Interactive Image Viewing in Mobile Devices Based on JPEG XR

Bernardetta Saba, Cristian Perra, and Daniele D. Giusto

Department of Electrical and Electronic Engineering, University of Cagliari
Piazza D'Armi, 09123 Cagliari, Italy
{bernardetta.saba,cperra}@diee.unica.it, ddgiusto@unica.it

Abstract. Services for high definition image browsing on mobile devices require a careful design since the user experience is heavily depending on the network bandwidth, processing delay, display resolution, image quality. Modern applications require coding technologies providing tools for resolution and quality scalability, for accessing spatial regions of interest (ROI), for reducing the domain of the coding algorithm decomposing large images into tiles. Some state-of-the-art technologies satisfying these requirements are the JPEG2000 and the JPEG XR. This paper presents the design of an interactive high resolution image viewing architecture for mobile devices based on JPEG XR. Display resolution, resolution scalability, image tiling are investigated in order to optimize the coding parameters with the objective to improve the user experience. Experimental tests are performed on a set of large images and comparisons against accessing the images without parameter optimization are reported.

Keywords: JPEG XR, Interactive Access.

1 Introduction

The wide variety of devices for mobile imaging (PDA, Mobile Phone, and Smartphone) requires that the compressed image is capable to adapt to different viewing conditions. A big problem for the development of these applications is related to the high costs of storage and data transmission. The modern compression technologies offer a solution to the problems providing very efficient tools for high image compression rates.

The ISO/IEC JPEG [1] is universally accepted as the standard "de facto" in the field of lossy compression, despite it is very dated. JPEG is a compression algorithm based on DCT transform. Every image, after decomposition in 8x8 blocks, is sent to the encoder, which in turns consists of three simple steps: a FDCT, that converts the 8x8 block from the spatial to frequency domain, a quantizer that eliminates less important information and an entropy encoder to reduce the amount of images needed.

The JPEG 2000 standard [2] uses a different transform algorithm based on wavelet and an innovative system for entropy coding named EBCOT that provides better

L. Atzori, J. Delgado, and D. Giusto (Eds.): MOBIMEDIA 2011, LNICST 79, pp. 227–241, 2012.

compression performance than JPEG at high compression ratios. Image compression both binary and continuous-tone, random access to the bit-stream, processing domain encoded and robustness to transmission errors are some of JPGE 2000 tools. Nevertheless, the new algorithm has not been as successful as hoped, despite all the novel functionalities.

The search for efficient representation for digital photographs is always alive and the ISO/IEC committee has recently completed the standardization process of the new JPEG XR image compression algorithm [3].

JPEG XR has been designed to manage the dynamics of modern acquisition sensors, offering a range of new features and benefits to the needs of digital photography. JPEG XR has been designed to reduce the limitations of the available formats that do not maximize the quality of stored data and fail to reach optimal performance according to the used devices. It allows representing multiple images into a single file, to decode only a part of an image, and crop, reduce in quality, tilt or rotate without having to decode the file. This compression algorithm has been made to minimize the encoder and decoder algorithmic complexity so as to obtain a minimal memory footprint. This is mainly due to the fact that the operations that are performed in encoding and decoding are very simple (basically it is a set of sum operations). This simplicity makes it very attractive for mobile imaging applications.

JPEG XR can process very large images by dividing them into tiles, each tile can be independently decoded. An analysis of JPEG XR coding efficiency can be found in [5]. It was evinced that for low bitrate, small tile sizes damage compression efficiency. Experimental results show that the goodness of tile size is closely linked to the bitrate. In particular, it is strongly recommended to use tile sizes 512x512 and above, while it is not advisable to use tile sizes below 64x64 when compression efficiency is very relevant.

A new architecture for JPEG XR encoding is described in [6]. The paper shows that there is no dependency information in the intra-macroblock; so every process of JPEG XR encoding (PCT/POT, Quantization, Prediction, Adaptive Scanning and Entropy Coding) may be pipelined. In the previous architectures the entropy coding was implemented in order to manage the dependency intra-blocks sequentially.

The significant progress in many aspects of digital technology, especially within the area of image acquisition, data storage, printing and display, have led to the creation of a large number of multimedia applications relating to digital images, such as image browsing.

It is in applications like these that the scalability plays a fundamental role. There are two types of scalability [7]: spatial and quality scalability. In the first scenario, it possible to zoom in and out the original image; in the second scenario it is possible to see the image at different quality levels.

A solution to navigate through the entire collection of large image databases is presented in [8]: a multi-modes and integrated image retrieval method consisting in combining direct search and browsing retrieval paradigms to obtain a third model called seamless interaction. In this way, the client can specify their needs with a simple query, resulting in a gain both in efficiency and in system speed.

