PRE-FETCHING STRATEGIES FOR REMOTE AND INTERACTIVE BROWSING OF JPEG2000 IMAGES

A. Descampe[1,a]*, C. De Vleeschouwer*[1,a]*, M. Iregui*[1]*, B. Macq*[1] *and Ferran Marquès*[2]

[1]Communications Laboratory, Université catholique de Louvain, Belgium
[2]Image Processing Group, Technical University of Catalonia, Barcelona, Spain

ABSTRACT

This paper considers the remote interactive browsing of large JPEG2000 images. In contrast with previous contributions, we focus on the dynamic nature of the system. Practically, we study the conditions under which *a priori* knowledge about the user behavior may help to improve the browsing system reactivity, when combined with appropriate pre-fetching mechanisms. In particular, our simulations show that, due to the latency inherent to client/server exchanges, a benefit can be drawn by scheduling future window of interest (WoI) data before all current WoI data have been sent to the client. They also reveal that an accurate knowledge of the user behavior is not necessary to get important improvements over conventional scheduling approaches.

1. INTRODUCTION AND RELATED WORK

The development and diversification of computer networks has created the potential for remote access to large scale images. In this paper, we are concerned with the scheduling of image data transmitted in a remote browsing system. By data scheduling, we mean the selection of data sent by the server to a client, at any point of the transmission session, so as to maximize the reactivity of the browsing application. Concerning the storage format of the images, our work takes advantage of the JPEG2000 compression standard ([1, 2]) which offers many desirable features for remote browsing [3].

In earlier contributions JPEG2000 data units are scheduled so as to rapidly increase the window of interest (WoI) displayed quality. In practice, the transmission order is defined either based on a predefined prioritization of JPEG2000 data units [4, 5], or based on a rate-distortion criterion [3]. Despite the dynamic nature of the interactive browsing session, we note that all these approaches are static in essence. That is, only the current state of the WoI is taken into account to define the JPEG2000 data transmission order. In contrast, our work proposes data scheduling strategies that take into account the fact that the WoI changes interactively along the browsing session.

In order to render the dynamically evolving WoI, we formalize the user behavior and define the conditions under which the partial or complete knowledge of this user behavior may help to improve the browsing system reactivity, i.e. to reduce the average time between consecutive navigation commands, when combined with appropriate scheduling mechanisms. In particular, anticipated pre-fetching is proposed to exploit the browsing system latency by transmitting data that are expected to become relevant in the future before all data corresponding to the current WoI have been forwarded to the client.

[a] A. Descampe and C. De Vleeschouwer are funded by the Belgian NSF.
The picture used for the experiments has been provided by SPOT Image in the scope of IST-2000-28646 PRIAM project.

Few works have already proposed to pre-fetch data based on the knowledge of the user behavior. In [6], Lin and Zheng propose heuristic mechanisms to improve the browsing system responsiveness. In contrast, our paper solves the pre-fetching problem in a formal rate-distortion (RD) framework. In [7], the authors combine JPEG2000 flexibility with a user navigation model to manage the client cache and pre-fetch data. However, the authors in [7] assume that user navigation commands are regularly spaced. So, they neglect the fact that the data displayed at the client play a key role on the release of a novel request by the user. In contrast, our work considers the latency induced by data transmission and interpretation, and analyzes its impact on data pre-fetching strategies.

The remainder of the paper is organized as follows. Section 2 describes key elements from JPEG2000 and the navigation system used. In Section 3, we review a static rate-distortion based scheduling of JPEG2000 units which supports our own contribution presented in Section 4. Simulations results, presented in section 5, validate our proposed scheduling methodology. Section 6 concludes the paper giving general guidelines to design an interactive remote browsing system.

2. SYSTEM BASICS

2.1. JPEG2000 code-stream abstraction

In this section, we explain briefly how the JPEG2000 codestream can be abstracted as a set of independent groups of hierarchically ordered packets. Interested readers may refer to the literature for additional information about JPEG2000 [1, 2], and the underlying EBCOT algorithm [8].

In short, JPEG2000 is based on a discrete wavelet transform, whose subbands are partitioned into *codeblocks* that are coded independently. Each codeblock is coded into an embedded bitstream, i.e. into a stream that provides a representation that is (close-to) optimal in the rate-distortion sense when truncated to any desired length. For each codeblock, a decreasing sequence of Lagrange multipliers $\{\lambda_q\}_{q>0}$ identifies an ordered set of truncation points that partition the codeblock bitstream into a sequence of incremental contributions [8]. Incremental contributions from the set of image codeblocks are then collected into so-called quality layers providing distortion scalability at the image level. Resolution scalability and spatial random access to the image result from the fact that each codeblock is associated to a specific subband and to a limited spatial region.

