Projet PERSEE
« Schémas Perceptuels et Codage vidéo 2D et 3D »
n° ANR-09-BLAN-0170

Livrable **D3.2** *31/03/2011*

2D coding tools implemented in Matlab/C/C++ (intermediate version of the software)

Vincent	RICORDEL	IRCYNN
Christine	GUILLEMOT	IRISA
Laurent	GUILLO	IRISA
Olivier	LE MEUR	IRISA
Marco	CAGNAZZO	LTCI
Béatrice	PESQUET-POPESCU	LTCI

Contents

1 Introduction

In this report we describe the algorithms and the tools developed for the task 3 within the PERSEE project. The target of this task is to propose a new representation and encoding method of the "classical" (i.e. 2D) video signal, taking into account the perceptual quality of the reconstructed signal. Moreover this task is being developed keeping in mind the desired compatibility with the representation of 3D video signals.

The programme of this task provides a preliminary analysis of the state of the art in 2D video coding, which has been performed during the first year, in parallel with the implementation of the first coding tools. Particular attention has been devoted to the most recent and on-going video standards, that are the subjects of section 2. At the end of this phase, several coding tools have been retained since they allow to improve the performance of video compression. In particular, the partners of the PERSEE project designed, implemented and tested methods for:

- Dense motion estimation, section 3.1
- Intra prediction based on generalized template matching, section 3.2
- Adaptive and directional transforms, section 3.3
- Exemplar-based inpainting, section 3.4
- Visual attention modeling, section 3.5

Finally, in section 3.6 we present a new idea about how taking into account the perceptual impact of geometric distortion due to the compressed representation of motion vectors. Section 4 shows the experimental results obtained for the proposed methods.

The introduction of new coding tools related to perceptual quality must be done in the framework of a complete video codec, otherwise the testing and the comparison of these tools would not be fair. Some of the necessary building blocks of a complete video codec (e.g. the arithmetic encoder) are nevertheless difficult to implement, and they impose a set of problems that are not directly related to the targets of the project (e.g. syntax, implementation efficiency, and so forth). Therefore the partners preferred to use some existing competitive platform and to concentrate their efforts to the really innovative tools. In particular, the implementations of recent video standards such as H.264 [41] was a natural choice as starting point. However, some tools have been tested independently from a specific platform. This is motivated by the fact that some tools are quite independent from the actual framework or that the integration effort into an existing platform would be not really necessary and very demanding in terms of time and resources.

However, the different solutions do not impair the objectives of the projects, since the partners are not tied to respect to syntax of the reference codecs, nor to limit the coding tools to those used in the standards. Rather, the selected software platforms should be seen as a ignition point from which the partners become able to develop innovative solutions.

The choice of the specific software platform was an important issue as well. As explained in section 2, the partners used the joint model (JM) implementation of H.264 and the HVC implementation of the next standard from ITU/ISO (unofficially named H.265). This double choice is related to the fact that at the beginning of the project, only the JM software was publicly available, and the H.265 standardization was just beginning (no software model existed). When the software model for H.265 became available, some of the parteners decided to reimplement their tools within the new platform, while others observed that the implementation of the methods in H.264 was a sufficient quantitative validation for the proposed methods.

The rest of this document is organized as follows. Section 2 describes the software platforms used within task three of the project. In particular, the JM and the HM models are described, and a brief historical of the software evolution within the standardization process is given as well. Section 3 resumes the scientific contributions, namely those related to the dense motion estimation, to the generalized template matching, to the use of adaptive transforms, to the exemplar-based matching, to the visual attention models and to the new geometric-distortion-based approaches. Finally section 4 provides the experimental results.

2 Software architecture description

2.1 From the JM to the HM : a brief history

Two organizations for standardization are involved in defining new video standards. They are the International Organization for Standardization (ISO) and the International Telecom Union (ITU-T). They both have a group dedicated to video compression. They are respectively the Motion Picture Expert Group (MPEG) and the Video Coding Expert Group (VCEG). In December 2001, MPEG and VCEG created a common group, the Joint Video Team (JVT), in order to specify a new video format H264/AVC. In 2004, the version of this new standard, MPEG-4p10/AVC/H264, is published. JVT had provided a Verification Model (VM), a software compatible with the new standard, especially with the new format of the bitstream, which was set once and for all. Its name is JM (Joint Model). Since that time, the JM has evolved up to the release 17.2 as new tools were integrated or bugs fixed. The syntax of the bitstream was still the same. However new tools may require a new syntax. That is why two branches, the KTA (Key Technical Area) branches, stemming from the main JM trunk were developed. The main one is the KTAv1 issued from the JM 11.0. It gathered all promising tools predicting an evolution of the current standard.

