, approximately, correspond to the same WOI retrieved by the client. WOIs are decoded at the end of each quality layer, hence, the quality achieved by all strategies when decoding the same WOI is equal, changing only the instant of time at which the layer is completely available at the client. Fig. 6 reports results for five of the strategies described before, namely, client-server, conventional JPIP proxy, JPIP proxy with simple prefetching, JPIP proxy with prefetching based on the user-navigation model, and JPIP proxy with prefetching based on the user-navigation model and QoS considerations.

In general, the WOIs requested by the clients at the beginning of their sessions are recovered similarly by all strategies since there are no data in the cache of the proxy. See, for instance, in Fig. 6(c) that all transmission strategies recover WOI 2 of client 3 at the same pace. As the browsing session advances, transmission strategies that employ proxy infrastructure recover WOIs more rapidly than the client-server strategy due to data already transmitted for other clients can

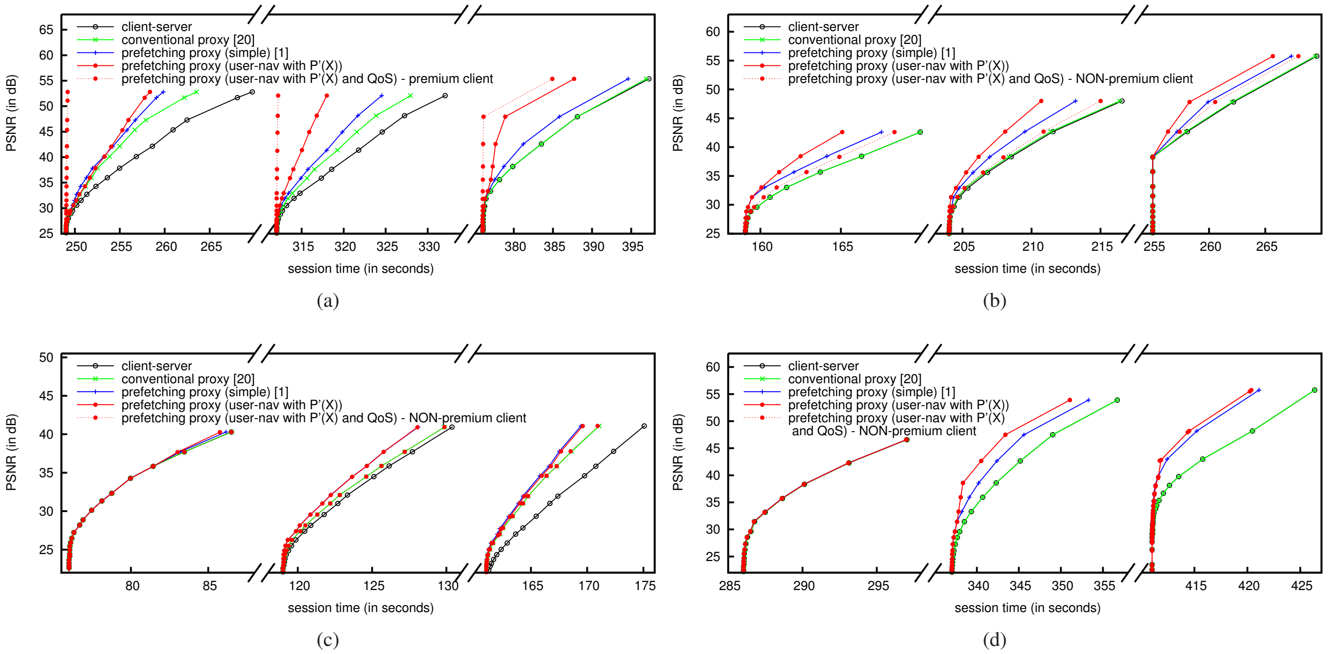


Fig. 6: Evaluation of the performance achieved by a proxy server with a prefetching strategy based on a user-navigation model compared to three other strategies. The image browsed is “Port-au-Prince airport”. The following WOIs are reported: (a) 5, 6, and 7 from client 1; (b) 5, 6, and 7 from client 2; (c) 2, 3, and 4 from client 3; and (d) 6, 7, and 8 from client 4.

be reused. This is seen in WOIs 3 and 4 requested by client 3 (Fig. 6(c)). Although it is not usual, in some occasions a WOI requested in the middle, or at the end, of a session is also recovered by all strategies similarly. See, for instance, WOI 6 requested by client 4 in Fig. 6(d). This happens only when the user moves to an area that was neither browsed by other clients nor prefetched by the proxy. As seen in Fig. 6, the most typical behavior is that the proxy using prefetching strategies based on the user-navigation model serves the WOI more rapidly than the other strategies. Note, for example, that the last WOI reported in Fig. 6(a) is delivered by our proxy to the client almost 10 seconds before all other transmission strategies. The differences between the conventional proxy and the proxies using prefetching become evident after the second minute of the session (not shown in the figure), approximately, since there have been enough idle moments to prefetch more data from the server. As seen in the figure, the differences are relevant for some WOIs (e.g., second and third WOI reported in Fig. 6(d)). It is also worth noting the improvement on performance achieved when the prefetching is carried out employing the user-navigation instead of uniform probabilities. For all WOIs reported in Fig. 6, to prefetch data employing the user-navigation model achieves equal or superior performance than when prefetching employs uniform probabilities. This demonstrates the superiority of the prefetching strategy based on the user-navigation model.

The proposed scheme to prefetch data also allows that one (group of) client(s) is served with a higher quality than the others. This mechanism is commonly referred to as QoS. In this context QoS is devised so that some of the JPIP clients have a higher priority during the prefetching mode than the others. The objective is to provide better responsiveness to some premium clients of the LAN. The framework deployed by the proposed JPIP proxy server allows the introduction of QoS by multiplying the probability $P(X)$ of the potential future WOIs requested by the client with premium service by a factor selected depending on the service that the client requires. See, for example, in Fig. 6(a) the results achieved when the probabilities of client 1 are multiplied by a factor of 4. The WOIs requested by this client are, in general, already in the cache of the proxy when it requires them, so they are delivered to the client almost instantly. Evidently, this degrades the responsiveness for some of the other clients because the proxy may not prefetch data for them. QoS strategies such as these may be useful in pay-per-service scenarios or emergencies in which a rescue team is in a critical situation.

The results achieved by the proxy server using the simple prefetching (i.e., that with uniform probabilities) are not reported in the following figures to avoid cluttering them, though similar results as those reported in Fig. 6 hold.

