Scalable Video Coding using Motion Compensated JPEG 2000

Carmelo Maturana-Espinosa, J.J. Sánchez-Hernández,
J.P. García-Ortiz and Vicente González-Ruiz

Dpto. de Arquitectura de Computadores y Electrónica, Universidad de Almería

Resumen— **This work[1] conducts a study of the efficiency of a straightforward extension of the JPEG 2000 standard, named Motion Compensated JPEG 2000 (MCJ2K), for the compression of temporally correlated sequences of images. MCJ2K is composed of two independent stages. In the first one, Motion Compensated Temporal Filtering (MCTF) is applied to each GOP of the image sequence, producing a temporal multiresolution representation and decreasing significatively the entropy. After this stage, the JPEG 2000 image compressor is used in order to efficiently encode the MCTF residuals and to incorporate spatial and quality scalability. The experimental results, compared to the obtained by other scalable (H.264/SVC) and non-scalable (H.264/AVC) video codecs, show that even without optimal bit-rate allocation between the images of a subband nor the temporal subbands, MCJ2K is quite competitive, especially for the task of the compression of high resolution image sequences.**

Palabras clave— **Motion Compensated Temporal Filtering, JPEG 2000, H.264/AVC, H.264/SVC, rate-distortion, motion estimation.**

I. Introduction

NO wadays, digital video is one of the most used media. The natural 3D structure of a video generate very high demands of memory and therefore, video compression algorithms become necessary in order to store and transmit it. For this reason, several standard video codecs has been proposed by the ISO/IEC (MPEG-*) and the ITU (H.26*). The last one of these standards, which in this case has been proposed jointly by the two organizations, is the scalable extension of the H.264/AVC standard, named H.264/SVC.

A. Video scalability

When there is some kind of flexibility in the decompression of the $E_i^{[q]}$ residuals, we say the the compressed video is scalable. Depending on the degree of freedom in this task, 3 types of scalability can be achieved:

A.1 Temporal scalability

It refeers to the posibility of decompressing only a subset of images of the code-stream, temporally equidistant. We define a temporal resolution level t as

$$V^t = \{V_{2^t \times i};\ 0 \leq i < \frac{\#V}{2^t}\} = \{V_{2i}^{t-1};\ 0 \leq i < \frac{\#V^{t-1}}{2}\}, \quad (1)$$

where $\#V$ is the number of images in V. Notice that $V = V^0$.

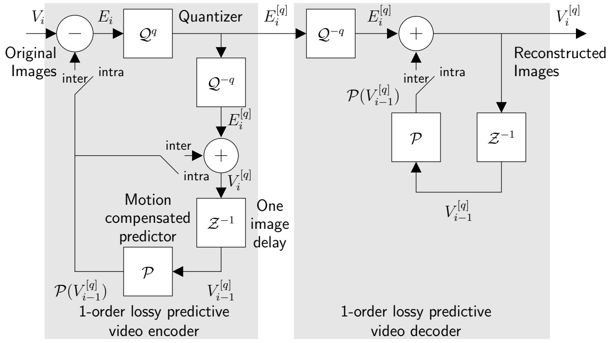

Fig. 1. Basic schema of a closed-loop lossy video codec based on motion compensation.

A.2 Spatial scalability

It refeers to the posiblity of decompressing only a reduced spatial version of the images of the code-stream. We define a spatial resolution level s as

$$V^{<s>} = \{\frac{Y}{2^s} \times \frac{X}{2^s} \text{ version of } V_i;\ 0 \leq i < \#V\}, \quad (2)$$

where X and Y are the dimensions of the images. Notice that $V = V^{<0>}$.

A.3 Quality scalability

It refeers to the posiblity of decompressing only a reduced quality version of the images of the code-stream. We define a quality level q as

$$V^{[q]} = \{\mathcal{Q}^q(V_i);\ 0 \leq i < \#V\}, \quad (3)$$

where $\mathcal{Q}$ is a quantizer (see Figure 1) applied typically in the transform domain in order to remove the least important visual information, and q is the quantization step. We define also that $V = V^{[0]}$.