A JPEG 2000 interactive browsing based on a client-server Vmedia architecture is presented in [9]. In this architecture, the user is allowed to choose and eventually change the region of interest (ROI) to decode. This means that only a portion of the data stream needs to be read, reducing the amount of data transferred. The Vmedia protocol is able to give a preview of the ROI, that becomes sharper as the remaining information arrive. There is the possibility to reuse the data delivered when the user decide to change its ROI, resulting in a reduction of load time. This approach is very useful for large images such as painting databases.

An image browsing application based on JPEG-XR is presented in [10]. Region of interest specified by the user are progressively downloaded and displayed with an approach aiming to minimize the transferred information between the request and the image presentation. Server side images are stored in frequency mode order and exploiting partitioning of images into tiles. This architecture provides user experience comparable to image browsing based on JPEG 2000. Nevertheless, results have shown that PSNR image quality using JPEG 2000 is higher than JPEG XR if multiple embedded bitrates are used.

A visual attention system for image browsing applications with large image database on mobile devices has been described in [11]. This algorithm defines a selective model of attention objects (face and text) within the image, based on eye movements, to show a specific region of interest. This method is also implemented in [12] to dynamically adjust the content of the image to the display size of different users.

A path for image browsing and searching for optimal path, based on the visual attention model [11,12] is defined in [13]. A set of new approaches to automate tasks of image browsing on mobile devices is also proposed. In [14] the visual attention model is examined spatially. There is a proposal of a new method for image browsing. In this case the detect ROI, first is decreasing ranking and then readapted on small displays.

An algorithm to identify the sufficient display resolution for image browsing is detailed in [15]. The authors use Kullback-Leibler (K-L) distance to determine data loss caused by down-scaling an image. The experiments, based on visual attention and spatial contrast masking, show that find the correct display resolution does not depend on users. Two algorithms are compared, those based on vision approach and those based on non-vision approach and although both achieve good performance, the first method gives better results than the second.

A technique to extract objects from large images and view them on small displays is described in [16]. The goal is to derive human faces in an image, thus making them the ROI image. The image is then segmented and will be measured saliency. So, there will be an adaption of the ROI and will start the image browsing. It is required a compatible aspect ratio to show ROI on small displays. Before, the search path was done manually; now is being made with an automated method of browse image.

This paper presents the design of an interactive high resolution image viewing architecture for mobile devices based on JPEG XR. Display resolution, resolution scalability, image tiling are investigated in order to optimize the coding parameters with the objective to improve the user experience. Experimental tests are performed on a set of large images and comparisons against accessing the images without parameter optimization are reported.

2 JPEG XR Overview

JPEG XR image transform [3] is based on a two-level hierarchical lapped transform [4], that is a concatenation of two flexible and independent transformation operators: FCT (Forward Core Transform) and OT (Overlap Transform).

In order to execute the transformation, an image JPEGXR is divided into 4x4 macroblocks, for each color plane. Every of them consist of 16 4x4 non-overlapping blocks, on each of which is applied the FCT transform, that produces 1 DC and 15AC coefficients of first stage for any block.

The DC coefficients of all blocks are grouped together in a 4x4 block and then a FCT transform is applied again.

So, this produces other 16 coefficients: 1 DC and 15 AC coefficients of second stage. Finally, this coefficients are mapped in the first pixel of each block that is a part of the macroblock. Fig. 1 shows the coefficients mapping of the two stages of FCT, that produce 240HP coefficients of first stage and 15LP and 1 DC coefficients of second stage.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256

DC coefficient LP coefficient HP coefficient

Fig. 1. JPEG-XR macroblock coefficient mapping

The FCT Transform, the core of the transform, consists of three elementary 2x2 filter operations:

- 2x2 Hadamard Transform: T2x2h;
- 1D rotate: TOdd;
- 2D rotate: TOddOdd.

FCT is applied to each 4x4 block in two different stage, as shown in Fig. 2. Each stage provides four 2x2 transform which may be done simultaneously or in a random sequence inside the stage. Anyhow, the second stage transform can initiate only if the first stage transform is completed.

The first stage of FCT includes four 2x2 Hadamard Transform (T2x2h): earlier is applied to corners (a, b, c, d) and then to the centers coefficients (e, f, g, h) of a 4x4 block; afterwards the T2x2h is applied to upper and lower edges (i, l, m, n) and finally to the right and left edges (o, p,q, r).

The second stage continues with a T2x2h for even-even basis (A, B, C, D), with a 1D rotation for even-odd basis (E, F, G, H) and odd-even basis (I, L, M, N) respectively and lastly with a 2D rotation for odd-odd basis (O, P, Q, R).