Fig. 7 reports the results achieved when five clients browse the “Catalunya” image. We recall that –only for this image– the first WOI transmitted by the server is situated at the top of the image. Results are reported for the proxy using the prefetching strategies based on the user-navigation model that employs probabilities $P'(X)$ and $P''(X)$. We recall that probabilities $P''(X)$ correspond to those reported in Fig. 4(d). Results for the the client-server strategy and the conventional proxy are also reported in this figure. Again, the experimental results indicate that proxy servers that employ prefetching

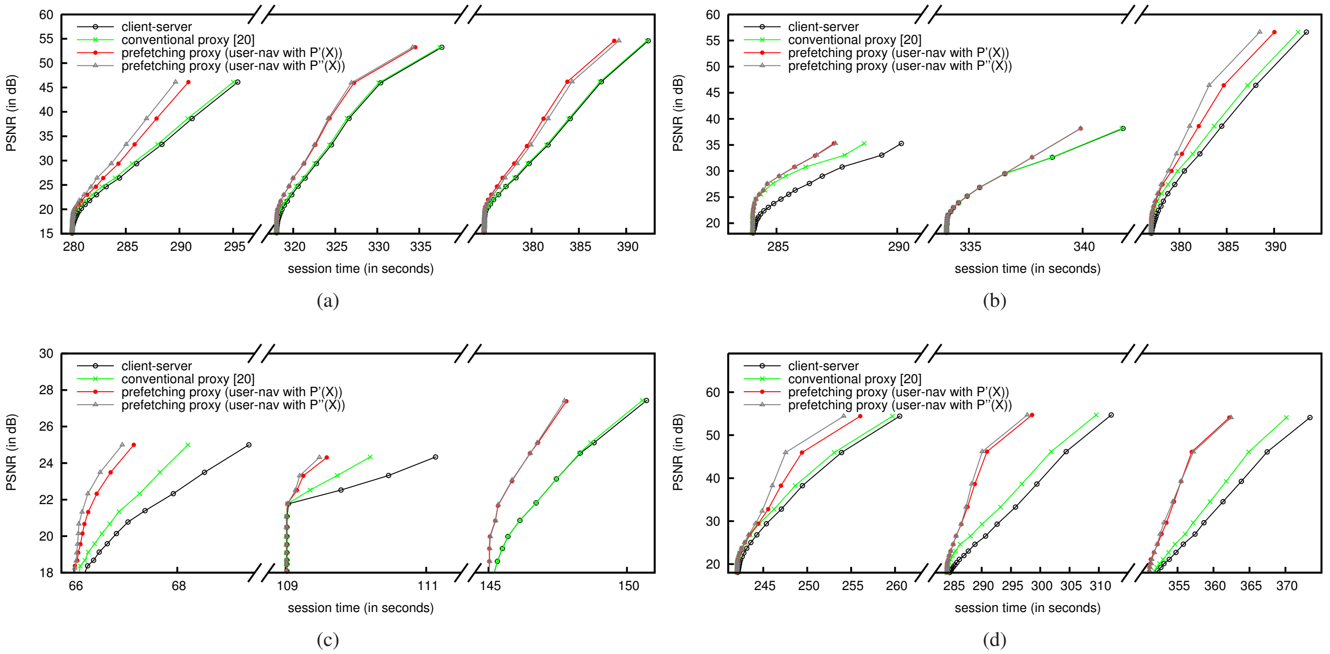
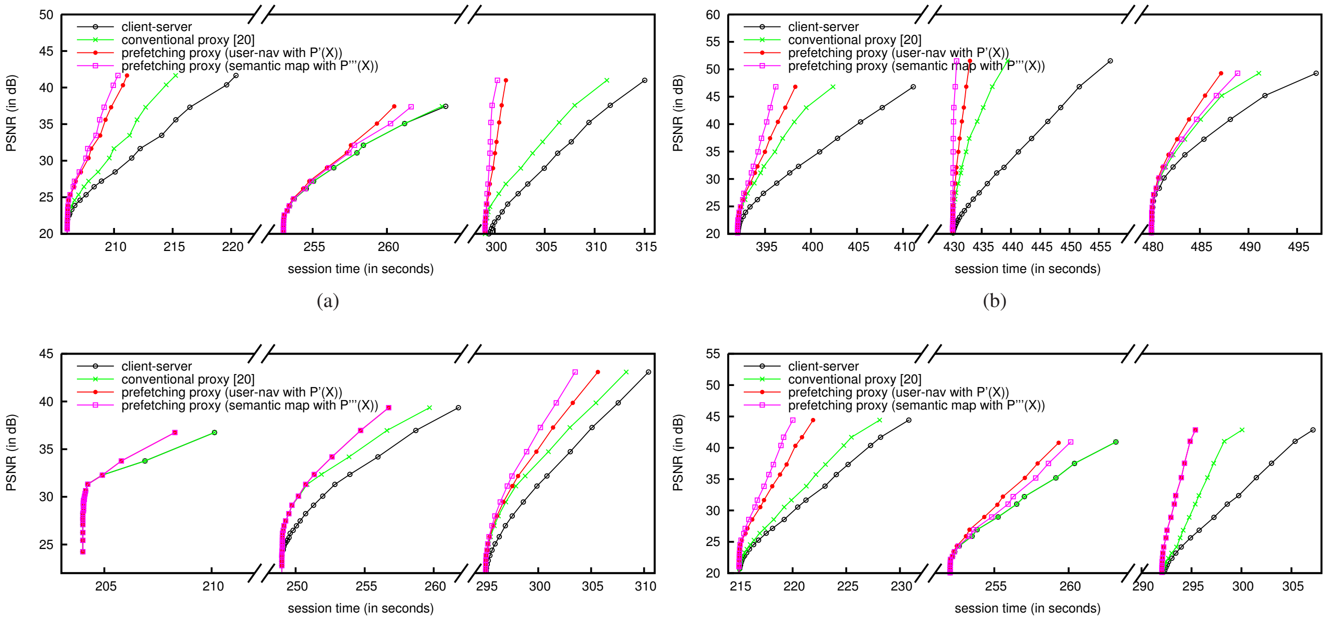


Fig. 7: Evaluation of the performance achieved by a proxy server with a prefetching strategy based on a user-navigation model extracted for the image “Catalunya” compared to three other strategies. The following WOIs are reported: (a) 8, 9, and 10 from client 1; (b) 7, 8, and 9 from client 2; (c) 2, 3, and 4 from client 4; and (d) 7, 8, and 9 from client 5.

strategies deliver WOIs to clients more rapidly than a conventional proxy server or a client-server strategy. The use of probabilities that have a major incidence on movements that go downward (i.e., $P''(X)$) permits the recovery of WOIs slightly faster than when probabilities of the user-navigation model are more generic (i.e., $P'(X)$). For the WOIs reported in this figure, the difference between these two prefetching strategies is two seconds, at most. These results suggest that the responsiveness of clients can be enhanced by using a specific probability model for images whose first WOI transmitted is not at the center of the image, though the gains achieved when doing so are modest.

Fig. 8 reports the results achieved when four clients browse the “Port-au-Prince city” image. The prefetching strategy based on the semantic map is deployed using the regions and priorities depicted in Fig. 5. This strategy is compared to the proxy that uses prefetching based on the user-navigation model with probabilities $P'(X)$, to a conventional proxy, and to a client-server strategy. The results achieved are similar to those reported previously. The proxy servers employing prefetching strategies achieve the best results. The prefetching strategy employing the semantic map achieves, on average, slightly better performance than that of the user-navigation model, though the differences are insignificant. For some WOIs, the use of probabilities $P'(X)$ achieves better results than the semantic map. These results indicate that, in this context, both the user-navigation model and the semantic map are appropriate models to predict with precision the next movements of the users. To the best of our knowledge, it is not possible to combine the semantic map with the user-navigation model to create a prefetching strategy that improves these results due to the probabilities given by both strategies would interfere.