B. Motion Compensated Temporal Filtering in JPEG2K

JPEG 2000 is the last image compression standard proposed by the ISO [1]. As many other image compressors, J2K is based on the scalar quantization of the pixels of a image in the wavelet domain. MCTF is the name that has the decorrelating procedure based on motion compensation which is applied to a sequence of images. In essence, is very similar to the schema presented in the Figure 1, althought there is not a loop in the encoder in order to cancel the desviation between its prediction and the ones produced at the decoder. In fact, this drift exists,

[1] This work has been funded by grants from the Spanish Ministry of Science and Innovation (TIN2008-01117) and Junta de Andalucía (P08-TIC-3518 and P10-TIC-6548), in part financed by the European Regional Development Fund (ERDF).

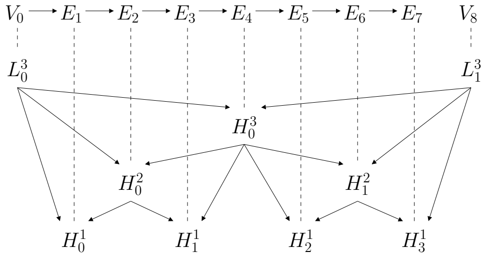

Fig. 2. Image dependencies in a closed-loop technique (up) and in MCTF (down).

but by the way the frames are processed, this drift is not accumulated along the time.

MCTF can be efficiently implemented as a dyadic cascade of low-pass (L) and high-pass (H) filters that generate a set of temporal subbands $\{L^{T-1}, H^{T-1}, \cdots, H^1\}$, where a sample in a subband is an image or a residual, depending on the subband.

To see more clearly the differences between MCTF and the closed-loop technique described, the Figure 2 shows the image dependencies produced by the two algorithms. As can be seen, in MCTF a distortion (or error) introduced in an image does not propagate linearly along the time. For this reason, the loop at the encoder can be removed and the quantization step q does not need to be known at encoding time.

The research community has investigated the Scalable Video Compression (SVC) topic for decades. During this time, a significant number of proposals have been published.

Most of the proposed SVC techniques are based on the MC 3D Discrete Wavelet Transform (DWT) [2], [3], [4], [5], [6], [7]. In all of these proposals, the temporal decorrelation (1D DWT) is performed before the spatial decorrelation (2D DWT), and for this reason, these techniques as also called t+2D video codecs. The main advantage of a t+2D algorithm is that the use of any standard MC procedure is straighfoward because it is applied on the image (non-transformed) domain. Esentially, t+2D tecniques are equivalent to MCTF+2D.

On the other hand, 2D+t techniques [8], [9], [10] can be also used. These perform the MC in the wavelet domain albeit this domain is not shift invariant and therefore, motion estimation have to be applied in the redundant DWT domain. The main advantage of a 2D+t algorithm over a t+2D is that, if motion information has also a multiresolution representation, it can be applied to each spatial resolution level of the images in order minimize the memory and computation requirements at the decoder.

Finally, although almost all algorithms are based on a open-loop schema, the MPEG selected an scalable extension of the H.264/AVC closed-loop video codec [11], named H.264/SVC [12]. In H.264/SVC, temporal scalability is performed by means of MCTF, spatial scalability using a multiresolution phyramid and quality scalability is an special case of spatial scalability where the resolutions remains constant. As a direct consecuence, the number of quality layers is dramatically reduced, but the designer also propose a *bit extractor* that increases considerably this number, to the point that the last drawback can be ignored in most of the applications.