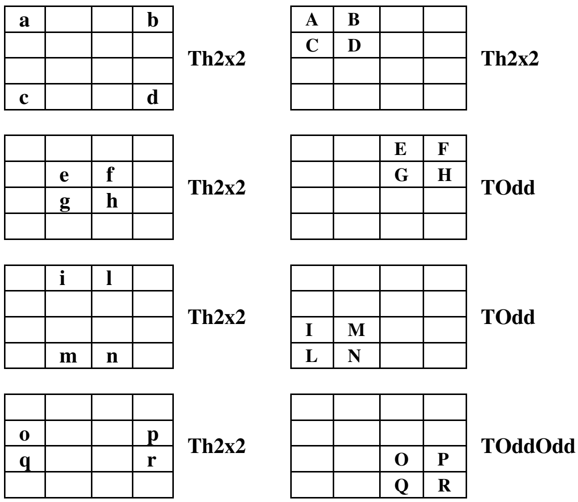


Fig. 2. Forward Core Transform steps

After the two stages, coefficients are re-ordered as shown in Fig.3.

i	Array[i]	i	Array[i]
0	0	8	1
1	8	9	11
2	4	10	15
3	6	11	13
4	2	12	9
5	10	13	3
6	14	14	7
7	12	15	5

Fig. 3. Forward Permutation

Every FCT operation is preceded by an optional filter OT (Fig.4), which is applied to a 4x4 areas, between the stacked two-dimensional blocks. (If OT_mode=0, no overlap operator is applied; if OT_mode=1 only the first level overlap is applied, otherwise if OT_mode=2 both level overlaps are performed). The overlap filter is designed to limit the blocking artifacts.

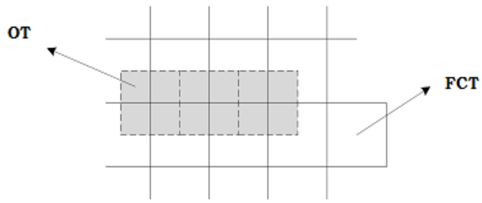


Fig. 4. Regions of support for the 4x4 FCT and OT operators [3]

The JPEG-XR decoder uses a block transform, called ICT, in which the stages are inverted. Every filter operations within the stages use its own inverse transform, and it is preceded by the inverse permutation function. The decoding process is summarize in Fig. 5.

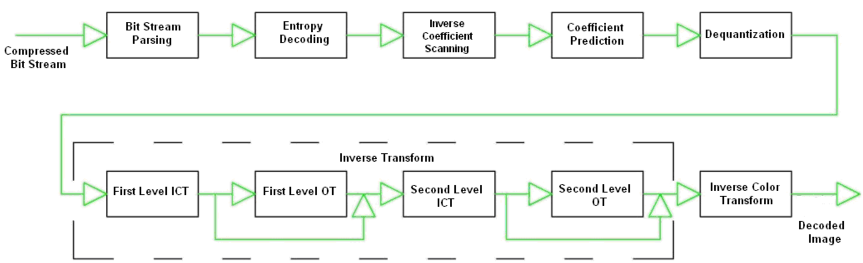


Fig. 5. Decoding process diagram [19]

The JPEG XR defines two approaches for access to the bit-stream: spatial and frequency. In both case, the bit-stream is composed by an image header followed by progressive tiles. In the first case the bit-stream of each tile is arranged in a macroblock order; in the second case the bit-stream of each tile is transmitted in multiple tile packets. It was considered the layout in the frequency mode, where the bit-stream of each tile is set up as a hierarchy of bands, as shown in Fig.6.

The tile coefficients are positioned in the following order: DC, LP, HP and FLEX band. FLEX provides additional information to HP band. The FLEX band may be not present.

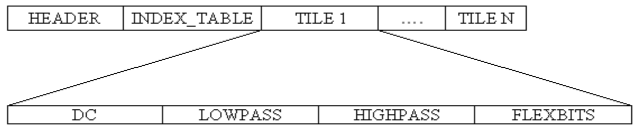


Fig. 6. Layout of JPEG XR bit-stream in frequency mode [3]

The group formed by DC, LP and HP sub-bands, produces different resolutions of the image information; while FLEX, if exists, can be used for progressive decoding.

The first scale of decoding can be obtained decoding the part of bit-stream that corresponds to DC, LP and HP coefficients.

It is then possible decode FLEX to produce the complete image.

A resolution equal to 1:16 is obtained decoding only the DC band. To obtain a 1:4 resolution of the image, only DC and LP bands need to be decoded.

3 Tiling

The process of partitioning a source image into rectangular non-overlapping blocks is called tiling. The JPEG 2000 standard compress each tile as an independent image from the other tiles. It implies that decoding any region of interest (ROI) inside a tile requires only the coded codestream of that tile. The tile are processed through the wavelet transform, (that can be reversible (RCT) or irreversible (ICT), to obtain different subbands, that are respectively LL,HL, LH and HH.

The JPEG XR standard is slightly different. A source image is divided into a grid of macroblock-aligned tiles, that enable fast local access. JPEG XR encoder can choose among three overlap filter operation, as described above: 1) non-overlapping that implies that no overlapping operation is performed; 2) one-level overlapping filter; 3) two-level overlapping filter. It is also possible decide whether to handle the tile boundaries in soft mode or in hard mode. In the first case, the overlapping filter is enabled within the tile and over the edges between the tiles. In the second case, the overlapping operation is applied only within the tile. In order to reconstruct a specific ROI, if the adjacent macroblock is placed in a distinct tile, it is necessary decode only a part of that tile.

