

INTERACTIVE BROWSING OF REMOTE JPEG 2000 IMAGE SEQUENCES

J.P. García Ortiz, V. González Ruiz, I. García

University of Almería
Comp. Architecture and Electronics Dept.
C.P. 04120, Almería, Spain

D. Müller, G. Dimitoglou

European Space Agency c/o
NASA Goddard Space Flight Center
Mailcode 671.1, Greenbelt, MD 20771

ABSTRACT

This paper studies a novel prefetching scheme for the remote browsing of sequences of high resolution JPEG 2000 images. Using this scheme, a user is able to select randomly any of the remote images for its analysis, repeating this process with other images after some undefined time. Our solution has been proposed in a low bit-rate communication context where the complete transmission of any of the images for its lossless recovery should take too much time for an interactive visualization. For this reason, quality scalability is used in order to minimize the decoding latency. Frequently, the user can also play a “video”, moving sequentially on the neighbour (consecutive in time over previous or following) images of the currently displayed one. With the objective of hiding also the link latency, the proposed data scheduler transmits in parallel data of the image that it is currently displayed and data of the rest of the temporally adjacent images. This scheduler uses a model based on the quality progression of the image in order to estimate which percentage of the bandwidth is dedicated to prefetch data. Our experimental results prove that a significant benefit can be achieved in terms of both subjective quality and responsiveness by means of prefetching.

1. INTRODUCTION

Some of the powerful features offered by the novel JPEG 2000 multi-part standard [3] are lossless/lossy compression, random access to the compressed streams, incremental decoding and high degree of spatial and quality scalability. These characteristics have led it to obtain the recognition as a state-of-the-art solution among applications for remote browsing of high-resolution images. In this kind of environments, the images are stored on the server side after being encoded with the highest required quality. Using the JPIP protocol, defined in Part 9 [4] of the JPEG 2000 standard, clients can interactively explore remote image data by specifying a window of interest

(WOI). This data exchange uses the available bandwidth efficiently, requiring no recoding or additional processes. The server extracts only the required data from the images and transmits it to the clients.

JPEG 2000 has already been successfully used in many scientific areas; e.g., in tele-microscopy [8] or tele-medicine [5]. A promising application in astronomy is the JHelioviewer project [6], developed by the European Space Agency (ESA) in collaboration with the National Aeronautics and Space Administration (NASA). Its main goal is to deploy an interactive, data browsing and analyzing platform to accommodate the staggering data volume of 1.4 TB of images per day that will be returned by the Solar Dynamics Observatory [7]. Among other data products, SDO will provide full-disk images of the Sun taken every 10 seconds in eight different ultraviolet spectral bands with a resolution of 4096×4096 pixels. As of today, the combination of JPIP and JPEG 2000 seems to offer the best solution in order to efficiently browse image data sets of this magnitude.

Each solar image is associated to a date/time stamp. The basic functionality supported by JHelioviewer is to allow users to explore the available data for an instant of time. Nevertheless, one of the most valued features is to offer the ability of moving smoothly through a solar image sequence given a specific range of time (ROT), and showing the changes in the WOI. In fact, this functionality might also be interesting for many other contexts, like tele-medicine or tele-microscopy.

In order to achieve a good responsiveness and avoid annoying quality gaps when moving through an image sequence, a special prefetching strategy is required. Based on extensive research, it appears that the only published contribution regarding prefetching strategies for remote browsing of JPEG 2000 images is the paper by A. Descampe et al. [2]. In their work, the authors propose and evaluate several solutions to anticipate future WOIs in order to achieve a better responsiveness. All of these solutions only take into consideration user exploration of single large-scale images, discarding the dynamic nature of a navigation along an image sequence. In the case studied in this paper, the resolution of the images is not as high as that used in the work by A. Descampe et al. as it is more important to improve the responsiveness to user