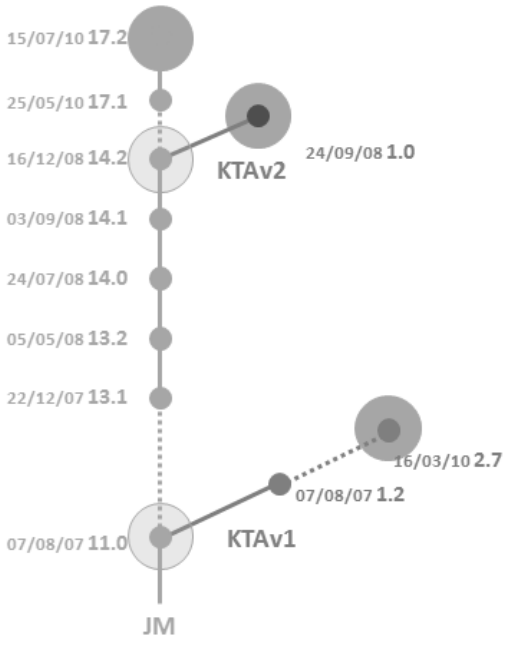


Figure 1: Tree of JM and KTA releases

In January 2010, MPEG and VCEG issued a call for proposals (CfP) to design a

new video standard, which would succeed MPEG-4/AVC/H264. Again MPEG and VCEG have created then a new joint group, which was in charge of developing this new standard: the Joint Collaborative Team on Video Coding (JCT-VC). In April 2010, the first meeting of JCT-VC had been held during which 27 proposals were tested and ranked. The best one should have been appointed as the very first version of a verification model candidate. However, as performances of the five first proposals were quite similar, it was decided to create a Test Model under Consideration (TMuC). It was initiated with the source code of the joint Samsung/BBC's proposal. Tools were then integrated, evaluated and kept or removed. Step by step, the TMuC has evolved up to the release 0.9. This release became the very first version of the test model in October 2010. It was named HM1.0, HM for HEVC test Model. Several Core experiment (CE) were and still are set in order to evaluate how relevant new tools are once integrated in the HM. Hence, the HM is still evolving and its new release will be HM3.0, which will be available before the next JCT-VC meeting in July 2011.

2.2 Description of 2D video codec HEVC Model

The 2D video codec used within the Persee project is based on the HEVC test Model (HM), which is currently being developed by the Joint Collaborative Team on Video Coding (JCT-VC). The following sections describe the architecture of this codec, release 2.0 (the 3.0 version will be released by mid April 2011).

2.2.1 General Coding Structure

The HM is a generalization of the H.264/AVC design. Most of its components are extension of existing ones, such as intra prediction which is now possible with larger blocks and with more directions. However most components take advantage of a new picture partitioning as described in the following sections. The figure below describes the HEVC encoder.

2.2.2 Picture Partitioning

One of the most beneficial elements for higher compression performance in high-resolution video comes due to introduction of larger block structures with flexible mechanisms of sub-partitioning. For this, the HM defines coding units (CUs) which define a sub-partitioning of a picture into regular regions of equal of (typically) variable size. The coding unit replaces the macroblock structure as known from the previous video coding standards. It contains one of several prediction units (PUs) and transform units (TUs). The basic partition geometry of all these elements is encoded by a scheme similar to the well known quad-tree segmentation structure. The following sections describe the structures and roles of these elements.

Treeblock (TB) partitioning

Pictures are divided into slices and slices are sequences of treeblocks. Each treeblock is comprised of NxN luma array and two corresponding chroma sample arrays (when

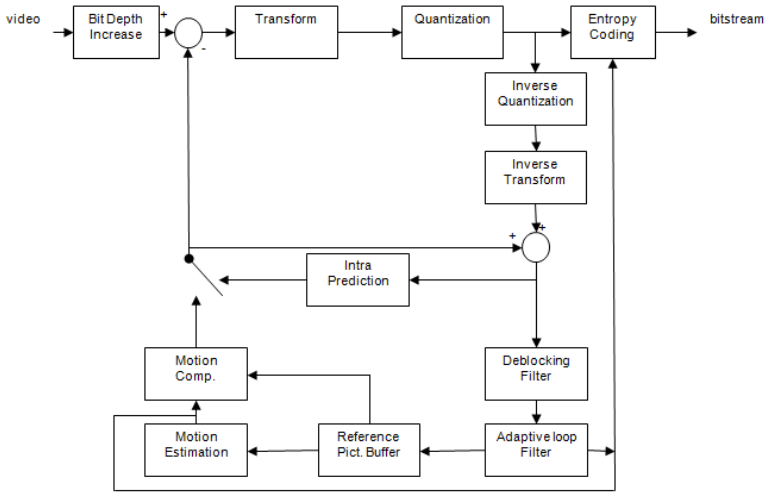


Figure 2: Architecture of the HEVC encoder

the chroma sampling format is not equal to 4:0:0). Each treeblock is assigned a partition signaling to identify the block sizes for intra or inter prediction and for transform coding. The partitioning is a recursive quadtree partitioning. The root is associated to the treeblock. The quadtree is split until a leaf is reached, which is referred to as the coding node. The coding node is the root node of two trees: the prediction tree and the transform tree. The maximum allowed size of a tree block is 64x64 luma samples.

Coding unit (CU) structure

The Coding Unit is (CU) the basic unit of region splitting used for both inter and intra prediction. It is always square and it may take a size from 8x8 luma samples up to the size of the treeblock. The CU concept allows recursive splitting into four equally sized blocks, starting from the treeblock. This process gives a content-adpative coding tree structure comprised of CU blocks, each of which may be as large as the treeblock or as small as 8x8. The flowing figure shows an example of a coding unit structure. The coding node and the associated prediction and transform units form together a coding unit.