Moreover, a structure for an optimized tile construction exists[17] and uses 256x256 ROI to reduce the overhead required to decode an image.

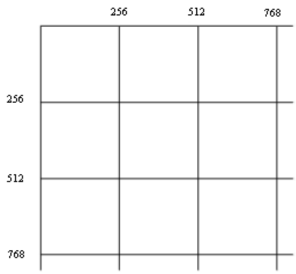
In the optimized case we can find one of these possibilities:

you want to decode a ROI that lies between two vertical tile;

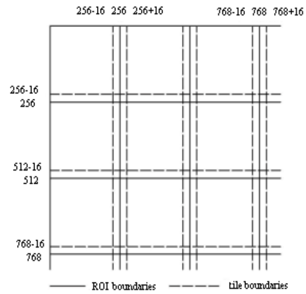
or you want to decode a ROI that lies between two horizontal tile.

In the first condition, the ROI should be on a small case separated from the edge of the horizontal tile. In this way it is not necessary decode the tile that are above.

In the second occurrence the ROI should be crushed into two neighbors vertical tile boundaries for the same reason of the previous case.



(a) JPEG 2000 e JPEG XR regular tiling



(b) JPEG XR Optimized Tile structure [17]

Fig. 7. Regular tiling (a) and optimized tiling (b)

4 Proposed Architecture

The proposed architecture is shown in Fig. 8. It is a client-server architecture for remote browsing applications for the packets exchange over the Web. The server has been implemented in C++ language and it is responsible for access management, allocation and release of resources. It stores a database of JPEG XR images and for each connection to a client user data are transferred by the means of the HTTP protocol.

At the server side JPEG XR images are stored with an optimized tile decomposition [17], in order to allow fast local access by the client device. An indexing tools analyses the JPEG XR codestream in order to extract the offset and the size of each subband. Indexes are stored in the index table for fast retrieval of the required information for the reconstruction of a ROI requested by the client. Index table is managed by the functional Index block in Fig. 8. This block has also the purpose of keeping track of the information already available at the client side and avoiding retransmission of the same data. The HTTP server receive incoming request and provide the chunks of data required by the HTTP client.

The client constructs the request on the basis of the current user view. In particular the display resolution and the level of zooms are used for determining what are the needed chunks of information and the corresponding indexes.

The proposed concept is slightly different from the concept used in the interactive protocol JPIP. In fact in the proposed architecture the computation of required chunk indexes is performed at client side, minimizing the information transferred for the request to a vector of index corresponding to the chunk number needed for display the current view to the user.

Moreover it is not needed to have an arbitrary ROI access but simplifying as much as possible the user interaction with the process of viewing high quality high resolution large images with mobile devices.

HTTP client receives incoming packet from the HTTP server containing JPEGXR image subband that are stored into the local cache. The block composer prepare the JPEG XR file merging the required chunks into a correct JPEG XR format in order to be decoded by the JPEG XR block.

The viewport block keep tracks of the visible portion of the image which is larger than the visualization device.

The user interacts with the display with classical image viewing operation such as zooming and panning.

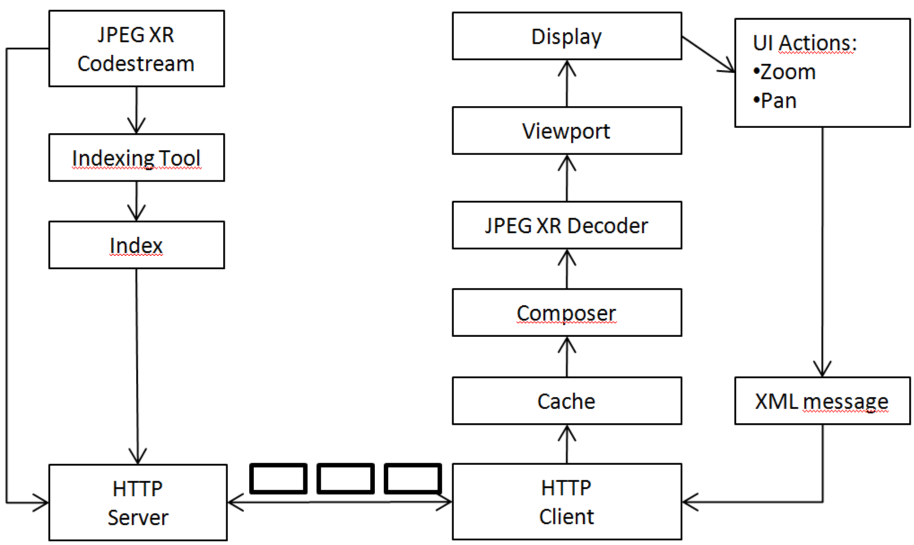


Fig. 8. Reference Architecture

5 Experimental Tests

In this section, we describe the tests performed and their results. Tests were conducted using the JPEG XR reference software and the KAKADU software [20]. The use case is as follows: once the desired ROI at a given resolution has been selected, the client must be able to zoom in and out the image that appears in his own device, or make horizontal or vertical scrolls of the current view of the image.