Fig. 9 reports the network statistics obtained in the experiment of Fig. 8. Only two strategies are depicted to avoid cluttering the figure, namely, the conventional proxy server, and the proxy server with prefetching based on the semantic map. The figure reports the Megabytes (MB) transmitted in the uplink and in the downlink. The uplink is the connection between the proxy and the server (i.e., B in Fig. 1). It carries the requests done to the server. The proxy with prefetching transmits approximately 2 MB in total, whereas the conventional proxy transmits 0.35 MB. These differences are caused because the proxy with prefetching requests more data when the connection is idle. Despite these differences, the amount of information transmitted by the two strategies in the uplink is very low. The downlink is the connection between the server and the proxy (i.e., C in Fig. 1). It carries the codestream segments requested to server, so the amount of information transmitted in this link is much larger than that of the uplink. All the information transmitted in the downlink is kept in the cache of the proxy, so the size of the proxy cache is proportional to the data transmitted in the downlink. Again, the proxy with prefetching transmits more information than the conventional proxy because it utilizes idle times to retrieve more data. It is important to provide a large cache to the proxy to avoid removing data. In general, this is not an issue because the cache can be saved as a file in the hard drive of the proxy. The capacity of the cache in the client is not critical either because the client only receives and caches



(a)

(b)

(c)

(d)

Fig. 8: Evaluation of the performance achieved by a proxy server with a prefetching strategy based on a semantic map compared to three other strategies. The image browsed is “Port-au-Prince city”. The following WOIs are reported: (a) 6, 7, and 8 from client 1; (b) 10, 11, and 12 from client 2; (c) 6, 7, and 8 from client 3; and (d) 6, 7, and 8 from client 4.

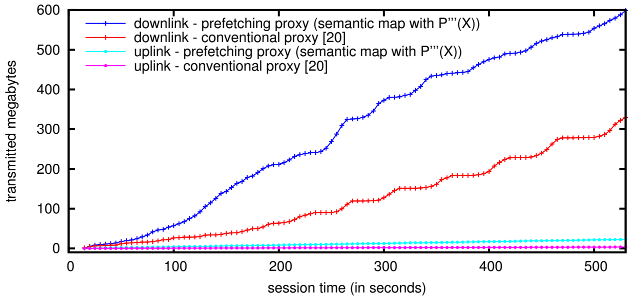


Fig. 9: Evaluation of the network statistics obtained in the same experiment as that of Fig. 8. The strategies evaluated are the conventional proxy and the proxy server with a prefetching strategy based on the semantic map.

those data corresponding to the requested WOIs. Furthermore, the cache of the client could be completely avoided by using a stateless connection between the client and the proxy.

To summarize all the results, Table I reports the average time required to transmit all layers from all WOIs requested from all clients, for the three images evaluated before. In addition to the other strategies discussed before, this table also reports the results achieved by the client-server strategy in which each client employs prefetching individually, and the strategy in which all clients prefetch individually and a conventional proxy server caches all transmitted data. On average for the three images, the response time required by the client-server strategy, the client-server strategy with prefetching at the clients, the conventional proxy, and the conventional proxy with prefetching at the clients is 11.8, 10.15, 9.31, and 7.6 seconds, respectively. The proxy server using prefetching strategies with the uniform model of probabilities achieves an average response time of 6.72 seconds, whereas the use of the user-navigation model decreases the response time to 6.19 seconds. The semantic-map based prefetching employed for the “Port-au-Prince city” image achieves virtually same results as those of the user-navigation model. These results indicate that the transmission strategies that enhance the most the responsiveness of the overall system are those that include a JPIP proxy server using prefetching strategies based on a user-navigation model or a semantic map. We remark that the average response time when using the proposed JPIP proxy server is nearly half as that needed by a client-server strategy, and approximately $1/3$ less than that needed by a conventional proxy. The use of prefetching at each client without any proxy achieves poor performance in these experiments, decreasing the responsiveness time of the clients in less than 1.2 seconds as compared to the average response time decrease of 5.6 seconds achieved by the proposed proxy employing the user-navigation model. The performance achieved when prefetching is carried out at each client can be improved using a conventional proxy server, though the achieved results do not outmatch those achieved when prefetching is carried out at the proxy.

VI. CONCLUSIONS

This paper considers the interactive browsing of large resolution images through the JPEG2000 Interactive Protocol

TABLE I: Evaluation of the average response time for all WOIs from all clients. Results are reported in seconds. Cells with a dash indicate that the corresponding transmission strategy does not apply on that image.

	Port-au-Prince			
	airport	city	Catalunya	*average*
conventional strategies				
client-server	15.45	12.53	7.42	*11.80*
client-server with prefetching at clients	13.88	10.74	5.82	*10.15*
conventional proxy [20]	13.50	8.73	5.70	*9.31*
conventional proxy with prefetching at clients	10.31	8.03	4.45	*7.60*
proposed strategies				
simple [1]	10.92	4.98	4.25	*6.72*
user-nav model $P'(X)$	9.87	4.65	4.04	*6.19*
user-nav model $P''(X)$	-	-	3.78	-
semantic map $P'''(X)$	-	4.63	-	-

(JPIP) when a collection of clients in a local area network (LAN) requests windows of interest (WOIs) from an image in a remote server. In this work, the capabilities of a JPIP proxy server located within the LAN are extended. Our main insight is to introduce prefetching strategies at the proxy to optimize the responsiveness of all clients simultaneously. Prefetching is carried out only during instants at which the connection between the server and the clients is idle. Key to maximize performance is to anticipate with precision the future movements of the users. Two strategies are proposed to do so. The first one is based on a user-navigation model extracted from logs collected by several image servers. The second strategy is based on the content, or the semantic, of the image. The proposed JPIP proxy server also reformulates the WOIs requested to the server so that the transmission of data between proxy and server is fully optimized. Experimental results carried out with earth observation images suggest that the proposed JPIP proxy server reduces the average response time of a conventional proxy server and a client-server strategy by $1/3$ and $1/2$, respectively. The lower response time achieved by the proposed strategy enhances the browsing experience of the end-user notably.

This work may be extended to images from other fields such as the medical or the video. Also, the models to predict future user movements might be extended to consider retrospective adaptive prefetching [31], or visual attention [32].

ACKNOWLEDGMENT

The authors thank our colleagues from CREAF for interesting discussions and suggestions, and the anonymous reviewers and associate editor for their comments and remarks.

REFERENCES

[1] J. L. Monteagudo-Pereira, F. Auli-Llinas, J. Serra-Sagrista, and J. Bartrina-Rapesta, “Smart JPIP proxy server with prefetching strategies,” in *Proc. IEEE Data Compression Conference*, Mar. 2010, pp. 99–108.

[2] J. L. Monteagudo-Pereira, F. Auli-Llinas, J. Serra-Sagrista, A. Zabala, J. Maso, and X. Pons, “Enhanced transmission of JPEG2000 imagery through JPIP proxy and user-navigation model,” in *Proc. IEEE Data Compression Conference*, Apr. 2012, pp. 22–31.

[3] *Information technology - JPEG 2000 image coding system - Part 1: Core coding system*, ISO/IEC Std. 15 444-1, Dec. 2000.

[4] S. Deshpande and W. Zeng, “Scalable streaming of JPEG2000 images using hypertext transfer protocol,” in *Proc. ACM International Conference on Multimedia*, vol. 9, Oct. 2001, pp. 372–381.

[5] D. Taubman, “Remote browsing of JPEG2000 images,” in *Proc. IEEE International Conference on Image Processing*, Sep. 2002, pp. 229–232.

[6] J. Li and H.-H. Sun, “On interactive browsing of large images,” *IEEE Trans. Multimedia*, vol. 5, no. 4, pp. 581–590, Dec. 2003.