Prediction unit (PU) structure

The prediction tree specifies the position and size of prediction blocks. The prediction tree and associated data are referred to as prediction unit (PU). A PU is the basic unit used for carrying the information related to the prediction processes (intra

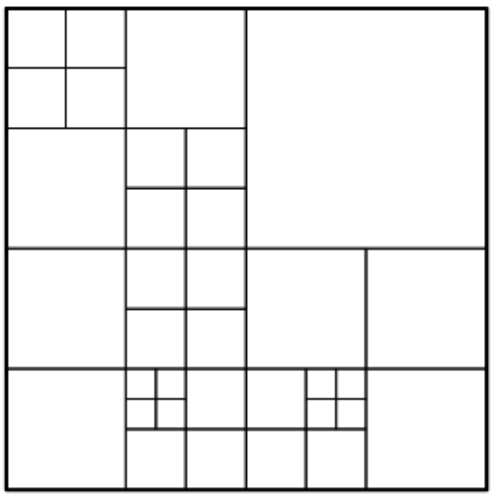


Figure 3: Example of Coding Unit structure

or inter). In general, it is not restricted to being square in shape, in order to facilitate partitioning which matches the boundaries of real objects in the picture. Each CU may contain one or more Pus. Four types of PU exist:

All these types are used for inter prediction. However, only the $2N \times 2N$ and the $N \times N$ are used for intra prediction.

Transform Unit (TU) structure

The Transform Unit (TU) is the basic unit used for the transform and quantization processes. It is always square and it may take a size from 4x4 up to 32x32 luma samples. Each CU may contain one or more TUs, where multiple TUs may be arranged in a quadtree structure, as illustrated in the figure below. The maximum quadtree depth is adjustable and is specified in the slice header syntax. For instance, the high efficiency configuration uses 3-level quadtree.

2.2.3 Intra prediction

At the level of PU, intra-prediction is performed from samples already decoded adjacent Pus, where the different modes are DC (flat average) or one of up to 33 angular directions. The total number of available prediction modes depends on the size of the corresponding PU, as shown in the table below.

The 33 possible intra prediction directions are illustrated in figure below:

For PU sizes where less than the full set 34 total intra prediction mode are allowed, the first N directions according to the mapping between the intra prediction direction and the intra prediction mode number specified in the figure below are used.

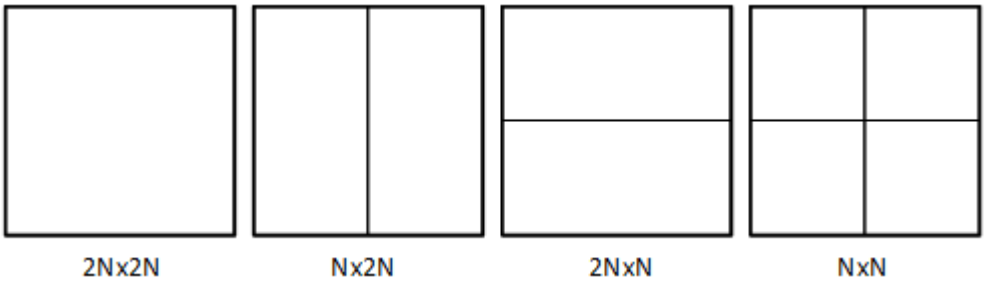


Figure 4: Four types of Prediction Unit structure

PU Size	Number of intra modes
4	17
8	34
16	34
32	34
64	3

Table 1: Number of supported intra modes according to PU size

2.2.4 Inter prediction

Inter-picture prediction is performed from region(s) of already decoded pictures stored in a reference picture buffer (with a prediction dependency independent of display order, as in AVC). This allows selection among multiple reference pictures, as well as bi-prediction from two reference pictures or two positions in the same reference picture. The reference area is selected by specifying a motion vector displacement and a reference picture index. For efficient encoding, skip and direct modes similar to the ones of AVC are defined and derivation of motion vectors from those of adjacent PUs is performed by a new scheme referred to as advanced motion vector competition (AMVP). The AMVP is an adaptative motion vector prediction technique that exploits spatio-temporal correlation of motion vector with neighbouring PUs. It constructs motion vector candidate list by firstly checking availability of left, top and temporally co-located PU positions and then removing candidates as a normative process. Then, the encoder can select the best predictor from the candidate list and transmits corresponding index indicating chosen candidate. The AMVP is used for prediction of all coded motions vectors and for derivation of skip motion vectors.

Another technique, the motion merge technique is available. The motion merge technique is to find neighbouring inter coded PU such that its motion parameters (motion vector, reference picture index and prediction direction index) can be inferred as the ones for the current PU. Similar to the AMVP technique, the encoder can

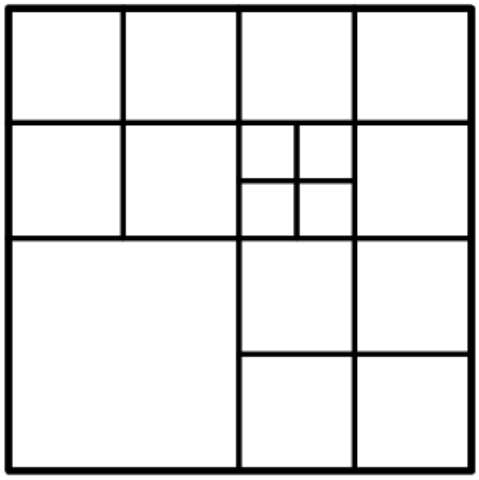


Figure 5: Example of Transform Unit Structure

also select the best candidate to be used to infer motion parameters from multiple candidates formed by spatial neighbouring PUs and temporally co-located PU, and transmits corresponding index indicating chosen candidate. This mode can be applied to any PU.