It is possible to request only some subset of the available sub-band coefficients. In fact, in the frequency mode, each sub-band can independently be decoded. All tiles of a particular sub-band are merged together in a unique data packet.

This allows creating smaller image previews using a resolution that fits the device used to load the image.

The client can request only the DC level transmission, if a low quality of image suffice. It is possible to increase the image quality, transmitting progressively all available sub-bands.

DC+LP+HP+FLEX sub-bands transmission ensures maximum image quality.

Removing irrelevant information obviously causes a loss of data image that must be quantified in some way. There are subjective and objective techniques. The subjective metric depends on the experience of the observers, while the objective metric consists to calculate the difference in statistical distribution of pixel values in digital images; in this way it is possible to quantify the distortion of the compressed image with respect to the original. The PSNR is the most widely used image objective metric.

The bitrate vs PSNR shows the efficiency of a compression algorithm: high bitrates provide a higher quality of the compressed data.

Several tests were performed using the three different overlapping filters and distinguishing between hard, soft and optimized mode. These results are then compared with those obtained with JPEG 2000 tests.

Image request at difference resolution level have been analyzed in order to report the bit-rate of the data transferred at each requests.

The following figures show the experimental results. Each figure contains only the average of the results of the most significant images used as a test set.

Experimental tests were carried out by defining three mode of encoding JPEG XR (soft, hard and optimized) and using the three OT mode (0,1,2). Subbands considered are: 1) only DC, 2) DC+LP, 3) DC+LP+HP, 4) DC+LP+HP+FLEX.

JPEG2000 codestream were produced in relation to JPEG XR bitrate, to make a comparison between the two coding algorithms.

In the first JPEG XR case (Fig. 9 (a-b-c)) the solution without any level of overlap was considered; in the second JPEG XR case (Fig. 10 (a-b-c)) only one overlapping filter was applied, while in the latter JPEG XR case (Fig. 11 (a-b-c)) both stages of overlapping have been taken into account. In all three cases there was a distinction between soft (a), hard (b) and optimized (c) mode, in order to compare this three methods.

As expected, experimental tests, in terms of bpp, show that in the case with L = 0 hard, soft and optimized mode show similar results. In the case with L= 1 soft and optimized mode show similar results, while the bpp slightly decreases in hard mode.

Finally, in the case with L= 2 optimized mode performance in terms of bpp presents results that are intermediate between those obtained with hard and soft mode.

In all experimental JPEG XR tests, therefore, hard tiles outperform soft tiles and optimized tile; but the results obtained with the tile optimization, in terms of overhead, are very acceptable if compared with those of JPEG 2000.

Fig. 12 shows the JPEG 2000 results obtained in the case of image subdivision in tiles of 256x256 size, without any type of optimization.

It is possible to notice that the JPEG 2000 bpp are very similar to those obtained with JPEG XR experiments, but in JPEG2000 we can see a gradual increase of bpp as bitrate increased, even if only DC.

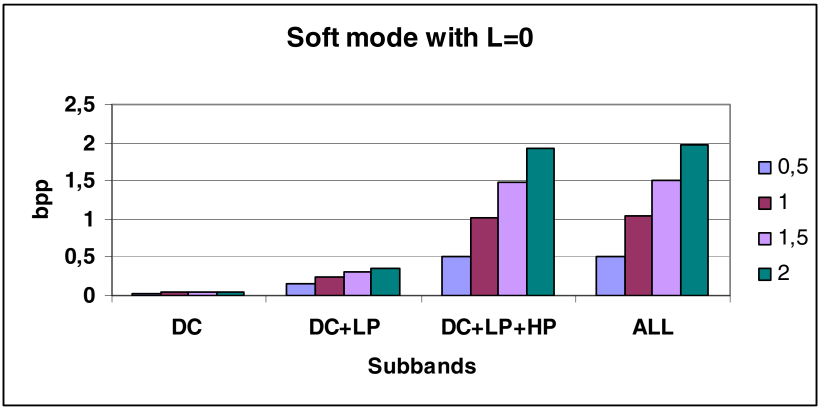


Fig. 9 (a). Bitrate for each JPEG XR subband for four target bitrates (soft mode, L=0)