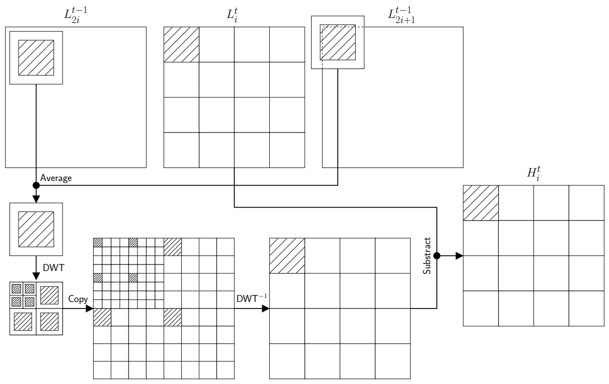

Fig. 3. The prediction step in MCJ2K.

II. Proposal

Our proposal, named MCJ2K (Motion Compensated J2K), is a pure t+2D technique which reutilizes as much as possible the functionanity of the actual J2K standard. Therefore, in this section we describe the basically the differencies with other implementations.

A. MCTF prunning

The GOP in MCJ2K has a fixed size. However, the coder checks if a residual has more o equal entropy than the original image and, if this happens, this image is not motion compensated and this information is written in the codestream.

B. Block overlaping in the prediction step

MCJ2K uses block-based motion compensation. To minimize the artifacts produced by the generation of the predictions in the image domain, the blocks can be overlaped (see Figure 3).

C. Scalability implementation

MCJ2K can produce temporal, quality and spatial codestreams. For the former two, only the order in which the J2K quality layers are decompressed need to be changed (see Figure 4). However, to achieve spatial scalability there are two options. The first one, althought is not optimal, is to use the same J2K progression order that for the quality scalability because the differences between an LRCP order and a RLCP order are, in general, small. The second one is to use the RLCP order which implies a packet reorganization at the server side (thinking in a transmission scenario where the decompressor has not a direct access to the codestream). The computational cost of this task is negligible. Finally, an important remark regarding spatial scalability is that motion comensation is always performed at the highest resolution in order to minimize the rate-distortion tradeoff. Obviously, this alternative can be inafordable in low-performance devices (such as PDAs) and

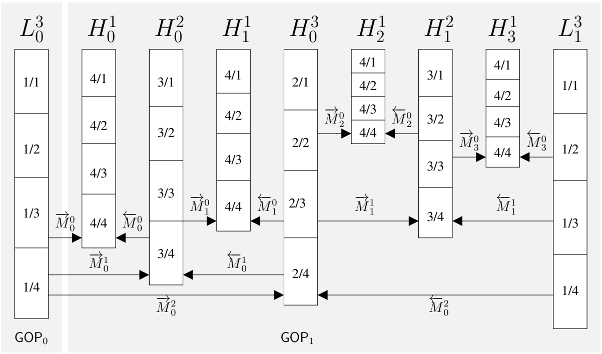

Fig. 4. Scalability in MCJ2K. The layers are descompressed using the order described in the boxes. For temporal scalability the firt number should be used and for quality/spatial scalability, se second one.

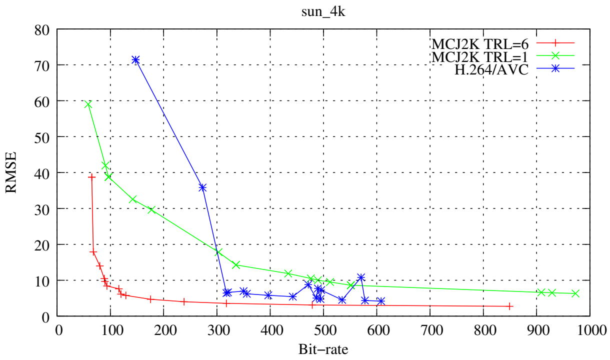

Fig. 6. Rate-distortion curves for the Sun video.

for this reason, in these cases, a multiresolution representation of the motion information can be used.

III. Evaluation

A large set of experiments have been carried out in order to find out the rate-distortion performance of MCJ2K, comparing it to H.264. The thirteen selected samples (see the table I), satisfy the requirement of resolution 4 times larger. This is enough to be representative in the encodings and bring together many features of the current reality, ranging from film to mobile devices.