For PU types Nx2N and 2NxN, the first PU in the decoding order is inferred to use merging mode. Both merging and motion vector difference coding modes are supported in the other PUs.

2.2.5 Transform and quantization

At the level of the TU (which typically would not be larger than the PU), an integer spatial transform similar in concept to the DCT is used, with a selectable block size ranging from 4x4 to 32x32. For the directional intra modes, which usually exhibit directional structures in the prediction residual, special modes derived from the mode dependent directional transforms (MDDT) are employed. PUs may be split into several TU and form the residual quadtree (RQT). The same scaling and quantification method as in H.264/AVC is used, with scaling matrices added for the transform sizes of 16x16 and 32x32. The transform coefficient scan is intra mode dependent (MDCS) and can be a zig-zag, horizontal or vertical scan.

2.2.6 Entropy coding

The HM defines two context-adaptive entropy coding schemes, one for operation in a higher complexity mode and for lower complexity mode. The higher complexity mode uses a binarization and context adaptation mechanism similar to the CABAC entropy

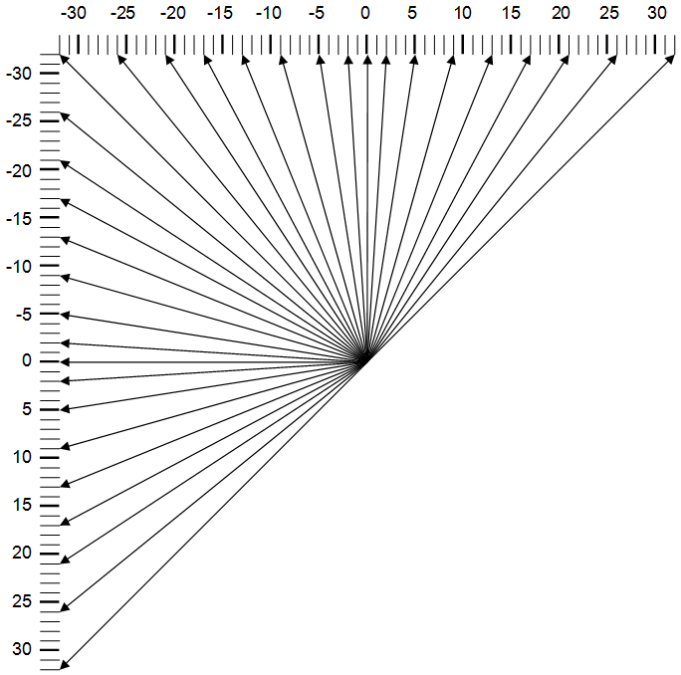


Figure 6: The 33 intra prediction directions

coder of AVC but uses a set of variable-length-to-variable-length codes (mapping a variable number of bins into a variable number of encoded bits) instead of an arithmetic coding engine. This is performed by a bank of parallel VLC coders — each of which is responsible for a certain range of probabilities of binary events (which are referred to as bins). While the coding performance is very similar to CABAC, it can be better parallelized and has higher throughput per processing cycle in a software or hardware implementation. The low complexity scheme, aka low complexity entropy coding (LCEC), is based on a variable length code (VLC) table selection for all syntax elements (based on eitherfixed-length code or exponential Golomb code as appropriate), with a particular code table that is selected in a context dependent fashion based on previous decoded values. This is similar in concept to the CAVLC scheme from AVC, but allows even simpler implementation due to the more systematic structure.

2.2.7 Loop filtering

Two kinds of filters are used for loop filtering: deblocking filter and adaptive loop filter (ALF). At the level of a CU, it is possible to switch on an adaptive loop filter (ALF) which is applied in the prediction loop prior to copying the frame into the reference picture buffer. This is a finite impulse filter (FIR) filter which is designed with the

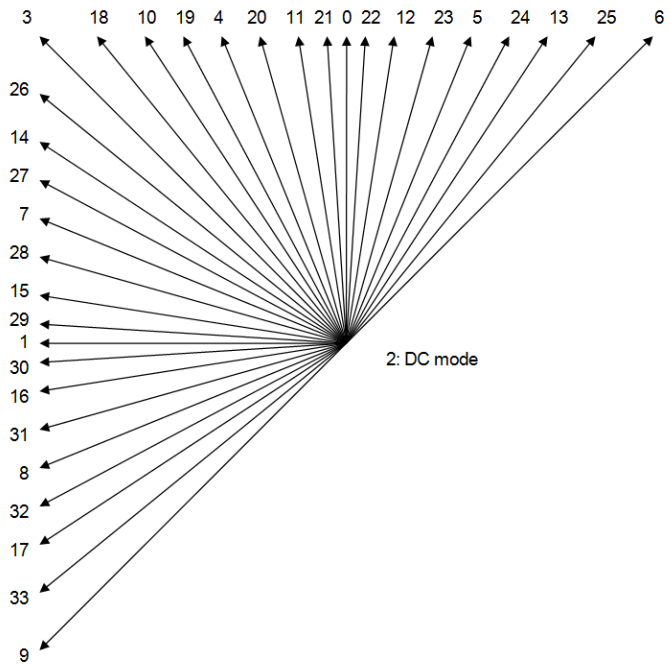


Figure 7: Mapping between intra prediction direction and intra prediction mode

goal to minimize distortion relative to the original picture (e.g; with a least-squares or Wiener filter optimization). For luma samples in each CU, the encoder makes a decision on whether or not the adaptive loop filter is applied and the appropriate signaling flag is included in the slice header. Filter coefficients are encoded at the slice level. The filter coefficient for each pixel is selected from multiple filters by computing the variance measure. Three filter sizes of 5x5, 7x7 and 9x9 are supported, but the maximum vertical difference for the current pixel is restricted from -3 to 3 inclusive as illustrated in the following figure.