[7] E. A. Politou, G. P. Pavlidis, and C. Chamzas, “JPEG2000 and dissemination of cultural heritage over the internet,” *IEEE Trans. Image Process.*, vol. 13, no. 3, pp. 293–301, Mar. 2004.

[8] R. Rosenbaum and H. Schumann, “JPEG2000-based image communication for modern browsing techniques,” in *Proc. SPIE International Conference on Image and Video Communications and Processing*, vol. 5685, Jan. 2005, pp. 1019–1030.

[9] *Information technology - JPEG 2000 image coding system Part 9: Interactivity tools, APIs and protocols*, ISO/IEC Std. 15 444-9, Dec. 2005.

[10] K. Krishnan, M. W. Marcellin, A. Bilgin, and M. S. Nadar, “Efficient transmission of compressed data for remote volume visualization,” *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1189–1199, Sep. 2006.

[11] V. J. Tuominen and J. Isola, “Linking whole-slide microscope images with DICOM by using JPEG2000 interactive protocol,” *SPRINGER Journal of Digital Imaging*, vol. 23, no. 4, pp. 454–462, Apr. 2010.

[12] J. Monteagudo-Pereira, J. Bartrina-Rapesta, F. Auli-Llinas, J. Serra-Sagrista, A. Zabala, and X. Pons, “Interactive transmission of spectrally wavelet-transformed hyperspectral images,” in *Proc. SPIE International Conference on Satellite Data Compression, Communication and Processing*, vol. 7084, Aug. 2008, pp. 708 405,1–12.

[13] D. Muller, B. Fleck, G. Dimitoglou, B. W. Caplins, D. E. Amadigwe, J. P. G. Ortiz, B. Wamsler, A. Alexanderian, V. K. Hughitt, and J. Ireland, “JHelioviewer: Visualizing large sets of solar images using JPEG 2000,” *IEEE Comput. Sci. Eng.*, vol. 11, no. 5, pp. 38–47, Sep. 2009.

[14] F. Auli-Llinas and D. Taubman, “Optimal delivery of motion JPEG2000 over JPIP with block-wise truncation of quality layers,” in *Proc. IEEE International Conference on Image Processing*, Oct. 2008, pp. 2856–2859.

[15] A. T. Naman and D. Taubman, “JPEG2000-Based scalable interactive video (JSIV),” *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1435–1449, May 2011.

[16] ——, “JPEG2000-Based scalable interactive video (JSIV) with motion compensation,” *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2650–2663, Sep. 2011.

[17] D. Taubman and R. Rosenbaum, “Rate-distortion optimized interactive browsing of JPEG2000 images,” in *Proc. IEEE International Conference on Image Processing*, vol. 3, Sep. 2003, pp. 765–768.

[18] A. Descampe, C. D. Vleeschouwer, M. Iregui, B. Macq, and F. Marques, “Prefetching and caching strategies for remote and interactive browsing of JPEG2000 images,” *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1339–1354, May 2007.

[19] J. P. G. Ortiz, V. G. Ruiz, M. F. Lopez, and I. Garcia, “Interactive transmission of JPEG2000 images using web proxy caching,” *IEEE Trans. Multimedia*, vol. 10, no. 4, pp. 629–636, Jun. 2008.

[20] L. Lima, D. Taubman, and R. Leonardi, “JPIP proxy server for remote browsing of JPEG2000 images,” in *Proc. IEEE Workshop on Multimedia Signal Processing*, Oct. 2008, pp. 844–849.

[21] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image compression fundamentals, standards and practice*. Norwell, Massachusetts 02061 USA: Kluwer Academic Publishers, 2002.

[22] F. Auli-Llinas, J. Bartrina-Rapesta, and J. Serra-Sagrista, “Enhanced JPEG2000 quality scalability through block-wise layer truncation,” *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1–11, 2010, article ID 803542.

[23] 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.

[24] T. Richter, “Compressing JPEG 2000 JPIP cache state information,” in *Proc. IEEE Data Compression Conference*, Apr. 2012, pp. 13–21.

[25] D. H. Lee, J. S. Kim, S. D. Kim, K. C. Kim, Y.-S. Kim, and J. Park, “Adaptation of a neighbor selection markov chain for prefetching tiled web GIS data,” in *Proc. International Conference on Advances in Information Systems*, Oct. 2002, pp. 213–222.

[26] S. Quinn and M. Gahegan, “A predictive model for frequently viewed tiles in a web map,” *Transactions in GIS*, vol. 14, no. 2, pp. 193–216, Apr. 2010.

[27] D. S. Taubman, “Localized distortion estimation from already compressed JPEG2000 images,” in *Proc. IEEE International Conference on Image Processing*, Oct. 2006, pp. 3089–3092.

[28] X. Pons. (2004) MiraMon. Geographic Information System and Remote Sensing software. Centre de Recerca Ecolgica i Aplicacions Forestals, CREAF. Bellaterra. ISBN: 84-931323-5-7. [Online]. Available: http://opengis.uab.cat/wms/satcat/

[29] G. Camps-Valls and L. Bruzzone, “Kernel-based methods for hyperspectral image classification,” vol. 43, no. 6, pp. 1351–1362, Jun. 2005.

[30] J. L. Monteagudo-Pereira. (2012) CADI. [Online]. Available: http://www.gici.uab.cat/CADI

[31] S. Yesilmurat and V. Isler, “Retrospective adaptive prefetching for interactive web GIS applications,” *SPRINGER Geoinformatica*, vol. 16, no. 3, pp. 435–466, Jul. 2012.

[32] Z. Li, S. Qin, and L. Itti, “Visual attention guided bit allocation in video compression,” *ELSEVIER Image and Vision Computing*, vol. 29, pp. 1–14, Jan. 2011.

José Lino Monteagudo-Pereira recieved the B.E. and M.S. degrees in 2002 and 2004, respectively, both in Telecommunications Engineering from the Universidad de Vigo (Spain). From October 2002 to August 2004 he was with the faculty of the School of Telecommunications Engineering, Universidad de Vigo, as a researcher on image processing and multimedia. He joined the Group on Interactive Coding of Images at the Universitat Autònoma de Barcelona (Spain) in September 2004 where he is currently working towards his Ph.D. degree. Since January 2009 he has been with the Ministry of Finance and Public Administrations (Spain) as a Network and Computer Systems Administrator. His research interests lie in the areas of image compression and interactive image transmission.

Francesc Aulí-Llinàs (S’2006-M’2008) is a Fellow of the Ramón y Cajal program funded by the Spanish Government. He is currently in the Department of Information and Communications Engineering, at the Universitat Autònoma de Barcelona (Spain). He received the B.Sc. and B.E. degrees in Computer Management Engineering and Computer Engineering in 2000 and 2002, respectively, both from the Universitat Autònoma de Barcelona (Spain), and for which he was granted with two extraordinary awards of Bachelor (awards given to the first students of the promotion). In 2004 and 2006 he respectively received the M.S. degree and the Ph.D. degree (with honors), both in Computer Science from the 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 (USA). He is the main developer of BOI, a JPEG2000 Part 1 implementation that was awarded with a free software mention from the Catalan Government. 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.