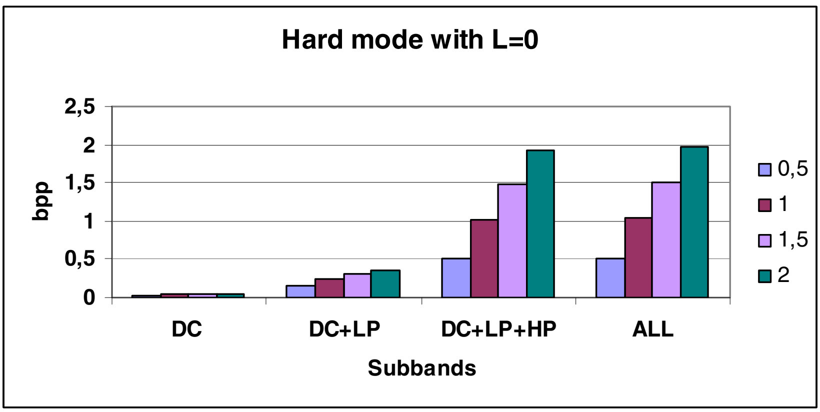


Fig. 9 (b). Bitrate for each JPEG XR subband for four target bitrates (hard mode, L=0)

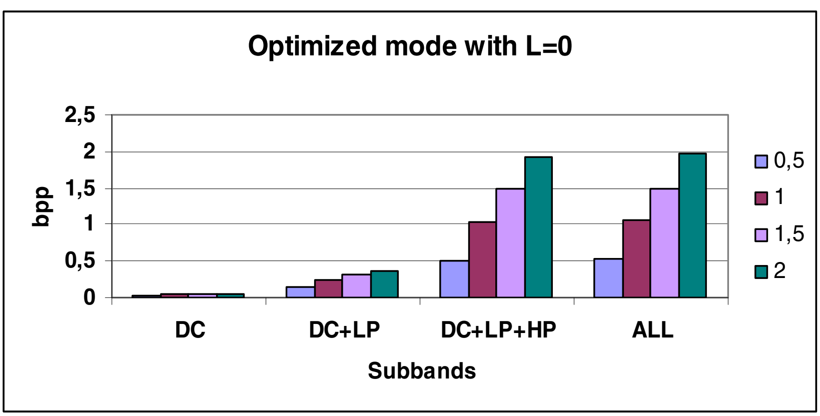


Fig. 9 (c). Bitrate for each JPEG XR subband for four target bitrates (optimized mode, L=0)

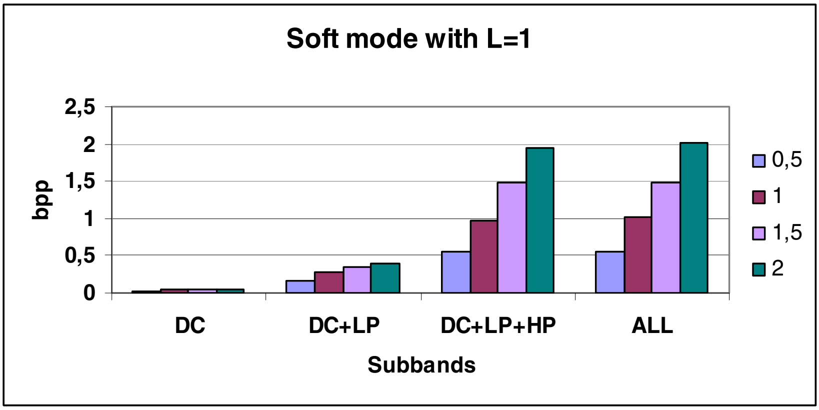


Fig. 10 (a). Bitrate for each JPEG XR subband for four target bitrates (soft mode, L=1)

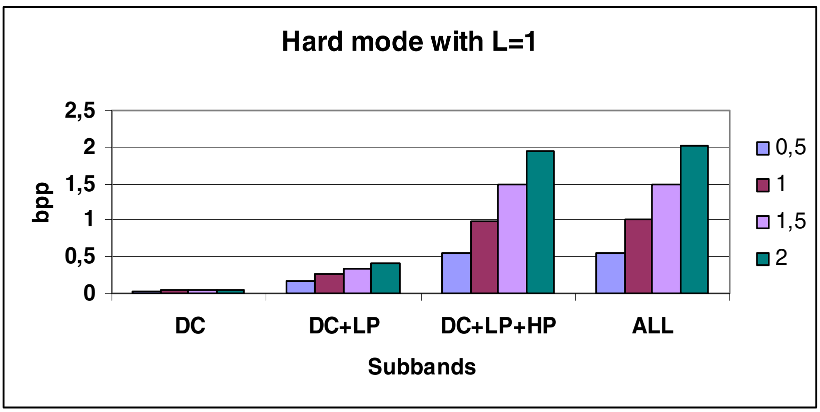


Fig. 10 (b). Bitrate for each JPEG XR subband for four target bitrates (hard mode, L=1)