A. Considerations to make a fair comparison

First, the resolution of the images have been extended in the Y direction in order to avoid simetry extension. 720p images have be extended to 736 pixels and 1080 images to 1088 pixels.

In the experiments, H.264/SVC has been used with 4 CGS layers and 4 quantization steps to get 16 quality layers. Using also the temporal layers, a total number of 20 layers has been created. Thus, as shown in the Figure 7, the amount used is about doubled Kbps to decode the layers: 4, 9, 14 and 19. The encodings are complete with post-process called Quality Layers (QL), which provides the codestream of information about quality layers without increasing its weight. This allows truncate layers at H.264/SVC, not limited to the 16 initial layers to fit a hypothetical bandwidth. The extractor of H.264/SVC with QL fits very well, although it has given poor results when we encode a single GOP.

```
Layer   Resolution   Framerate   Bitrate  MinBitrate      DTQ
  0      1920x1088      3.1250    820.80      820.80  (0,0,0)
  1      1920x1088      6.2500   1159.00     1159.00  (0,1,0)
  2      1920x1088      2.5000   1465.50     1465.50  (0,2,0)
  3      1920x1088      5.0000   1684.00     1684.00  (0,3,0)
  4      1920x1088     50.0000   1909.00     1909.00  (0,4,0)
  9      1920x1088     50.0000   4552.00              (0,4,1)
 14      1920x1088     50.0000  10640.00              (0,4,2)
 19      1920x1088     50.0000  24030.00              (0,4,3)
```

Fig. 7. Summary of the layers.

To be objective in assessing the MCJ2K[2]. codec, the codestream consists of separate files, to evaluate the performance of each. For example, to measure the efficiency of MV and headers, we have coded image sequences without textures, i.e. black images. In MCJ2K this size is directly affected by the size of macroblock. Since macroblocks MCJ2K can be very large, thus the number of MV decreases. Also if the resolution of the video is very high, the number of headers (one per frame) is less in proportion to the amount of textures. These two conditions were combined in high-resolution videos 1080p or greater. This weight compared between codec is: 3 times higher in CIF for MCJ2K, 2 in 4CIF, and 1/3 in 720p. In 1080p is 0.7 times lower than in SVC.

The use of a text compressor (PAQ8O, *http://mattmahoney.net/dc/*) has been carried out to determine the entropy contained in the encodings. This is interesting because the content that this compressor can compress, will not correspond to the textures nor the motion information. This is especially useful for H.264/SVC which returns the codestream in a single file. The results show that the codestream produced by H.264/SVC contains a little redundancy and in the codestream H.264/AVC we have found absolutely no redundancy. However, for low resolution codestreams, MCJ2K compression has been remarkable (see Figure 5). PAQ8O Is an excellent compressor although is used with the option -fast. In their documentation says that you can use specialized models for images, but we have not used it. Finally, to be more realistic, each file in a codestream was compressed independently, so that can be sent for decompression without restrictions.

B. Parameters

The implementation of MCJ2K has not yet automated compression with optimal parameters, as does SVC[3]. Therefore we have studied the behaviour of the following parameters:

1. **Size of macroblocks:** Is defined by two variables: IBS and FBS. IBS is the size of the blocks in the motion estimation process. And FBS, corresponds to the minimum block size allowed in the motion estimation. The couples of values

[2] Tests have been carried out using the Kakadu software Kakadu (*http://www.kakadusoftware.com/*)

[3] All parameters are automatic in H.264 and have no control over them.