THIS WORK HAS BEEN FUNDED BY GRANTS FROM THE SPANISH MINISTRY OF SCIENCE AND INNOVATION (TIN2008-01117) AND JUNTA DE ANDALUCÍA (P08-TIC-3518), IN PART FINANCED BY THE EUROPEAN REGIONAL DEVELOPMENT FUND (ERDF).

movements along the sequence as opposed to through just one single image. Moreover, all the solutions proposed by [2] require a special scheduling of the JPEG 2000 packets, which is rather difficult to implement considering the existing standard. This paper proposes an efficient prefetching strategy for remote browsing of JPEG 2000 image sequences that offers good performance and smooth transitions, as well as easy implementation.

The rest of the paper is organized as follows: in Section 2 the problem in question is analyzed in detail. Section 3 is dedicated to explaining the proposed solution, which is later evaluated in Section 4. The paper ends with some conclusions and future work (Section 5).

2. PROBLEM DESCRIPTION

Client/server JPIP communication is based on the exchange of requests and responses. Within each request, clients specify, among other parameters, the remote file to explore and the WOI to be shown. Files may contain a sequence of N different images (in the case of JHelioviewer, they are related to a specific ROT), so requests must also include the desired range of images $[a, b]$, with $0 \leq a \leq b \leq N - 1$. The same WOI is retrieved from all the images within the range.

There are two kind of possible JPIP requests: stateless and session-oriented. Stateless requests are independent of each other, and no state is recorded during the exchange of messages between clients and servers. In the case of the session-oriented requests, all the requests related to the same remote image file are associated with the same session. This allows the server to remember which image parts have been sent to a client, thus avoiding redundant transmissions (e.g. for overlapped WOIs). Most of the remote browsing applications use this kind of communication. JPIP servers assume by default that clients always retain all of the data received within a session. If a client needs to remove some of the data; for example, in order to release memory, the server should be notified. This study does not deal with client resource restrictions, and we therefore assume that there are none. Session-oriented communications allow clients to control the flow of data. For a certain WOI, a client can specify within the request the maximum length L desired for the server response in order to perform a data flow control between the client and the server. Therefore, due to the session state information, the client can retrieve the data of the WOI by L increments, by simply repeating the same request several times. In the case of image sequences, the response data should be uniformly distributed over the requested range by the server. For each request, the L value may be adapted depending on specific requirements. The most common scheme is a proper balance between the exploitation of the available bandwidth and the response time to WOI changes by the user. For large L values, any of the blocks of data of a new WOI would take a long time to be received, especially at low bandwidths, after

receiving the data of the previous WOI. This would generate long response times for the user interactions. On the contrary, a shorter elapsed time is achieved when a smaller value of L is used, even though additional overhead is generated due to the increase in the number of transmitted protocol headers.

Some client applications, like `kdu_show` [1], adapt the L value according to the relation between the RTT (round-trip time) propagation delay¹ and the time T taken to extract the message data from the communication link, for each server response. Then, L is modified with the aim of equalling RTT/T to a certain target ratio (L is decreased if the ratio is higher and increased if it is lower), just after retrieving a server response and before performing the next request. This method is implemented by the clients in the solution proposed by this paper.

This paper focuses on JPIP applications, such as JHelioviewer, designed to explore remote image sequences. Fig. 1 shows an example of five sequential time instances during a remote browsing session using JHelioviewer. Once the user has selected a ROT, the server builds a virtual JPEG 2000 file which only contains links to those solar images whose date/time stamp belongs to that ROT. The client starts a JPIP session for that file and requests the first image, displayed at time t_0 . The user can then watch the sequence of images belonging to the ROT, one by one. In this example, at time t_2 the user zooms-in to a certain region, thus changing the current WOI. From this new WOI, the user continues moving forward through the sequence, to instances t_3 and t_4 .