Codeblocks associated to a given resolution are grouped into *precincts*, based on their spatial location [1, 3]. As a consequence of the quality layering defined above, a precinct can also be viewed as a hierarchy of *packets*, each packet collecting the parts of the codestream that correspond to a given quality among all codeblocks

matching the precinct resolution and position. Hence, packets are the basic access unit in the JPEG2000 codestream.

2.2. Remote browsing

The architecture of the remote browsing system studied in this paper is largely inspired by the works in [5]. Based on the WoI features coming from the client, a JPEG2000 parser generates and addresses appropriate requests to the server in order to provide the client with the JPEG2000 packets that are the most relevant to the WoI. This parser involves a scheduler to select the right packets to send at any point of the browsing session, in order to maximize the browsing system reactivity. The scheduler optimization certainly constitutes the key contribution of our paper.

3. PACKET SCHEDULING FOR A STATIC WOI

In [3], Taubman and Rosenbaum proposed a rate-distortion optimal solution to download JPEG2000 packets in an order that maximizes, at any point of transmission, the quality displayed in a pre-defined WoI. As this work supports our own contribution (Section 4), we briefly review in this section the notations and main results of this paper.

The solution in [3] consists in a greedy algorithm that selects the packets to transmit in decreasing order of WoI distortion reduction per unit of transmission. Based on Section 2.1, we know that a JPEG2000 packet is completely defined by its precinct indices r and p[1] , and by its layer index q. Following [3], we introduce $s_b(r,p,q)$ to be the size in bytes of the q^{th} packet of the (r,p) precinct, and denote $d(r,p,q)$ to be the amount by which the distortion, measured on the whole image, is decreased if the packet (r,p,q) is decoded compared to the distortion if only the packets (r,p,i), $i < q$, are decoded. As a consequence of that definition, the metric $d(r,p,q)$ has to be additive. This property is verified from the approximation of the Mean Square Error (MSE) defined in [8].

Based on the distortion reduction provided by a packet on the whole image $d(r,p,q)$, we follow [3] to define the distortion gain on a static WoI w as

$$d_\xi(r,p,q,w) = \xi(r,p,w).d(r,p,q) \tag{1}$$

where $\xi(r,p,w)$ denotes the fraction of pixels affected by precinct (r,p) that belongs to WoI w. Hence, packets have to be forwarded to the client in decreasing order of $d_\xi(r,p,q)/s_b(r,p,q)$. An alternative, proposed in [3], consists in approximating the distortion reduction per unit of transmission, i.e. $d(r,p,q)/s_b(r,p,q)$, by the Lagrange multiplier λ_q used to define the truncation points of the q^{th} quality layer.

4. DYNAMIC PACKET SCHEDULING

In contrast to Section 3, we now consider a dynamically evolving WoI, and the consequences it has in terms of packet scheduling decisions. For example, some packets that are currently useless might become critical in a near future and should be transmitted anticipatively in order to reduce the navigation system latency. Fundamentally, the anticipation of future commands is motivated by the fact that there exists a *reaction delay* between the emission of a packet by the server and the feedback it causes in terms of WoI definition. This reaction delay causes a waste of resources if current WoI data are still forwarded by the server while sufficient packets have already been sent for the user to choose the next command. This delay includes transmission and decoding delays but also the time the user needs to analyze the newly displayed visual information.

In practice, the anticipation of future WoI relies on two components. First, it depends on the *a priori* knowledge of future requests (modelization of the user behavior). Second, it requires the definition of schedulers that favor the download of promising JPEG2000 packets (optimal image pre-fetching strategies) rather than useless quality increments for the current WoI.

4.1. User navigation and reaction models

We characterize the user by two components. First, a navigation model defines which command is expected to be released next. Second, a reaction model defines when the next command is triggered, in response to data transmitted for current WoI. Our work does not pretend to define the "ultimate" navigation and reaction models, which we believe are tightly connected to the envisioned application. Rather we fix these models *a priori* and are interested in the analysis of how appropriate scheduling mechanisms are able to take advantage of their knowledge.

To define our navigation model, we adopt the formalism introduced in [7]. We assume that the user only browses through various image resolutions and spatial locations and that he has only 6 possible actions : zoom-in, zoom-out, up, down, right and left. To define the sequence of user's actions, we then assume that an action is chosen identical to the previous one with probability δ, while all other actions have the same probability $(1-\delta)/5$. Hence, the δ parameter reflects the deterministic nature of the navigation process. Although the server can calculate with this model the probability that a packet appears in the WoI at any future step, we decided to restrict our prediction of the future to a "myopic" one-step look ahead.