Joan Serra-Sagristà (S’97-M’05-SM’11) received the Ph.D. degree in Computer Science from Universitat Autònoma de Barcelona (UAB), Spain, in 1999. He is currently an Associate Professor at the Department of Information and Communications Engineering, UAB. From September 1997 to December 1998, he was at University of Bonn, Germany, funded by DAAD. His current research interests focus on data compression, with special attention to image coding for remote sensing and telemedicine applications. He serves as Associate Editor of IEEE Trans. on Image Processing and as Program Committee co-chair for IEEE Data Compression Conference. He has co-authored over one hundred publications. He was the recipient of the Spanish Intensification Young Investigator Award in 2006.

Chapter 4

CADI Software

CADI Software [45] is an implementation of the JPEG2000 Interactive Protocol (JPIP) [27] developed as the practicum part of this thesis. The design of the application has been carried out using UML [56] and, with the aim of achieving a multi-platform application, Java has been chosen as the programming language. The source code of CADI Software and its documentation is under the General Public Licence (GPL) [57], allowing its free use, modification and distribution. It can be found on the GICI web page (http://gici.uab.es/CADI) and on Sourceforge as the CADI Software project (http://sourceforge.net/projects/cadi).

CADI Software implements and supports all features of the *Profile 0: Basic Communication* defined by the JPIP FPDAM4 [58]. In addition, it provides support for several capabilities of the *Profile 1: Enhanced Communications*.

Figure 4.1 depicts the architecture and main design of CADI Software, which is composed of four applications following a client-server model: the graphical user interface *CADIViewer*; the JPIP client *CADIClient*; the JPIP server *CADIServer*; and the JPIP proxy *CADIProxy*.

CADIViewer

This application is a graphical user interface to display the transmitted images. Users request images hosted in a remote server and, as in any graphical image viewer, they can browse over the images employing zoomings, pannings, or adjusting the image

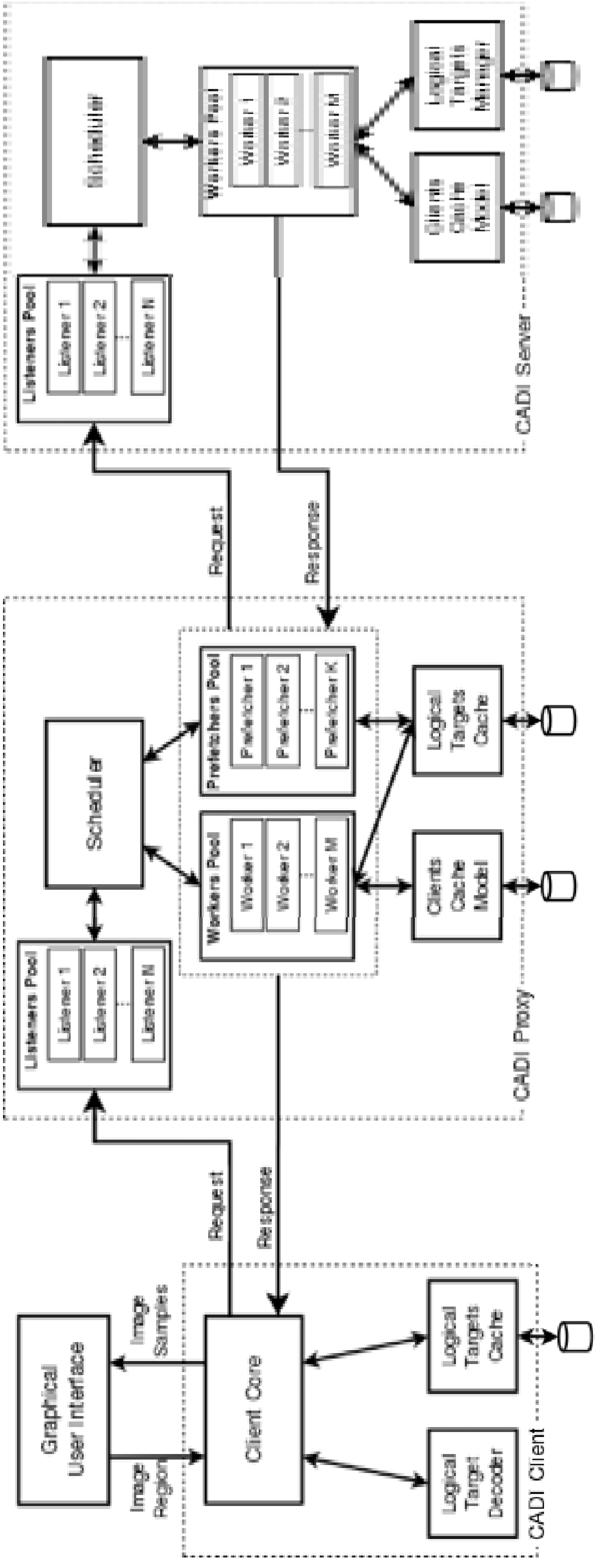


Figure 4.1: Architecture of CADI Software.

quality. It is worth noting that the viewer can be easily replaced by another application to fulfill specific requirements of different scenarios because it only requests a spatial area and quality of the image and receives the raw image that is finally displayed.

CADIClient

It is the application that implements the client of the JPIP protocol. It represents the middleware between the Graphical User Interface (GUI) and the JPIP server. It receives a request for a specific area of the image from the GUI and it forwards the request to the JPIP server if the data available in the *client cache* do not fulfill the request. *CADIClient* is composed of a main module, the *Client core*, and two auxiliary modules, the *Logical Targets Cache* and the *Logical target decoder*. The *Client core* manages the interface with the GUI, the communications with the JPIP proxy or server, and the interconnections between the auxiliary modules. The *Logical Targets Cache* implements a cache for storing all data delivered by the server. The *Logical target decoder* is a module addressed to decode the compressed codestream. The *Logical Targets Cache* can be disabled and its memory size restricted, allowing a light-weight version for constrained resource devices.

CADIServer

This application implements the server of the JPIP protocol. It listens to requests from clients and sends their responses. The high-level module design of the CADIServer is composed of five modules. All clients' requests are received by the *Listeners Pool* that gathers them in a queue, allowing the prioritization of some client requests over others. When the *Scheduler* receives clients' requests from the *Listeners Pool*, it parses and assigns them to a *Worker* of the *Workers Pool*, where prioritization policies can be applied. Then, the *Worker* module carries out the processing of the client requests, building the response with information about the image and/or segments of the compressed codestream. Then, the response is delivered to the client. To perform all these tasks, the CADIServer delegates specific tasks to two auxiliary modules: the

Clients Cache Model and the *Logical Targets Manager*. The former is responsible for keeping a model of the clients' cache status so that the responses to the clients can be optimized avoiding transmission of redundant data. The *Logical Targets Manager* plays an important role in managing the images hosted in the JPIP server. On the one hand, it defines an interface for managing information about the image and accessing to segments of the compressed codestream from different image coding systems. On the other hand, when an image is requested by a client, it is indexed once and its indices are kept in memory to minimize disk accesses and to reduce the response time.

It is worth nothing that the *Scheduler* is the core of the CADIServer because it supervises the *Listeners Pool* and *Workers Pool* tasks, it can adjust the size of the pool on demand, and it is able to apply a prioritization policy in the clients' requests.