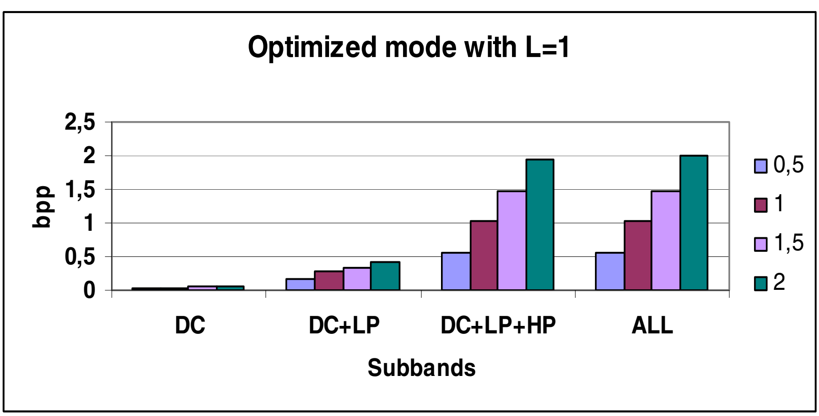


Fig. 10 (c). Bitrate for each JPEG XR subband for four target bitrates (optimized mode, L=1)

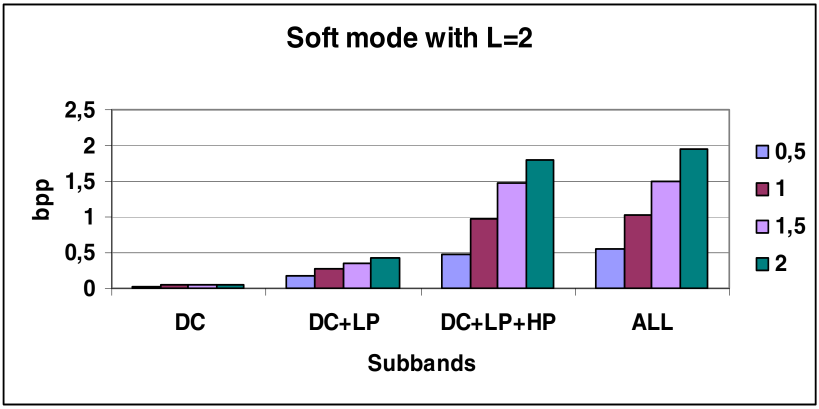


Fig. 11 (a). Bitrate for each JPEG XR subband for four target bitrates (soft mode, L=2)

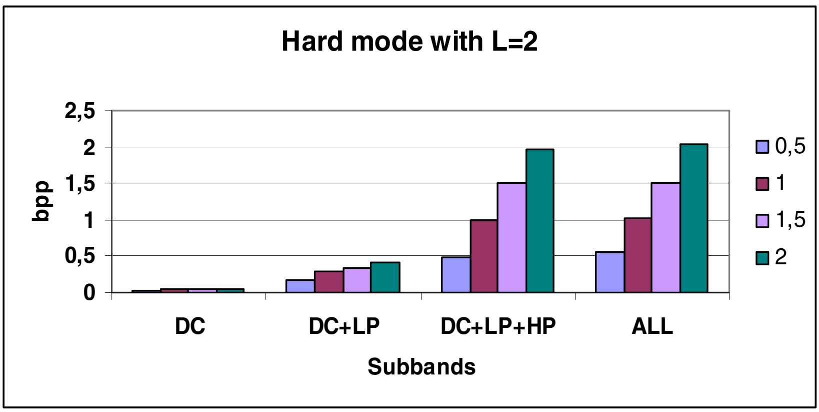


Fig. 11 (b). Bitrate for each JPEG XR subband for four target bitrates (hard mode, L=2)

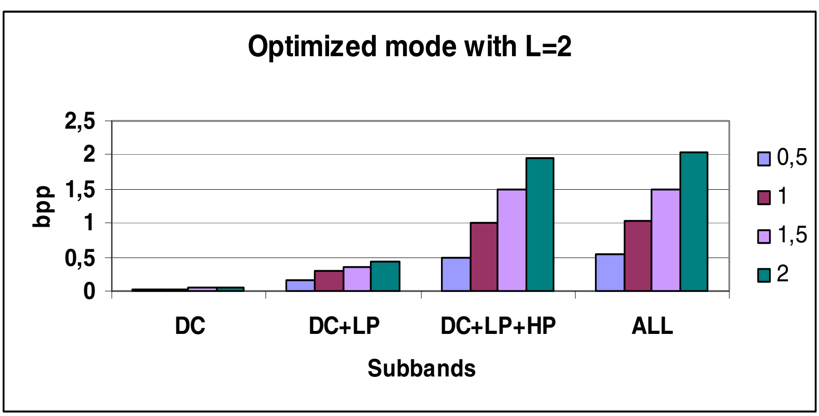


Fig. 11 (c). Bitrate for each JPEG XR subband for four target bitrates (optimized mode, L=2)