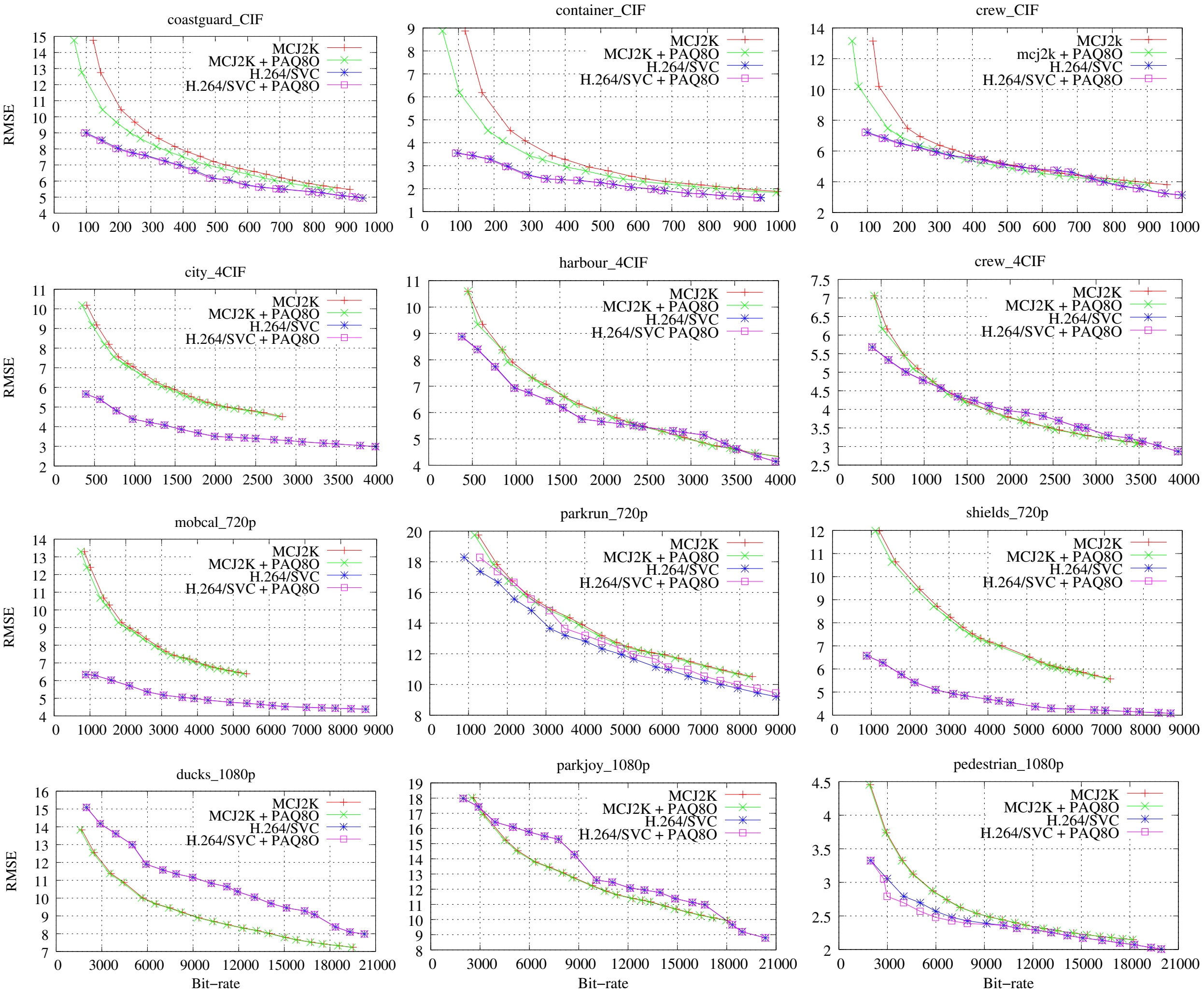

Fig. 5. Rate-distortion curves for the testing image sequences.