CADIProxy

The *CADIProxy* is a JPIP proxy. As any proxy, it has two interfaces, one to deal with JPIP servers and another to deal with JPIP clients. After receiving a request for a spatial area of the image from a client, it parses the request and checks if there are enough data available in the *Logical Targets Cache* to fulfill the request. If so, it delivers the segments of the codestream belonging to the requested area. If data in the cache do not correspond to the requested region of the image, it forwards the request to the server, stores the response from the server in the cache and then it delivers the segments of the codestream that satisfy the client's request. If only a part of the data is available in the cache, it delivers the available data to the client, and, at the same time, it requests the remainder data from the server. Moreover, the CADIProxy includes prefetching capabilities that downloads segments of the codestream anticipating potential future requests of clients.

The architecture is similar to that of the server except that it has a new module, the *Prefetchers Pool*, which is addressed to the prefetching capabilities. This module, if enabled, is only active when all client's requests can be responded with data available in the cache, and so, the network connection between the proxy and the server is idle.

Chapter 5

Conclusions

5.1 Summary

A growing amount of the Internet traffic corresponds to multimedia content and, in particular, to the transmission of images. Although many different types of images are transmitted, a common trend nowadays is to use high resolution images, which results in an increase of the size of the transmitted data, and in the interactive browsing of images. Nowadays, image transmission systems are gaining attention because they reduce transmission costs and achieve a more fluent navigation. In this regard, the JPEG2000 Interactive Protocol is an outstanding client-server protocol that achieves an efficient transmission of images. Furthermore, the deployment of proxy servers along the network infrastructure reduces the bandwidth usage and prefetching strategies help to improve the fluency of the navigation.

The work carried out in this thesis falls within the scope of evaluating the interactive transmission of spectrally-wavelet transformed hyperspectral images and the development of JPIP proxy servers that incorporate prefetching strategies.

Hyperspectral images are characterized by high spatial and spectral resolution, so compression must be used to improve their storage and transmission. Image coding systems commonly apply a spectral decorrelator to remove redundancies across the spectral domain, thereby achieving better coding performance. The most common spectral decorrelators used are the Karhunen-Loève Transform (KLT) and the

Discrete Wavelet Transform (DWT). The KLT presents a superior coding performance than the DWT, but its main drawbacks are high computational costs, the need of high memory resources, and the lack of scalability.

When a spectral decorrelator is applied, the component scalability is reduced because there is not a one-to-one relationship between the transformed components and the original ones. As the number of decomposition levels increases or the kernel gets larger, the redundancies among the spectral dimension are reduced at the expense of poorer component scalability. Then, remote browsing of images may be affected because more components than those to be displayed might be needed for their recovery. This issue is addressed in Chapter 2, which evaluates the impact of the spectral transform on interactive transmissions.

The influence of the DWT parameters, kernel and number of decomposition levels in an interactive transmission context were presented in [28]. The main conclusion is that when a group of components, or the whole image, is required, the transmission is more efficient if the image has been spectrally transformed. However, when only one component, or a reduced group of them, is requested, spectrally transformed images have to deliver several groups of transformed components, which reduces the transmission efficiency. The number of transformed components needed to recover a range of image components depends on the kernel length and the number of decomposition levels, so a trade-off between coding performance and efficiency has to be achieved in an interactive transmission scenario.

On the other hand, as previously noted, new architectures and protocols allow users to access images hosted in remote servers without having to download the whole image. This scenario requires networks with high bandwidth to achieve acceptable response times and a fluent navigation. In order to alleviate the latency of the network, proxy infrastructures and prefetching techniques that predict and fetch data in anticipation of the user requests may be employed.

Chapter 3 tackles the deployment of prefetching strategies in JPIP proxy servers within an interactive JPEG2000 scenario that includes clients within a local area network. In this scenario, the clients share the same Internet connection to browse images hosted in remote servers. We propose the deployment of JPIP proxy servers in the local area network, as a means to reduce the transmission times and network traffic. The inclusion of prefetching strategies in the proxy enhances the responsiveness of the system. The most relevant aspect of the proposed prefetching mechanisms is that prefetching is carried out by the proxy only when the connection between the clients and the server is idle. Therefore, from the Internet link point of view, it works as a statistical time division multiplexer where traffic from the clients has the highest priority whereas the traffic from prefetching does not have a guaranteed bandwidth.

This thesis provides three new prefetching strategies to reduce the response time and to enhance the responsiveness. The first approach, referred to as simple prefetching [53], proposes a simple prediction algorithm based on the assumption that all user's movements have the same probability to be performed. A more accurate prediction model, referred to as user-navigation model [54], that is based on the behavior of users is our second approach. The last strategy, referred to as semantic-map model [55], proposes the use of the semantic information of the image to predict the next movements of the user.

The simple prefetching is the simplest approach since it neither takes into account the behavior of the users nor the nature, or the properties, of the image. Its main advantage is that it can be applied to any JPIP proxy server. Contrarily, the user-navigating and semantic-map models consider the behavior of the users when browsing images and the contextual information of the image, respectively. Since the probabilities for the user-navigation model are obtained from user browsing sessions, proper functioning is only guaranteed when it is applied to the same or similar scenario for which the probabilities has been obtained. Regarding the semantic-map model, it depends on the image information, so the map must be computed for each image.

Table 5.1 reports a summary of the average times for the three proposed prefetching strategies compared to a simple client-server and a conventional proxy architectures when a pool of clients perform a browsing session over three remote sensing images. On average, the three proposed prefetching mechanisms outperforms the client-server and conventional proxy, mainly because the conventional strategies do

Table 5.1: Summary of the average response time for a browsing session from all clients over three remote sensing images. Results are reported in seconds. Cells with a dash indicate that the corresponding transmission strategy does not apply on that image.

	Port-au-Prince			
	airport	city	Catalunya	*average*
client-server	15.45	12.53	7.42	*11.80*
conventional proxy [39]	13.50	8.73	5.70	*9.31*
simple prefetching [53]	10.92	4.98	4.25	*6.72*
user-navigation [54]	9.87	4.65	4.04	*6.19*
semantic map [55]	-	4.63	-	-

not incorporate prediction mechanisms. Among the three proposed strategies, the user-navigation and the semantic-map models yield a better responsiveness for the overall system due to the use of accurate prediction models. Nevertheless, it is worth noting that the response time achieved by the simple prefetching is also competitive.

To summarize, if images have context information and the semantic map can be computed, prefetching with semantic map should be used. If the semantic map can not be computed, a user-navigation model should be used when it is available for that scenario. If these two approaches are not feasible, the simplest prefetching strategy can always be used.

5.2 Future work

The research presented in this thesis has been mainly focused on preemptive strategies applied to JPIP proxy infrastructure. Following this line of work, there are several directions in which the research can be extended.

All three prefetching strategies proposed in this work have been tested on corpus of images from the remote sensing field. The first and more immediate line of investigation is the extension of these prefetching strategies to other corpus of images. The simplest prefetching approach might achieve similar results to those of the remote sensing scenario due to the algorithm does not depend on neither the image nor the scenario. However, the user-navigation and the semantic-map models depend on the scenario and the image content, respectively, so both models are not easily extrapolated to other images or scenarios.

Another extension of the prefetching strategies to investigate is an adaptive prefetching based on the users' history. Since the JPIP proxy logs the users' activity, it could be feasible to build a prefetching algorithm depending on the available users for a period of time or, even, the algorithm could be optimized for certain clients (e.g., premium clients).

A second future line of research is to analyze the mutual influence of the prefetching strategy and the cache replacement policy. Nowadays, storage-space size has significantly increased and storage costs have reduced, but the disk space is still limited. Therefore, any JPIP proxy server should implement a cache replacement policy. This policy could interfere with the prefetching strategy producing an abnormal functioning because the replacement policy may remove data from the cache that are prioritized by the prefetching strategy. Thus, the cache replacement policy and the prefetching strategy should work coordinately.