This type of interactive browsing requires a new communication scheme capable of offering smooth transitions while maintaining good responsiveness, and designed to be implemented over session-oriented JPIP communications. An added value is that this scheme is easy to implement on the client end by simply addressing how to combine the parameters L and $[a, b]$ of every request, and without requiring any server or protocol modification.

3. THE PROPOSED PREFETCHING STRATEGY

In order to minimize the visual quality differences between the currently displayed image (i_c) and the rest of images of the ROT ($[i_a, i_b]$, where $a \leq c \leq b$), the model herein proposed implements a prefetching scheduler that decides which images of a given ROT are going to be prefetched. It also decides what amount of the bit budget is going to be dedicated to the WOI and also to the ROT. We start by addressing the second aspect first. In the technique here in proposed, it is assumed that the images have been encoded using suitable compression parameters in order that they allow the JPEG 2000 scalability to be fully exploited. It is also assumed that, for every WOI request, the JPIP server delivers the associated data minimizing the distortion of the displayed image. This

¹RTT is the elapsed time between the instant in which the client generates the request and the instant in which the reply arrives to the client.

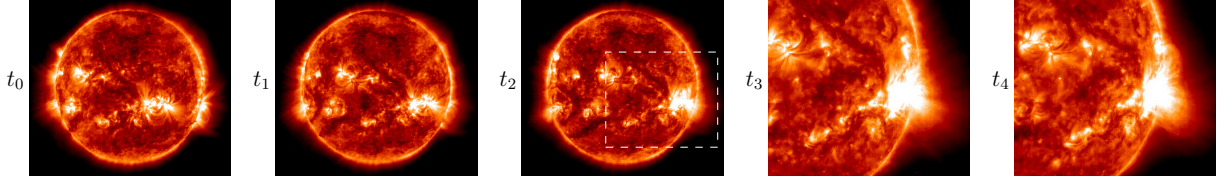


Fig. 1. An example of different time instances from a remote browsing session.

means that the Mean Square Error (MSE) between the final WOI w and the reconstructed one $w'_{(b)}$ when b bits have been received, follows a function

$$E_w(b) = MSE(w, w'_{(b)})$$

with an approximately exponential behaviour (with a negative exponent). When b bits have been received, the value of $E_w(b)$ can be used to decide which percentage of the next request is assigned to both the WOI and the ROT prefetching.

A main issue related to the calculation of $E_w(b)$ is that it depends on the content of w ; i.e., image data that is only known at the end of the transmission. However, taking into account the exponential trends of $E_w(b)$, this behaviour is fairly similar to a function that only considers the differential quality increments of w' , that is,

$$dE_w(b) = MSE(w'_{(b)}, w'_{(b-K)})$$

for certain constant increment K . The value K must be as small as possible taking into account the granularity of the image data. This must allow decodings with increments of K , especially at low bit-rates, because, on the contrary, the function $dE_w(b)$ would go to zero.

When the $dE_w(b)$ value is near zero, most of the requested data should be dedicated to the ROT prefetching since the data received is poorly increasing the quality of the WOI. With this idea in mind, the normalized percentage of each request to be dedicated to the prefetching can be modelled by

$$P_w(b) = \sigma e^{-dE_w(b)}$$

where the value $0 \leq \sigma \leq 1$ is used to limit the percentage.

According to $P_w(b)$, when the L value has been adjusted after receiving a server response, the next request to the server would be divided into $L_c = (1 - P_w(b))L$ bytes for the current WOI w , within the current image shown i_c , and $L_p = P_w(b)L$ bytes for the images $\{[i_{c-\lambda}, i_{c-1}], [i_{c+1}, i_{c+\lambda}]\}$, for the same WOI, where λ defines the size of the ROT (a window of images that is centered in i_c but does not include i_c).