Name	Resolution	FPS	URL
coastguard	CIF	30	http://trace.eas.asu.edu/yuv/coastguard/coastguard_cif.7z
container	CIF	30	http://trace.eas.asu.edu/yuv/container/container_cif.7z
crew	CIF	30	http://media.xiph.org/video/derf/y4m/crew_cif.y4m
city	4CIF	30	ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/CITY_704x576_30_orig_01_yuv.zip
harbour	4CIF	30	ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/HARBOUR_704x576_30_orig_01_yuv.zip
crew	4CIF	30	ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/CREW_704x576_30_orig_01_yuv.zip
mobcal	720p	50	http://media.xiph.org/video/derf/y4m/720p50_mobcal_ter.y4m
parkrun	720p	50	http://media.xiph.org/video/derf/y4m/720p50_parkrun_ter.y4m
shields	720p	50	http://media.xiph.org/video/derf/y4m/720p50_shields_ter.y4m
ducks	1080p	50	http://media.xiph.org/video/derf/y4m/ducks_take_off_1080p50.y4m
parkjoy	1080p	50	http://media.xiph.org/video/derf/y4m/park_joy_1080p50.y4m
pedestrian	1080p	25	http://media.xiph.org/video/derf/y4m/pedestrian_area_1080p25.y4m
sun	4k	30	http://delphi.nascom.nasa.gov/jp2/AIA/171/2012/01/01/

TABLA I

Video sequences used in the experiments.

IBS-FBS used for review were: 64-64, 64-32, 64-16, 64-8, 32-32, 32-16 and 32-8. And exceptionally 16-8 and 1024-512.

2. **TRL:** The temporal resolution levels is the number of iterations of the temporal transform (see Section: I). Have been evaluated from 1 to 8.
3. **Search Range:** The search range is the size of the searching area of the motion estimation. Its default value is four and have been evaluated for 1, 2, 4, 8, 16, 32 and 64.
4. **Over-pixel:** Block overlaping is the number of overlaped pixels between the blocks in the motion estimation. Their values were from 0 to 2.
5. **Sub-pixel:** This parameter is called subpixel accuracy. This specifies the accuracy of the MVs. For example, a value of 1 means that each vector can point to a specific position within a pixel, with an accuracy of 1/2. We have evaluated our experiemtos from 0 to 3.
6. **Antialiasing:** This filter enhances the recon-

structions from the point of subjective visual quality but expensive for values. Therefore always remains off.

7. **filters for DWT:**
   (a) **Haar[13]:** In MCJ2K this operation is used for two purposes: the progressive calculation of the MVs and to rescale textures. The calculation of the MVs is always done with Haar.
   (b) **5/3** and **Daubechies 13/7:** This formulation [14] is based on the use of recurrence relations to generate progressively finer discrete samplings of an implicit mother wavelet function; each resolution is twice that of the previous scale.

C. How to evaluate the coding

The Figures 5 and 6 illustrate the characteristics of the compressibility for each codec are represented by curves R/D, where R is the compression ratio or bit-rate, drawn on the X axis in the unit of Kbps and D is the suffered distortion, expressed as the root mean square error (RMSE) between the original sample and the reconstruction, in Y. It is important to note that these characteristics on the results of each encoding, are rooted in the inherent characteristics of the processes carried out by each codec. A good curve is closer to the center of coordinates.

The las procedure is appropriate since it is not possible to scan a wide range of test encodings for two reasons: the first is that the encoding of some codecs is extremely costly in computation time and the second is that due to the variety of resolutions, the range in which they would have to look for is enormous. It is necessary to establish what amounts should be sought. Thus, for example, is set for the CIF resolution is suitable from 100 to 300 Kbps, which are found by adjusting the input parameters are suitable as the quantizations in each layer. This reference is performed extrapolated to any other video size, resulting that the 720p containing 921,600 pixels obtain a index on CIF of 9.09 times.