In addition, a deblocking filter (similar to the deblocking filter design in AVC) is operated within the prediction loop. The display output of the decoder is written to the decoded picture buffer after applying these two filters.

2.2.8 Internal bit depth increase (IBDI)

IBDI (Internal Bit Depth Increase) is a coding tool that increases the bit depth of input picture at encoder side and decreases the bit depth of output picture to input bit depth. By increasing internal bit depth, the accuracy of all internal processes is increased. This intends to reduce the rounding error of intra-fame prediction, spatial

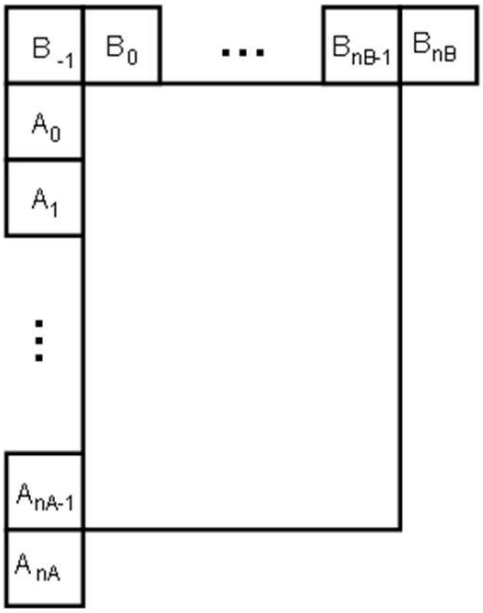


Figure 8: Spatial motion vector candidates for AMVP

transform, in-loop filtering in order to improve coding efficiency. IBDI is activated only the high efficiency (HE) configuration. 2 bits for 8-bit luma sample are added for internal precision.

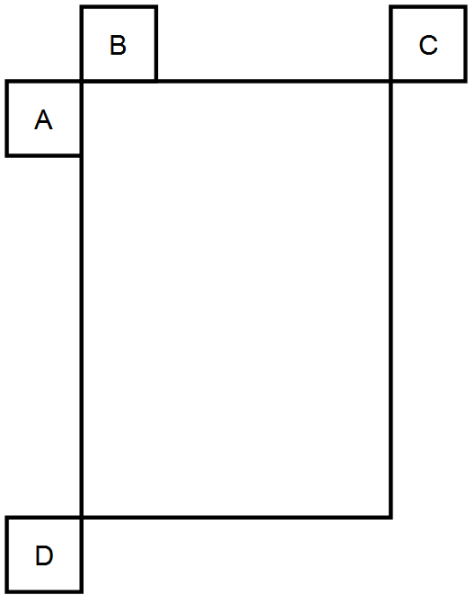


Figure 9: Spatial candidates for Motion Merge

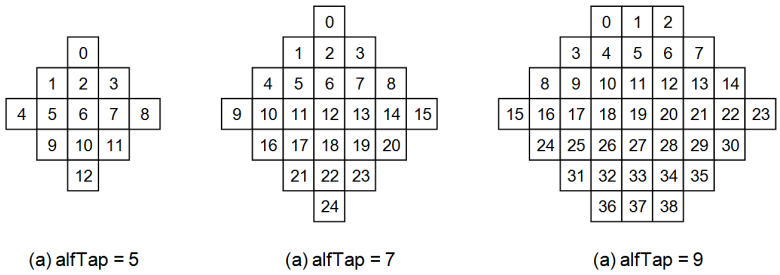


Figure 10: Filter shape for luma samples according to the filter length

3 Proposed tools

3.1 Pel-recursive motion estimation

The most common algorithms for motion estimation (ME) in the context of video compression are based on the matching of blocks of pixels (block matching algorithms, BMA). However, this is not the only solution: gradient and pel-recursive (PR) techniques, have been developed for video analysis and they solve the optical flow problem using a differential approach [13]. These methods produce a dense motion vector field (MVF), which does not fit into the classical video coding paradigm, since it would demand an extremely high coding rate. On the contrary, it is quite well suited to the distributed video coding (DVC) paradigm, where the dense MVF is estimated only at the decoder side, and it has been proved that the pel-recursive method for motion estimation introduced by Cafforio and Rocca [3–5] improve the estimation quality of missing frames (Wyner-Ziv frames in the context of DVC) [6, 7].

Starting from this observation, we have developed a way to introduce the Cafforio-Rocca algorithm (CRA) within the H.264 encoder, and in general in a hybrid video encoder. We can therefore provide an alternative coding mode for Inter macroblocks (MB). In particular we used the JM V.11 KTA1.9 implementation, but the produced software is readily usable in newer version of the software. The new mode is encoded exactly as a classical Inter-mode MB, but the decoder is able to use a modified version of the CRA and then to compute a motion vector per each pixel of the MB. The motion-compensated prediction is then more accurate, and there is room for RD-performance improvement.