It is also interesting whether the prefetching strategy proposed in this work could be applied to Motion JPEG2000. Motion JPEG2000 considers the use of JPEG2000 for sequences of images. Since there are many differences between a sequence of images and a still image, it is a challenge to extend the prefetching mechanisms proposed in this work to Motion JPEG2000.

Another, less immediate, line of research is the extension of the prefetching strategies to JPIP proxy servers within a P2P network. In this case, the prefetching algorithm should be fully adaptive in the event of changes in the environment and, moreover, its convergence rate must be as short as possible.

Besides the scientific contribution, this thesis provides an open-source implementation of the JPIP protocol. Following this technological line, there are some extensions of the implementation for future work. The first one is the developing of several features to achieve full support for all the *Profile 1: Enhanced Communications* capabilities, which allow to test new experiments and would increase the interoperability between the available implementations. Moreover, Richter et al. [59] consider Profile 1 will be the most widespread and the Profile 0 is only a bare-bone feature set for very limited applications.

It would be interesting to obtain a new version of the client that could be fully integrated within a web browser. This line of work would add value to the implementation because it would expand the number of potential users. And last, and also less immediate, is to provide support for Motion JPEG2000. In this way, the aforementioned extension of the prefetching strategies for Motion JPEG2000 could be tested.

Appendix A

Acronyms

DWT Discrete Wavelet Transform

GPL General Public Licence

GUI Graphical User Interface

HTTP Hypertext Transfer Protocol

HTTP/1.1 Hypertext Transfer Protocol

IPP Internet Imaging Protocol

JPIK JPeg2000 Interactive Kakadu

JPIP JPEG2000 Interactive Protocol

JPP JPIP Precinct

JPT JPIP Tile-part

KLT Karhunen-Loève Transform

TCP Transmission Control Protocol

Vmedia virtual media access protocol

WOI Window of Interest

Bibliography

[1] J. Serra-Sagrista and F. Auli-Llinas, *Computational Intelligence For Remote Sensing*. USA: Springer, 2008, ch. Remote Sensing Data Compression , pp. 27 61.

[2] P. Kulkarni, A. Bilgin, M. W. Marcellin, J. C. Dagher, J. H. Kasner, T. J. Flohr, and J. C. Rountree, *Hyperspectral Data Compression.* USA: Springer, 2006, ch. Compression of Earth Science Data with JPEG2000 , pp. 347 378.

[3] J. P. Kerekes and J. R. Scott, *Hyperspectral Data Exploitation: Theory and Applications*. Hoboken, NJ, USA: John Wiley & Sons Inc., 2007, ch. Hyperspectral Imaging Systems , pp. 19 46.

[4] R. Choplin, Picture Archiving and Communication Systems: an overview, *Radiographics*, vol. 12, pp. 127 129, Jan. 1992.

[5] Digital Image and Communication in Medicine, DICOM, Oct. 2012. [Online]. Available: http://medical.nema.org/

[6] GMES and Earth Observation combined with Position based Image and sensor Communications Technology for Universal Rescue, Emergency and Surveillance (GEO-PICTURES), European Union Framework Program VII, Space Project FP7- 242390, 2010. [Online]. Available: http://www.geo-pictures.eu/

[7] D. Muller, B. Fleck, G. Dimitoglou, B. W. Caplins, D. E. Amadigwe, J. P. García Ortiz, B. Wamsler, A. Alexanderian, V. K. Hughitt, and J. Ireland, JHelioviewer: Visualizing Large Sets of Solar Images Using JPEG 2000, *IEEE Computing in Science and Engineering*, vol. 11, no. 5, pp. 38 47, Sep. 2009.

[8] *Information technology - JPEG 2000 image coding system - Part 1: Core coding system*, ISO/IEC Std. 15 444-1, Dec. 2000.

[9] D. Taubman, High performance scalable image compression with EBCOT, *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158 1170, 2000.

[10] W. Pennebaker and J. Mitchell, *JPEG still image data compression standard.* New York: Van Nostrand Reinhold, 1993.

[11] ISO/IEC, Digital compressing and coding for continuous-tone still images, International Standard Organization / International Electrotechnical Commission, Tech. Rep. ISO/IEC 10918-1, 1994.

[12] CompuServe Incorporated, Graphics Interchange Format (tm): A standard defining a mechanism for the storage and transmission of raster-based graphics information, http://www.w3.org/Graphics/GIF/spec-gif87.txt.

[13] X. Pons, MiraMon, Geographic Information Systems and Remote Sensing software. Center for Ecological Research and Forestry Applications, (CREAF). Bellaterra, 1994, ISBN: 84-931323-5-7.

[14] I3A - The International Imaging Industry Association, FlashPix File Format, http://www.i3a.org/i_flashpix.html.

[15] Internet Imaging Protocol, I3A - The International Imaging Industry Association, http://www.i3a.org/i_iip.html.

[16] LizardTech, http://www.lizardtech.com.

[17] Earth Resource Mapper, http://www.ermapper.com.

[18] S. Deshpande and W. Zeng, Scalable Streaming of JPEG2000 Images using Hypertext Transfer Protocol, vol. 9, International Multimedia Conference. New York, USA: ACM Press, 2001, pp. 372 381.

[19] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, Hyper-Text Transfer Protocol HTTP/1.1, World Wide Web

Consortium and the Internet Engineering Task Force, Tech. Rep., 1999, rFC 2616.

[20] S. T. Lian and T.-S. Chan, A bandwidth effective streaming of JPEG2000 images using hypertext transfer protocol, in *Proc. IEEE International Conference on Multimedia and Expo. 2002. ICME 02*, vol. 1, 2002, pp. 525 528.

[21] J. P. García Ortiz, V. González Ruiz, and I. Garcia, Improving Web proxy caching on browsing JPEG2000 remote images with JPIP, vol. 1. WSEAS Transactions on Information Science and Application, 2004, pp. 167 172.

[22] J. P. García Ortiz, V. González Ruiz, M. F. López, and I. García, Interactive Transmission of JPEG2000 Images Using Web Proxy Caching, *IEEE Transactions on Multimedia*, vol. 10, no. 4, pp. 629 636, 2008.

[23] J. Li and H. H. Sun, On Interactive Browsing of Large Images, *IEEE Transactions on Multimedia*, vol. 5, no. 4, pp. 581 590, 2003.

[24] D. Taubman, Remote browsing of jpeg2000 images, in *Proceedings of IEEE International Conference on Image Processing 2002*, vol. 1. IEEE, September 2002, pp. 229 232.

[25] D. S. Taubman, Proposal and Implementation of JPIP (JPEG2000 Internet Protocol) in Kakadu V3.3, http://www.kakadusoftware.com, Aug. 2002.

[26] Joint Photographic Experts Group (JPEG), http://www.jpeg.org.

[27] *Information technology - JPEG 2000 image coding system - Part 9: Interactivity tools, APIs and protocols*, ISO/IEC Std. 15 444-9, Dec. 2005.

[28] J. Monteagudo-Pereira, J. Bartrina-Rapesta, F. Auli-Llinas, J. Serra-Sagrista, A. Zabala, and X. Pons, Interactive transmission of spectrally wavelet-transformed hyperspectral images, in *Proc. SPIE International Conference on Satellite Data Compression, Communication and Processing*, vol. 7084, Aug. 2008, pp. 708 405,1 12.