D. Parameter dependencies

To find the optimal values for each parameter, there have been an average of 50 tests per level of quantization, which is a minimum of 150 tests per sample, just for the codec MCJ2K. In order to compare the performance of chest compressions, using curves of at least three points in a wide range. This amount of tests could have been much higher because of the long duration of some of the test. The way to reduce the number of tests has been to discern the parameters dependencies to test multiple parameters simultaneously. Parameters have been evaluated in groups of macroblocks progressive size and TRL, because one of them clearly affects the other.

E. Results

1. **CIF:** It's where we are furthest from the good coding SVC, due to poor coding of headers and MV excessive weight. This is consistent with the high compression of plain text that can be done, yet still better SVC. The behavior improves and tends to equal to SVC in high ranges
2. **4CIF and 720p:** MCJ2K has improved markedly compared with a lower resolution. In fact the entropy of the codestream is greater as reflected PAQ8O.
3. **1080p:** From this resolution, MCJ2K compressor is usually better, although there is no clear winner. For example, for the video "pedestrian" SVC wins, even with its encoding more data correlated from codestream size of 3k Kbps, this correlation decreases progressively up to 10k. This shows that in this particular video has a better coding and is even improved.
4. **4k:** The AVC codestream shows a irregular distortion with lower bandwidth to 600 Kbps from here stabilizes. This indicates that probably the way decisions of AVC codec does not have a linear behavior. The encodings for the AVC codec have been demanding certain Kbps. Specifically ask for codestreams from 50 Kbps to 800 Kbps, 50 Kbps steady increase, although as we shall see returned encodings do not correspond exactly to those bandwidths, in fact, not even are equidistant. The order was the type of encoding:

```
x264 --bitrate 100 --sar 4096:4096 --fps 30.0 --frames 32
-o x264_100.avi sun_4kx30x420x129.yuv
```

Now look in Figure 6 at the maximum compression of both codecs and the progression of the codestream according to the bandwidth. Regardless of the bitrate requirements of the instruction, the maximum compression, which could make the AVC video is to 147.3 Kbps with an error of 71 RMSE. The inflection point is in the 270 Kbps when the distortion decreases dramatically to 35.8 RMSE, although MCJ2K, is still better.

This rate of 270 Kbps is obtained by requiring a bit-rate of 100 Kbps in the instruction encoding, if required less then always returns the codestream that was once talked of 71 RMSE. However, MCJ2K has been much more flexible, being able to compress the video to a maximum of only 65 Kbps with a distortion of 38.7 RMSE. It is also important to note that the maximum slope of the curve R/D is much more inclined in MCJ2K than in AVC, i.e. 67.9 Kbps obtain a distortion of 17.9 RMSE. The value needed was a macroblock size of very large, we chose 1024, but the results are similar to 512 or 2048. And a temporal resolution level too high, in fact, the maximum length of this video, in this case 6. Remember that the TRL is the number of levels that are structured correlations between frames, so that the greater TRL more exploits the correlation between images and not only between adjacent images. It is clear that this structure has a cost but is offset to a certain level of temporal correlation, common in all sequences. It is clear that the correlation decreases in direct proportion to the distance between frames. It is therefore useful parameter FBS, mentioned above, decreasing progressively his size. In the Figure 6 can see the difference between use a TRL high or low is very relevant. In the tests statistically the best valued for length of 129 frames was around five.

Finally we can strongly say that MCJ2K compression capability has improved according to encode video larger. In this case, AVC does not exceed the MCJ2K to bit-rate ranges of the order of Gbps, probably by working with integers, and can get to restore the video without distortion.

IV. Conclusions and future work

One of the weaknesses of MCJ2K has been the video encoding low resolution. This is due in large part to the excessive weight have the header of each frame. This header is repeated on each image even though such information is constant. This is evidenced by the high compression that can make a generic text compressor in low resolution encodings, where the vector and headers size is more significant. This is due to the use of kakadu (external software and closed). Fortunately, we can change it easily MCJ2K thanks to modular programming, however you need a study on what to choose textures compressor.