To model the reaction of the user, we adopt a rudimental formalism. Fundamentally, the model has to reflect that (i) a user command is triggered in response to the content displayed in the current WoI, and (ii) packets sent out by the server need some time to be conveyed, displayed and interpreted at the client. In the rest of the paper, we assume that the next navigation command is triggered τ seconds after the PSNR in the current WoI went beyond a threshold Θ, in dBs.

We now present the three pre-fetching strategies proposed to take advantage of the knowledge of user behavior models.

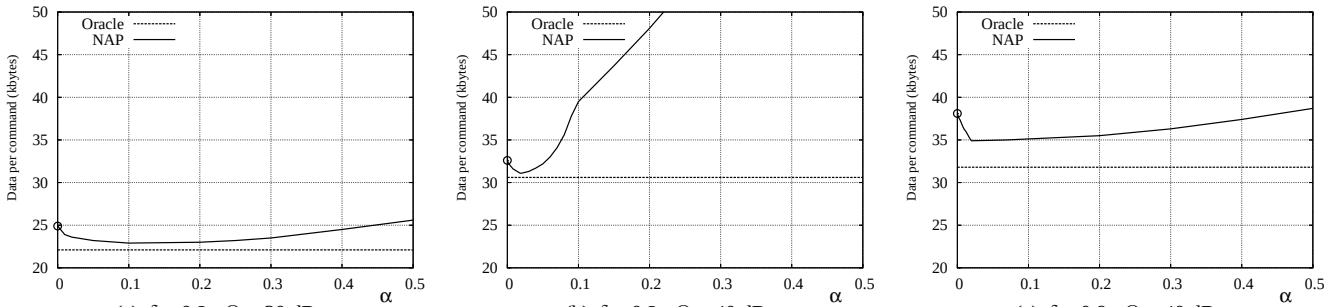
4.2. Oracle scheduler

The *oracle scheduler* knows all about the user behavior. It knows the δ parameter of the navigation model, and the Θ and τ parameters of the reaction models. Hence, the oracle feeds the current WoI until the Θ quality threshold is achieved. Upon that point, the oracle selects the JPEG2000 packets so as to maximize the gain in quality expected per unit of transmission for future WoIs. The expected gain in quality is defined, at a given time t, by

$$E[d_\xi(r,p,q,W_t)] = \sum_{w \in \mathcal{W}_t} p_{W_t}(w).d_\xi(r,p,q,w). \tag{2}$$

where W_t denote the random variable that defines the next WoI selected within the set $\mathcal{W}_t$ of possible future WoIs, $p_{W_t}(w)$ denotes the probability for w to be the next WoI and $d_\xi(r,p,q,w)$, which is defined in Equation (1), denotes the reduction of distortion provided on w by the q^{th} quality layer of the (r,p) precinct.

[1] r and p respectively characterizing the resolution and the spatial location of the precinct.



(a) δ = 0.2 ; Θ = 30 dB (b) δ = 0.2 ; Θ = 40 dB (c) δ = 0.9 ; Θ = 40 dB

Fig. 1. Average number of bytes transmitted between two consecutive navigation commands, for the oracle scheduler (dashed line), the natural pre-fetching (circle) and the anticipated pre-fetching (solid line), as a function of the α pre-fetching parameter and for two levels of predictability (δ) and quality threshold (Θ). Reaction delay B_τ = 10 kbytes. Client cache size M = 150 kbytes.

4.3. Natural pre-fetching

In this case, we assume that the user reaction model is unknown, i.e. the scheduler ignores the impact of the current WoI quality on the instant of release of the next navigation command. A natural way to schedule data is to first feed the current WoI as long as there are packets relevant to this WoI (and not till the threshold Θ has been achieved, as for the oracle). Once this is done, data for future WoI is pre-fetched, because it is always better to transmit some additional image data than not transmit any information [7].

4.4. Anticipated pre-fetching

In this case, we still assume that the user reaction model is unknown. But in contrast to section 4.3, we propose to consider pre-fetching *before* all the information relevant to the current WoI has been transmitted to the client. Our investigation is motivated by (i) the observation that there exists a delay between the emission of image data by the server and their interpretation by the user, and (ii) the fact that the current WoI quality increments are obtained at increasing cost (in terms of transmission resources) as the downloading process goes ahead. As a consequence, at some point, it might be more beneficial to download a JPEG2000 packet that has a high probability to bring a significant gain in the future, rather than wasting resources for non-significant increments of the current WoI.