Fig. 2 shows a rate/distortion comparison, in terms of peak signal-to-noise ratio (PSNR) in decibels, between a WOI retrieved normally and the same WOI retrieved prefetching data according to $P_w(b)$. Also, the value of $P_w(b)$ is shown. As can be observed, the loss of quality produced by the ROT prefetching is negligible at the beginning of the transmission and visually insignificant at the end, where a user could

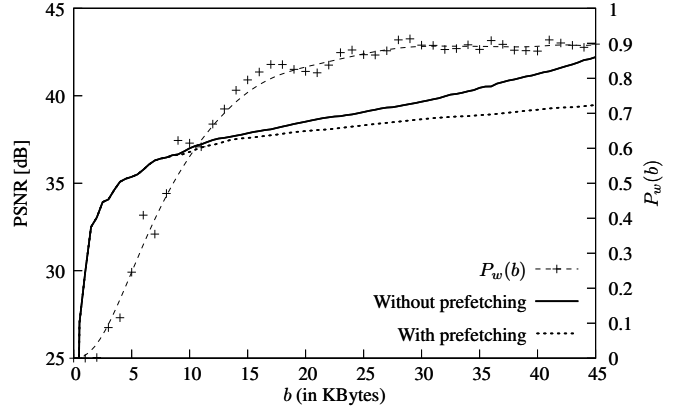


Fig. 2. Quality of a WOI retrieved with and without prefetching. The normalized percentage $P_w(b)$ (with $\sigma = 1$ and $K = 1000$ bytes) has been also represented.

hardly differentiate between both WOIs. Moreover, when 45 K Bytes have been retrieved, around 25% of this data has been used for prefetching.

From an implementation perspective, several points must be clarified. JPIP does not allow the specification of different values of L within the same request. Due to this limitation, in order to carry out the prefetching of the ROT, each request must be divided into two requests: one for i_c with L_c and another one for the prefetching with L_p . These requests should be sent continuously to avoid any communication delays and benefit from the transmission pipelines.

Since an independent request must be used for i_c as well as for prefetching, it is necessary to control the overhead generated by the protocol. Experience leads to the assumption that, on average, JPIP servers include around $H = 300$ additional bytes within each response due to headers. Once the values L_c and L_p have been obtained, they must be modified depending on the ratio H/L_p . If this ratio is not below a certain threshold, L_c is set to L , thus discarding the second request for prefetching. A good value for this threshold is 0.5 or less. Changes in the value of H might also be taken into account during the communication for better performance.

Once the current WOI has been completely received, the client must continue prefetching data, using $L_p = L$. This makes it possible to also exploit the time spent by the user to analyze the content of the image.

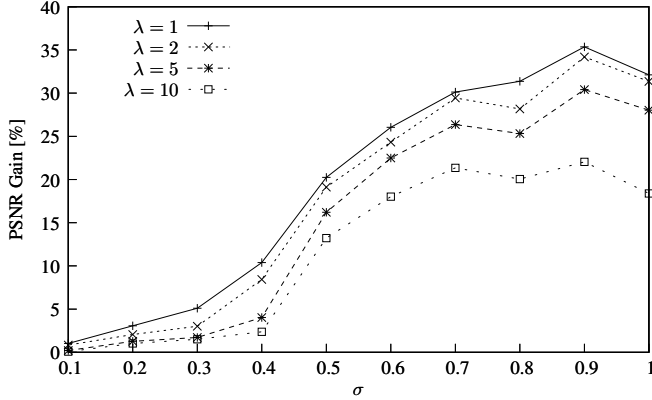


Fig. 3. Experimental results.

4. EXPERIMENTAL RESULTS

This proposed solution has been implemented in the JHelioviewer client. A random user browsing session composed of 200 consecutive movements over a remote file that contains a sequence of eighty-eight 4096×4096 solar images has been generated. The same session was simulated twice, once with prefetching technique of this study, and another time without it, for each evaluated condition. The Kakadu JPIP server [1] has been used for these experiments. The client/server bandwidth has been fixed to a constant bit-rate of 1 M bit/s, with a RTT of one second (1 s).