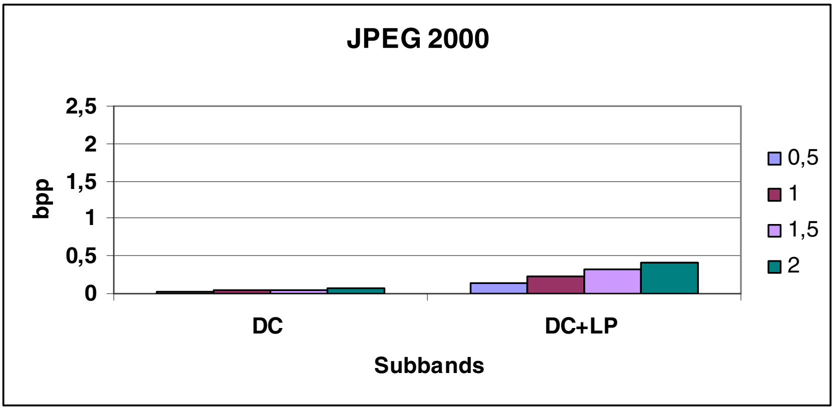


Fig. 12. Bitrate for each JPEG 2000 subband for four target bitrates

6 Conclusions

An architecture for remote browsing of large images coded using fast local access means provided by JPEG XR has been presented. In order to minimize the transferred information, the JPEG XR coded file format should make use of the frequency mode order and partitioning of images into tiles. The main goal is transmitting only some subset of the available sub-band coefficients.

This is necessary to allow an interactive access to portion of images (ROI), that are downloaded and displayed, minimizing the amount of data transferred and maintaining an acceptable image quality for the user experience.

References

1. Wallace, G.K.: The JPEG still picture compression standard. Communications of the ACM 34(4), 31–44 (1991)
2. Skodras, A., Christopoulos, C., Ebrahimi, T.: The JPEG 2000 still image compression standard. IEEE Signal Processing Magazine 18(5), 36–58 (2001)
3. ITU-T Rec. T.832 | ISO/IEC 29199-2, Information technology – JPEG-XR image coding system – Image coding specification, `http://www.itu.int/rec/T-REC-T.832`
4. Malvar, H.S.: Efficient signal coding with hierarchical lapped transforms. In: Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Albuquerque, pp. 1519–1522 (April 1990)
5. Tu, C., Sullivan, G.J., Srinivasan, S.: Effect of tile size on JPEG XR compression efficiency, ISO/IEC JTC 1/SC 29/WG 1 N4980, Japan (April 2009)
6. Hattori, K., Tsutsui, H., Ochi, H., Nakamura, Y.: A High-Throughput Pipelined Architecture for JPEG-XR Encoding. In: IEEE/ACM/IFIP 7th Workshop on Embedded Systems for Real-Time Multimedia, ESTIMedia 2009 (2009)
7. Jain, A., Panchanathan, S.: Scalable Compression for Image Browsing. IEEE Consumer Electronics 40(3) (August 1994)

8. Pecenovic, Z., Do, M.N., Vetterli, M., Pu, P.: Integrated Browsing and Searching of Large Image Collections. In: Laurini, R. (ed.) VISUAL 2000. LNCS, vol. 1929, pp. 279–289. Springer, Heidelberg (2000)
9. Li, J., Sun, H.-H.: On interactive browsing of large images. IEEE Transactions on Multimedia 5(4), 581–590 (2003)
10. Perra, C., Giusto, D.D.: An image browsing application based on JPEG-XR. In: International Workshop on Content-Based Multimedia Indexing, June 18-20, pp. 396–401 (2008)
11. Fan, X., Xie, X., Ma, W.Y., Zhang, H.J., Zhou, H.Q.: Visual attention based image browsing on mobile devices. In: Proceedings of ICME, Baltimore, MD, vol. I, pp. 53–56 (2003)
12. Chen, L.Q., Xie, X., et al.: A visual attention model for adapting images on small displays. ACM Multimedia Systems Journal 9(4), 353–364 (2003)
13. Xie, X., Liu, H., Ma, W.-Y., Zhang, H.-J.: Browsing large pictures under limited display sizes. IEEE Transactions on Multimedia 8(4), 707–715 (2006)
14. Hasan, M.A., Kim, C.: An Automatic Image Browsing Technique for Small Display Users. In: ICACT 2009: Proceedings of the 11th International Conference on Advanced Communication Technology, vol. 3 (2009)
15. Fan, X., Xie, X., Ma, W.-Y.: Detecting the sufficient display resolution for image browsing. In: 7th International Conference on Mobile Data Management (MDM 2006), p. 22 (2006)
16. Liu, H., Jiang, S., Huang, Q., Xu, C., Gao, W.: Region-based visual attention analysis with its application in image browsing on small displays. In: Proceedings of the 15th International Conference on Multimedia, pp. 305–308 (2007)
17. Tu, C., Sullivan, G.J., Srinivasan, S.: Optimizing JPEG XR tile structure for fast local access, Microsoft Corporation, Redmond, USA
18. Christopoulos, C., Skodras, A., Ebrahimi, T.: The JPEG 2000 Still Image Coding: An Overview. IEEE Transactions on Consumer Electronics 46(4), 1103–1127 (2000)
19. Dufaux, F., Sullivan, G.J., Ebrahimi, T.: The JPEG-XR image coding standard. IEEE Signal Processing Magazine 26(6), 195–199, 204–204 (2009)
20. Kakadu Software Home, `http://www.kakadusoftware.com`