The size of the macroblocks is very important to make the most of the correlation between frames, and the weight vectors seem even more justified, in fact the best values of TRL, have been around the value 5. This can be clearly seen in Figure 6, which represents the MCJ2K compression capacity to a value of TRL=6 or 1. Generally, it holds that the best compression, corresponds to greater TRL, since there is a greater likelihood that the macroblocks are reused. Compared to H.264, the codec MCJ2K still better, even at TRL=1 for a range of rate small or medium.

On one hand, the size of the macroblocks of MCJ2K, can become very large, and therefore the number of MV decreases. On the other hand, if the video resolution is high, the number of headers (one per frame), is smaller in proportion to the increase in the number of textures. And finally, what happens is that these two situations occur together in high-resolution video, such as FULL-HD or greater. Therefore, comparisons with H.264/SVC, including H.264/AVC, improve with large videos. Specifically, in comparison with AVC, is demonstrated better a scalability of MCJ2K, always giving a linear response in the encodings, without great leaps in distorion.

The large video perform well in all ranges, a line of future work would improve video encoding small, focused on mobile devices. So, ignoring that it is possible to select another texture compressor, we will address the problem rethinking how vectors and headers are encoded. In particular one may regard information as we removed the textures for a higher compression, we can remove information vectors, because it is harvested in small band widths. One solution would be scalable vectors, so that by sending fewer data, in this situation wasted, so could send other useful information.

Referencias

[1] The Joint Photographic Experts Group, *ISO/IEC 15444-1 (JPEG2000, Part 1).*

[2] J. R. Ohm, “Three-dimensional subband coding with motion compensation,” *IEEE Transactions on Image Processing*, vol. 3, pp. 559–571, 1994.

[3] Seung-Jong Choi and John W. Woods, “Motion-compensated 3-d subband coding of video,” *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 155–167, 1999.

[4] B. Pesquet-Popescu and V. Bottreau, “Three-dimensional lifting schemes for motion compensated video compression,” in *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, vol. 3, pp. 1793–1796.

[5] Lin Luo, Jin Li, Shipeng Li, Zhenquan Zhuang, and Ya-Qin Zhang, “Motion compensated lifting wavelet and its application in video coding,” in *IEEE International Conference on Multimedia and Expo*, 2001, pp. 365–368.

[6] A. Secker and D. Taubman, “Lifting-based invertible motion adaptive transform (limat) framework for highly scalable video compression,” *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1530–1542, December 2003.

[7] Guillaume Boisson, Edouard François, and Christine Guillemot, “Accuracy scalable motion coding for efficient scalable video compression,” in *Proceedings of 11th IEEE International Conference on Image Processing, ICIP’2004*, Singapore, October 2004.

[8] Yiannis Andreopoulos, Mihaela van der Schaar, Adrian Munteanu, Joeri Barbarien, Peter Schelkens, and Jan Cornelis, “Fully-Scalable Wavelet Video Coding Using In-Band Motion Compensated Temporal Filtering,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’03)*, April 2003, vol. 3, pp. 417 – 420.

[9] J.E. Fowler, S. Cui, and Y. Wang, “Motion compensation via redundant-wavelet multihypothesis,” *IEEE Transactions on Image Processing*, vol. 15, pp. 3102–3113, 2006.

[10] Guillaume Boisson and Edouard François, “Removing redundancy in multi-resolution scalable video coding schemes,” in *Proceedings of 13th IEEE International Conference on Image Processing, ICIP’2006*, Atlanta, GA, October 2006.

[11] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra, “Overview of the h.264/avc video coding standard,” *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 13, no. 7, 2003.

[12] Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, “Overview of the scalable video coding extension of the h.264/avc standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[13] Alfred H., “Zur theorie der orthogonalen funktionensysteme,” *Mathematische Annalen 69 (3)*, vol. 4, pp. 331–371, 1910.

[14] Jensen, “La cour-harbo,” *Ripples in Mathematics*, vol. Berlin: Springer, pp. 157–160, 2001.