In the following we describe the original version of the CRA, the modification needed to use it into an hybrid encoder, the practical issues related to H.264 implementation, and finally we show the obtained performances.

3.1.1 The original algorithm

The CR algorithm is a dense pel-recursive motion estimation algorithm: this means that it produces a motion vector (MV) for each pixel, and that previously computed vectors can be used for the initialization of the next pixel to be processed. When applying the original CRA, we suppose that we have the current image, indicated as I_k, and a reference one, indicated as I_h. This reference can be the previous ($h = k-1$) or any other image in the sequence.

More precisely, once a proper scanning order has been defined, the CRA consists in applying for each pixel $\mathbf{p} = (n, m)$ three steps, producing the output vector $\widehat{\mathbf{v}}(\mathbf{p})$.

1. **A priori estimation.** The motion vector is initialized with a function of the vectors which have been computed for the previous pixels. For example, one can use the average of neighboring vectors. However, a different initialization is needed for the first pixel: it can be for example a motion vector computed with a block matching algorithm (BMA). The result of this step is called *a priori* vector, and it is indicated as $\mathbf{v}^0$.

2. **Validation.** The *a priori* vector is compared to the null vector. In particular, we compute

$$A = \left|I_k(\mathbf{p}) - I_h(\mathbf{p} + \mathbf{v}^0)\right|$$
$$B = |I_k(\mathbf{p}) - I_h(\mathbf{p})| + \gamma$$

If the prediction error for the current pixel is less than the one for the null vector (possibly incremented by a a positive quantity γ), – that is, if $A < B$ – the a priori is retained as validated vector: $\mathbf{v}^1 = \mathbf{v}^0$; otherwise, the null vector is retained, that is $\mathbf{v}^1 = \mathbf{0}$.

3. **Refinement.** The vector retained from the validation step, $\mathbf{v}^1$, is refined by adding to it a correction $\delta\mathbf{v}$. This correction is obtained by minimizing the energy of first-order approximate prediction error, under a constraint on the norm of the correction. A few calculations show that this correction is given by:

$$\delta\mathbf{v}(n,m) = \frac{-e_{n,m}}{\lambda + \|\boldsymbol{\varphi}_{n,m}\|^2}\boldsymbol{\varphi}_{n,m} \tag{1}$$

where λ is the Lagrangian parameter of the constrained problem; $e_{n,m}$ is the prediction error associated to the MV $\mathbf{v}^1$, and $\boldsymbol{\varphi}$ is the spatial gradient of the reference image motion-compensated with $\mathbf{v}^1$.

In conclusion, for each pixel $\mathbf{p}$, the output vector is $\widehat{\mathbf{v}}(\mathbf{p}) = \mathbf{v}^1 + \delta\mathbf{v}$.

3.1.2 Adapting the Cafforio-Rocca algorithm in a hybrid coder

The basic idea is to use the CRA to refine the MV produced by the classical BMA into an H.264 coder. This should be done by using only data available at the decoder as well, so that no new information has to be sent, apart from some signalling bits to indicate that this new coding mode is used. In other words, the new mode (denoted as CR mode) is encoded exactly like a standard Inter mode, but with a flag telling the decoder that the CRA should be used to decode the data. The operation of the new mode is the following.

At the encoder side, first a classical Inter coding for the given partition is performed, be it for example a 16×16 partition. The encoded information (motion vector and quantized transform coefficient of the residual) is decoded as a classical Inter16×16 MB, and the corresponding cost function $J_{\text{Inter}} = D_{\text{Inter}} + \lambda_{\text{Inter}} R$ is evaluated. Then, *the same encoded information* (namely the same residual) is decoded using the CR mode. The details about the CR mode decoding are provided later on; for the moment we remark that we need to compute the cost function J_{CR} associated to the mode, and that when all the allowed coding modes have been tested, the encoder chooses the one with the smallest cost function. In particular, if the CR mode is chosen, the sent information is the same it would transmit for the Inter mode, but with a flag signalling the decoder to used the CRA for decoding.

Now we explain how the information in a Inter MB can be decoded using the CRA. When we receive the Inter MB, we can decode the motion vector and the residual, and

we can compute a (quantized) version of the current MB. Moreover, in the codec frame buffer, we have a quantized version of the reference image. We use this information (the Inter MV, the decoded current MB and the decoded reference image) to perform the modified CRA. This is done as follows[1]:

A priori estimation. If the current pixel is the first one in the scan order, we use the Inter motion vector, $\mathbf{v}^0(\mathbf{p}) = \mathbf{v}_{\text{Inter}}(\mathbf{p})$. Otherwise we can initialize the vector using a function of the already computed neighboring vectors, that is $\mathbf{v}^0(\mathbf{p}) = f\left(\{\widehat{\mathbf{v}}(\mathbf{q})\}_{\mathbf{q}\in N(\mathbf{p})}\right)$. We could also initialize all the pixels of the block with the Inter vector, but this results less efficient.