We propose a weighted sum to trade-off current and future expected benefits. Formally, based on notations of Section 4.2 and letting α denote a weighting factor between 0 and 1, the gain associated to packet (r, p, q) is given by

$$d_\alpha(r,p,q,t) = (1-\alpha).d_\xi(r,p,q,w(t)) + \alpha.E[d_\xi(r,p,q,W_t)], \quad (3)$$

where $d_\xi(r,p,q,w(t))$ and $E[d_\xi(r,p,q,W_t)]$ are defined in Equation (1) and Equation (2), respectively.

5. SIMULATION EXPERIMENTS

In this section, we rely on simulations to compare the oracle scheduler with the natural and anticipated pre-fetching methodologies. As a result, we identify the conditions that make the accurate knowledge of navigation and reaction models useful in improving the browsing system responsiveness. Only the main simulation results are presented here, a complete analysis is detailed in [9].

Given that the navigation model is assumed to be known by all schedulers, we based our simulations on a fixed sequence of navigation commands generated by the chosen model. Despite the sequence of navigation commands is fixed, the instants at which commands are triggered depends on the scheduler. Indeed, these instants are defined by the reaction model, in response to the quality displayed in the WoI, which in turn depends on the scheduling strategy. As a consequence, the average time elapsed -or equivalently and without loss of generality the average amount of data transmitted- between two consecutive navigation commands provides an objective measurement of the scheduler performance.

The generated navigation sequence contains 1000 commands. The picture used is a 32768×32768 greyscale image defined based on a crop of downtown Toulouse. JPEG2000 coding parameters have been defined as follows. Tiles have a size of 1024×1024 pixels. Six resolution levels have been considered for each tile, and precincts of 128×128 have been defined at each resolution level. Ten quality layers respectively target a quality of 26, 30, 32, 34, 36, 38, 40, 42, 44, and 46 dB in each tile.

Figure 1 plots the average number of bytes transmitted per navigation command for the 3 schedulers, as a function of the α pre-fetching parameter. In these figures, we note that the oracle scheduler (dashed line) performs significantly better than the natural pre-fetching solution (circle) and that, for a well-chosen α, anticipated pre-fetching (solid line) also improves the natural one. We also observe that the best α value decreases as the quality threshold Θ increases. Indeed, small (large) α postpones (speeds up) the anticipation of future WoIs, and a large Θ means that a high quality is required in the current WoI before moving to the next user request. Finally, we observe that the solid line goes up slower for increased α when the navigation model is predictable (i.e. δ is high). We conclude that it is easier and less risky to implement anticipated pre-fetching when future WoIs are reliably predicted than in presence of a random navigation.

Figure 2 presents the rate overhead computed for the natural and the anticipated pre-fetching, relatively to the oracle scheduler. The graph plots the overhead for a predictable (δ = 0.9) and a random (δ = 0.2) navigation model, as a function of the quality threshold Θ (Figure 2(a)) and reaction delay B_τ (Figure 2(b)). For anticipated pre-fetching, the α value is chosen as the one that minimizes the overhead.

Regarding Figure 2(a), we first observe that the overhead increases with the model predictability. It makes sense because the anticipation of future WoIs is expected to be more efficient and, there-

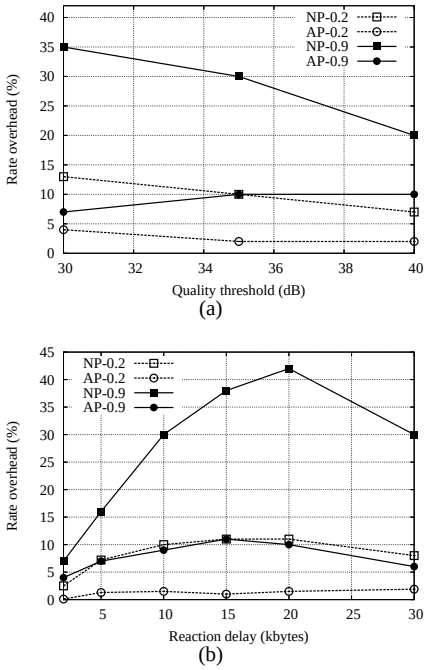


Fig. 2. Overhead in average number of bytes between consecutive navigation commands for natural (NP) and anticipated (AP) pre-fetching, compared to the oracle scheduler. Graphs are plotted as a function of (a) quality threshold Θ (with $B_\tau = 10$ kB) and (b) the reaction delay B_τ (with $\Theta = 35$ dB). The cache size is 150 kB.

fore, the performance of the oracle further increases. We observe also that the relative overhead of natural pre-fetching (NP) over the oracle scheduler and over anticipated pre-fetching decreases as Θ increases. We conclude that anticipated pre-fetching works well as long as the quality required to trigger the next navigation command is not too high (typically 35 dB). However, when a larger quality is required, it is recommended to acquire a better knowledge of the user reaction model so as to design a dedicated scheduler that performs closer to the oracle.