A slightly modified version of the user model proposed by A. Descampe et al. has been used for generating the random user browsing session of the 200 movements. The possible user movements have been reduced to five: panning, zoom in, zoom out, move forward and move backward. The first movement consists of changing the position of the current WOI to a new random position, within the same resolution level, with a distance equal to 512 pixels. The zooming movements change the resolution level of the WOI one by one. The last two movement types allow the moving through the sequence one image at a time. None of these movements modify the WOI dimension, which is always 1024×1024 pixels. It is assumed that the user behaviour is defined by a first order Markov process, so given a current movement, the probability of choosing the same movement as the next one is δ , whilst the probability of choosing a different one is $(1 - \delta)/4$. The experiments in this study have used a value of $\delta = 0.3$.

During the simulated browsing session, as soon as the quality of the current reconstructed WOI achieves a quality superior to certain threshold Θ , in terms of PSNR, the next movement is triggered. A value of $\Theta = 38$ dB was chosen for the experiments in this study.

A total of 40 different conditions have been evaluated, varying σ from 0.1 to 1, in steps of 0.1, and giving to λ the values 1, 2, 5 and 10. The average difference of the PSNR obtained when the remote browsing session is performed with

and without prefetching has been calculated for each condition. This difference is expressed as a percentage relative to the first session. The value used for K has been 1000 bytes.

Fig. 3 shows the results of these experiments. As it can be observed, independently of the values of σ and λ , there is always an improvement of the obtained PSNR. The best results are achieved with high values for σ and low values for λ . With low σ values the performance is hardly improved. High λ values are not recommended according to the user model studied here. If movements through the sequence were more predictable, the λ value might be increased.

5. CONCLUSIONS

This paper proposes a new efficient prefetching technique for interactive remote browsing of JPEG 2000 image sequences. Its most relevant characteristics are: i) it offers an easy implementation that can be added to any existing JPIP client/server architecture; ii) from the server bandwidth, a percentage estimated according to a differential quality model function is allocated to prefetching; and iii) an average improvement of the reconstructed WOI is always achieved, independently of how much fine-tuning is carried out. In continuation of this study, future work is intended to be carried which will analyze an extension of this technique that utilises sequences of images for video streaming.

6. REFERENCES

- [1] Kakadu JPEG 2000 SDK. <http://www.kakadusoftware.com>.
- [2] A. Descampe, C. D. Vleeschouwer, M. Iregui, B. Macq, and F. Marques. Prefetching and caching strategies for remote and interactive browsing of JPEG 2000 images. *IEEE Transactions on Image Processing*, 16(5):1339–1354, May 2007.
- [3] International Organization for Standardization. Information Technology - JPEG 2000 Image Coding System - Core Coding System. ISO/IEC 15444-1:2004, September 2004.
- [4] International Organization for Standardization. Information Technology - JPEG 2000 Image Coding System - Interactivity Tools, APIs and Protocols. ISO/IEC 15444-9:2005, November 2005.
- [5] K. Krishnan, M. Marcellin, A. Bilgin, and M. Nadar. Efficient transmission of compressed data for remote volume visualization. *IEEE Transactions on Medical Imaging*, 25:1189–1199, September 2006.
- [6] D. Müller, B. Fleck, G. Dimitoglou, B. W. Caplins, D. E. Amadigwe, J. P. G. Ortiz, A. A. B. Wamsler, V. K. Hughitt, and J. Ireland. JHelioviewer: Visualizing large sets of solar images using JPEG 2000. *Computing in Science and Engineering*, 11(5):38–47, September 2009.
- [7] W. Pesnell. The Solar Dynamics Observatory: Your eye on the Sun. In *37th COSPAR Scientific Assembly*, volume 37 of *COSPAR, Plenary Meeting*, pages 2412–+, 2008.
- [8] V. Tuominen and J. Isola. The application of JPEG 2000 in virtual microscopy. *Journal of Digital Imaging*, 2007.