[29] K. Krishnan, M. W. Marcellin, A. Bilgin, and M. S. Nadar, Efficient Transmission of Compressed Data for Remote Volume Visualization, *IEEE Transactions on Medical Imaging*, vol. 25, no. 9, pp. 1189 1199, Sep. 2006.

[30] V. J. Tuominen and J. Isola, Linking Whole-Slide Microscope Images with DICOM by Using JPEG2000 Interactive Protocol, *Journal of Digital Imaging: The Official Journal of the Society for Computer Applications in Radiology*, vol. 23, no. 4, pp. 454 462, Apr. 2010.

[31] F. Auli-Llinas and D. Taubman, Optimal delivery of Motion JPEG2000 over JPIP with block-wise truncation of quality layers, in *Proc. IEEE International Conference on Image Processing*, Oct. 2008, pp. 2856 2859.

[32] A. T. Naman and D. S. Taubman, JPEG2000-Based Scalable Interactive Video (JSIV), *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1435 1449, May 2011.

[33] A. T. Naman and D. Taubman, JPEG2000-Based Scalable Interactive Video (JSIV) With Motion Compensation, *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2650 2663, Sep. 2011.

[34] T. Richter, Compressing JPEG 2000 JPIP Cache State Information, in *Data Compression Conference (DCC)*, Apr. 2012, pp. 13 21.

[35] D. Taubman and R. Rosenbaum, Rate-distortion optimized interactive browsing of JPEG2000 images, in *Proceedings of IEEE International Conference on Image Processing 2003*, vol. 3, International Multimedia Conference. Barcelona: IEEE, September 2001, pp. 765 768 vol.2.

[36] M. Moshfeghi and J. Ta, Efficient image browsing with JPEG2000 internet protocol, vol. 5371. Proc. of SPIE Medical Imaging 2004: PACS and Imaging Informatics, 2004, pp. 31 42.

[37] A. Descampe, C. D. Vleeschouwer, M. Iregui, B. Macq, and F. Marques, Prefetching and Caching Strategies for Remote and Interactive Browsing of

JPEG2000 Images, *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1339 1354, May 2007.

[38] J. P. García Ortiz, V. González Ruiz, I. García, D. Muller, and G. Dimitoglou, Interactive Browsing of Remote JPEG 2000 Image Sequences, in *International Conference on Pattern Recognition (ICPR), 2010 20th.* IEEE Press, Aug. 2010, pp. 3179 3182.

[39] L. Lima, D. Taubman, and R. Leonardi, JPIP proxy server for remote browsing of JPEG2000 images, in *Proc. IEEE Workshop on Multimedia Signal Processing*, Oct. 2008, pp. 844 849.

[40] J. Liu and J. Xu, Proxy caching for media streaming over the Internet , *Communications Magazine, IEEE*, vol. 42, no. 8, pp. 88 94, 2004.

[41] Z. Su, T. Washizawa, J. Katto, and Y. Yasuda, Hierarchical image caching in content distribution networks, in *TENCON 02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering*, vol. 25, 2002, pp. 786 790.

[42] Z. SU, T. Washizawa, J. Katto, and Y. Yasuda, Integrated Pre-Fetching and Replacing Algorithm for Graceful Image Caching(Multimedia Systems), *IEICE Transactions on Communications*, vol. E86-B, no. 9, pp. 2753 2763, Sep 2003.

[43] D. H. Lee, J. S. Kim, S. D. Kim, K. C. Kim, Y.-S. Kim, and J. Park, Adaptation of a Neighbor Selection Markov Chain for Prefetching Tiled Web GIS Data, in *Proc. International Conference on Advances in Information Systems*, Oct. 2002, pp. 213 222.

[44] S. Quinn and M. Gahegan, A Predictive Model for Frequently Viewed Tiles in a Web Map, *Transactions in GIS*, vol. 14, no. 2, pp. 193 216, Apr. 2010.

[45] Group on Interactive Coding of Images, CADI software; Open Source JPIP implementation, 2008-2013, [Online]. Available: http://gici.uab.es/CADI and http://sourceforge.net/projects/cadi.

[46] Sourceforge. http://sourceforge.net.

[47] S. Qian, A. Hollinger, D. Williams, and D. Manak, Vector Quantization Using Spectral Index-Based Multiple Subcodebooks for Hyperspectral Data Compression, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1183 1190, May 2000.

[48] J. Saghri, A. Tescher, and J. Reagan, Practical transform coding of multispectral imagery, *IEEE Signal Processing Magazine*, vol. 12, no. 1, pp. 32 43, 1995.

[49] J. E. Fowler and J. T. Rucker, *Hyperspectral Data Exploitation: Theory and Applications.* Hoboken, NJ, USA: John Wiley & Sons Inc., 2007, ch. 3D wavelet-Based Compression of Hyperspectral Imagery , pp. 379 407.

[50] B. Penna, T. Tillo, E. Magli, and G. Olmo, Progressive 3-D coding of hyperspectral images based on JPEG2000, *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 125 129, 2006.

[51] X. Tang and W. A. Pearlman, *Hyperspectral Data Compression.* USA: Springer, 2006, ch. Three-Dimensional Wavelet-Based Compression of hyperspectral Images , pp. 273 308.

[52] M. Klimesh, A. Kiely, H. Xie, and N. Aranki, ICER-3D: A Progressive Wavelet-Based Compressor for Hyperspectral Images, IPN Progress Report, Tech. Rep., February 2006, http://ipnpr.jpl.nasa.gov/progress_report/42-164/164A.pdf.

[53] J. L. Monteagudo-Pereira, F. Auli-Llinas, J. Serra-Sagrista, and J. Bartrina-Rapesta, Smart JPIP Proxy Server with Prefetching Strategies, in *IEEE Data Compression Conference 2010 (DCC 2010).* IEEE Press, Mar. 2010, pp. 99 108.

[54] J. L. Monteagudo-Pereira, F. Auli-Llinas, J. Serra-Sagrista, A. Zabala, J. Maso, and X. Pons, Enhanced Transmission of JPEG2000 Imagery through JPIP Proxy and User-Navigation Model, in *IEEE Data Compression Conference 2012 (DCC 2012).* IEEE Press, Apr. 2012, pp. 22 31.

[55] J. L. Monteagudo-Pereira, F. Auli-Llinas, and J. Serra-Sagrista, JPIP Proxy Server with Prefetching Strategies Based on User-Navigation Model and Semantic Map, *IEEE Transactions on Multimedia*, 2013, in Press.

[56] Object Management Group, Unified Modeling Language, http://www.uml.org/.

[57] Free Software Foundation, GNU General Public License, http://www.gnu.org/licenses/gpl.html, Jun. 2007, version 3.

[58] JPEG 2000 image coding system: Interactivity tools, APIs and protocols. Amendment 4: JPIP server and client profiles , International Organization for Standardization / International Electrotechnical Commission (ISO/IEC); International Telecommunication Union-Telecom Standardization (ITU-T), May 2010.

[59] T. Richter, B. Brower, S. Martucci, and A. Tzannes, Interoperability in JPIP and its standardization in JPEG2000 Part 9, in *Proc. SPIE International Conference on Satellite Data Compression, Communication and Processing*, vol. 7443. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Aug. 2009, pp. 744 316,1 10.