Validation. We compare three prediction errors and we choose the vector associated to the best one. This is another change with respect to the original CRA, where only 2 vectors are compared. First, we dispose of the quantized version of the motion compensated error obtained by first step. Second, we compute the error associated to a prediction with the null vector. This prediction can be computed since we dispose of the (quantized) reference frame, and of the (quantized) current block, decoded using the Inter16 mode. This quantity is possibly incremented by a positive quantity γ, in order to avoid unnecessary reset of the motion vector. Finally, we can use again the Inter vector. This is useless only for the first pixel, but can turn out important if two objects are present in the block. In conclusion, the validated vector is one among $\mathbf{v}^0(\mathbf{p})$, $\mathbf{0}$ and $\mathbf{v}_{\text{Inter}}(\mathbf{p})$; we keep as validated vector the one associated to the least error.

Refinement. The refinement formula in Eq. (1) can be used with the following modification. The error $e_{n,m}$ is the quantized MCed error; the gradient $\boldsymbol{\varphi}$ is computed on the motion-compensated reference image.

Now we discuss the modification made on the CRA, and how the impact on the algorithm's effectiveness. First, we dispose only of the quantized version of the motion-compensation error, not of the actual one. This affects both the refinement and the validation steps. Moreover, this makes impossible to use the algorithm for the Skip mode, which is almost equivalent to code the MC error with zero bits. The second problem is that we can compute the gradient only on the decoded reference image. This affects the refinement step. These remarks suggest that the CRA should be used carefully when the quantization is heavy. Finally, we observe that the residual decoded in the CR mode, is the one computed with the single motion vector computed by the BMA. On the other hand, the CRA provides other, possible improved vectors, which are not perfectly fit with the sent residual. However, the improvement in vector accurateness leaves room for possible performance gain, as shown in the next section.

Because of the complexity of the H.264 encoder, we have take into account some other specific problems, such as the management of CBP bits, the MB partition, the multiple frame reference, the mode competition and the bitstream format. Most of the solutions adopted are quite straightforward, so we do not described them here for

[1] We should first define a scanning order of pixels; we have considered a raster scan order, a forward-backward scan, and a spiral scan.

the sake of brevity. However, some other issues were not considered yet in our first implementation, namely the management of B-frames and of color information.

3.1.3 Specific modification for H.264

The problem of coding or not the second residual has been described in the case of an H264 encoder, however, it would be very similar for any other hybrid coder. However there are other problems more specific to H.264. We give here some details about them.

CBP bits. In H.264 there can exists Inter MBs with no bit for the residual. This happens when all coefficients are quantized to zero or when very few coefficients survive the quantization and so the decoder decides that it is not worth to code any of them. This is signaled very efficiently by using a few bits in the so-called CBP field. A zero in the CBP bits means that the corresponding MB has a residual quantized to zero. This allows an early termination of the MB bitstream. As for the Skip mode, a null residual makes useless the CRA. Therefore, when the Intra mode provides a null CBP, the CRA is not used. Since this situation occurs mainly at very-low bit-rates, it is understood that in this case the CRA is often not applicable.

Partition. In H.264 each MB can undergo to a different partition. So for each pixel, the initialization must care about the partition and its border, in order to not use vectors from another partition to initialize to MV of the current pixel.

Multiple frame reference. Each subblock in a MB can be compensated using any block in a set of reference frame. This means that the initialization vector must be properly used; moreover, the gradient for this subblock must be computed on the correct reference frame. In conclusion the gradient is not computed on the same image for each pixel, but it can vary from subbblock to subblock.

Interpolation H.264 has its own interpolation technique for the half-pixel precision; however we used a bilinear interpolation for the arbitrary-precision motion compensation, for the sake of simplicity.

Mode competition. The new coding mode is implemented in the framework of a RD-optimized coding mode selection. So we computed the cost function $J = D + \lambda R$ for the new mode as well. The new mode is used only if it has the least cost function value.

Bitstream format. In the case that no new residual is computed (A coder), the H.264 bitstream is almost unchanged: only a bit is added in order to discriminate between Inter and CR modes. On the other hand, when the B coder is used, besides the CR flag, when the CR mode is selected, we have to introduce the second residual in the bitstream.

Other feature of H.264 that were **not** taken into account are the following.

B-frames The generalized B-frames requires a non trivial modification of the JM code in order to manage a CR-like mode; for the moment this possibility has not been considered (*i.e.* the CR mode is not used in B-frames).

8x8 Transform The software `JM v11 kta1.9` allows two spatial transforms, of sizes 4x4 and 8x8. The management of two transforms is not easy in the case of CRA; since the 8x8 transform is non-normative, (it is a tool among the Fidelity Range Extensions), in this first implementation we do not consider it.

Color For the sake of simplicity we did not implement the CR coding on the Chroma.

3.2 Intra prediction based on generalized template matching

Closed-loop intra prediction plays an important role in minimizing the encoded information of an image or an intra frame in a video sequence. E.g., in H.264/AVC, there are two intra prediction types called Intra-16x16 and Intra-4x4 respectively [40]. The Intra-16x16 type supports four intra prediction modes while the Intra-4x4 type supports nine modes. Each 4x4 block is predicted from prior encoded samples from spatially neighboring blocks. In addition to the so-called "DC" mode which consists in predicting the entire 4x4 block from the mean of neighboring pixels, eight directional prediction modes are specified. The prediction is done by simply "propagating" the pixel values along the specified direction. This approach is suitable in presence of contours, when the directional mode chosen corresponds to the orientation of the contour. However, it fails in more complex textured areas.