Regarding Figure 2(b), we note that natural pre-fetching (NP) overhead rapidly increases as B_τ increases. It reflects the incapacity of natural pre-fetching to take an advantage of the reaction delay. Unsurprisingly, that incapacity is even more penalizing when future WoIs are predicted accurately, i.e. when $\delta = 0.9$. In contrast, anticipated pre-fetching (AP) overhead remains small and about constant with B_τ. It demonstrates that anticipated pre-fetching performs close-to-optimal pre-fetching during the reaction delay, whatever the reaction delay duration or the model predictability.

For completeness, the influence of the cache size has been studied in the same way as the other parameters and is detailed in [9]. This analysis has revealed that all scheduling approaches get a similar benefit from an increased memory size. However, we observe that a large cache becomes more important as future navigation commands get less predictable (for small δ). For such unpredictable navigation models, the use of anticipated pre-fetching, or even the accurate estimation of the user reaction model, is less beneficial than a large client cache memory.

6. CONCLUSION

This paper considers the issues of scheduling JPEG2000 data in client/server interactive browsing applications. Focusing on the dynamic nature of the system, we propose a framework to analyze the responsiveness of the browsing system as a function of different scheduling schemes. Based on our simulations, we draw the following guidelines to design an interactive remote browsing system.

First, natural pre-fetching, combined with a large client cache memory, provides a simple and satisfying scheduling solution when future navigation commands are unpredictable.

In contrast, when future WoIs can be predicted reliably, the client cache size becomes less crucial, but it becomes important to derive scheduling policies that are able to anticipate future WoIs. This statement becomes even truer as the user reaction delay increases. In particular, when the level of quality in the current WoI required by the user to trigger the next command is reasonable (typically below 35 dB), a simple scheduling strategy that transmits packets in decreasing order of the weighted benefit provided on the current and future WoIs achieves close to optimal performance, nearly independently of the weighting factor.

Hence, it appears that sophisticated scheduling mechanisms, based on a deeper knowledge of the user reaction model, are only recommended when future commands are highly predictable and when a high quality is needed by the user to take a decision about the next WoI. This result is interesting because, in practice, the reaction model is expected to be difficult to estimate and formalize, especially because the reactivity of a user in front of the information displayed in a WoI depends on complex and subjective interpretation abilities.

7. REFERENCES

[1] ISO/IEC 15444-1, "JPEG2000 image coding system," 2000.

[2] M. Rabbani and R. Joshi, "An overview of the JPEG2000 image compression standard," *Signal Processing: Image processing*, vol. 17, pp. 3–48, 2002.

[3] D. Taubman and R. Rosenbaum, "Rate-distortion optimized interactive browsing of JPEG2000 images," in *IEEE Int. Conf. on Image Processing (ICIP)*, September 2003.

[4] R. Rosenbaum and D. Taubman, "Remote display of raster images using JPEG2000 and the rectangular fisheye-view," *Journal of the Int. Conf. WSCG*, vol. 11, no. 3, pp. 387–393, 2003.

[5] Jin Li and Hong-Hui Sun, "On interactive browsing of large images," *IEEE Transactions on Multimedia*, vol. 5, no. 4, pp. 581–590, December 2003.

[6] Chengjiang Lin and Yuan F. Zheng, "Fast browsing of large scale images using server prefetching and client cache techniques," in *Applications of Digital Image Processing XXII SPIE*, July 1999, pp. 376–387.

[7] A. Descampe, Jihong Ou, P. Chevalier, and B. Macq, "Data prefetching for smooth navigation of large scale JPEG2000 images," in *IEEE Int. Conf. on Multimedia and Expo (ICME)*, July 2005.

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

[9] A. Descampe, C. De Vleeschouwer, M. Iregui, B. Macq, and F. Marques, "Pre-fetching and caching strategies for remote interactive browsing of JPEG2000 images," Tech. Rep., TELE - UCL, Belgium, October 2005.