An alternative spatial prediction algorithm based on **template matching** has been described in [28]. In this method, the block to be predicted of size 4x4 is further divided into four 2x2 sub-blocks. Template matching based prediction is conducted for each sub-block accordingly. The best candidate sub-block of size 2x2 is determined by minimizing the sum of absolute distance (SAD) between template and candidate neighborhood. The four best match candidate sub-blocks constitute the prediction of the block to be predicted. This approach has later been improved in [29] by averaging the multiple template matching predictors, including larger and directional templates, as a result of more than 15% coding efficiency in H.264/AVC. Any extensions and variations of this method are straightforward. In the experiments reported in this paper, 8x8 block size has been used without further dividing the block into sub-blocks.

Here, a spatial prediction method based on sparse signal approximation (such as matching pursuits (MP) [18], orthogonal matching pursuits (OMP) [22], etc.) has been considered and assessed comparatively to the template matching based spatial prediction technique. The principle of the approach, as initially proposed in [33], is to first search for the linear combination of basis functions which best approximates known sample values in a causal neighborhood (template), and keep the same linear combination of basis functions to approximate the unknown sample values in the block to be predicted. Since a good representation of the template does not necessarily lead to a good approximation of the block to be predicted, the iteration number, which minimizes a chosen criterion, needs to be transmitted to the decoder. The

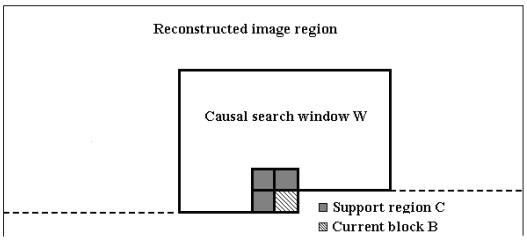


Figure 11: C is the approximation support (template), B is the current block to be predicted and W is the window from which texture patches are taken to construct the dictionary to be used for the prediction of B.

considered criteria are the mean square error (MSE) of the predicted signal and a rate-distortion (RD) cost function when the prediction is used in a coding scheme.

Note that, this approach can be seen as an extension of the template matching based prediction (which keeps one basis function with a weighting coefficient equal to 1). In order to have a fair comparison with template matching, the sparse prediction algorithm is iterated only once. In the experiments reported here, the OMP algorithm has been used by considering a *locally adaptive dictionary* as defined in [33]. In addition, both a static and MSE/RD optimized dynamic templates are used. The best approximation support (or template) among a set of seven pre-defined templates is selected according to the corresponding criterion, that is minimizing either the residual MSE or the RD cost function on the predicted block.

The proposed spatial prediction method has been assessed in a coding scheme in which the residue blocks are encoded with an algorithm similar to JPEG. The approximation support type (if dynamic templates are used) is Huffman encoded. The prediction and coding PSNR/bit-rate performance curves show a gain up to 3 dB when compared with the conventional template matching based prediction.

3.2.1 Template matching and sparse prediction

Let S denote a region in the image containing a block B of $n \times n$ pixels and its causal neighborhood C used as approximation support (template) as shown in Fig. 11. The region S contains 4 blocks, hence of size $N = 2n \times 2n$ pixels, for running the prediction algorithm. In the region S, there are known values (the template C) and unknowns (the values of the pixels of the block B to be predicted). The principle of the prediction approach is to first search for the best approximation for the known values in C, and keep the same procedure to approximate the unknown pixel values in B.

The N sample values of the area S are stacked in a vector $\mathbf{b}$. Let A be the corresponding dictionary for the prediction algorithm represented by a matrix of dimension $N \times M$, where $M \geq N$. The dictionary A is constructed by stacking the luminance values of all patches in a given causal search window W in the reconstructed image region as shown in Fig. 11. The use of a causal window guarantees that the decoder can construct the same dictionary.

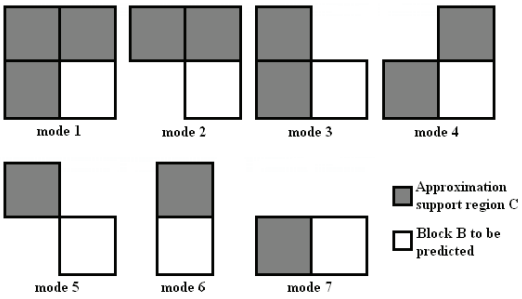


Figure 12: Seven possible modes for approximation support (dynamic template) selection. Mode 1 corresponds to the conventional static template.

3.2.2 Template matching (TM) based spatial prediction

Given $A \in \mathbf{R}^{N\times M}$ and $\mathbf{b} \in \mathbf{R}^{N}$, the template matching algorithm searches the best match between template and candidate sample values. The vector $\mathbf{b}$ is known as the *template* (also referred as the filter mask), and the matrix A is referred as the dictionary where its columns $\mathbf{a}_j$ are the *candidates*. The candidates correspond to the luminance values of texture patches extracted from the search window W.

The problem of template matching seeks a solution to minimization of a distance d as

$$\underset{j\in\{1\ldots M\}}{\arg\min}\ \{d_j : d_j = \mathrm{DIST}\,(\mathbf{b}, \mathbf{a}_j)\}\,.$$

Here, the operator DIST denotes a simple distance metric such as sum of squared distance (SSD), SAD, MSE, etc. The best match (minimum distance) candidate is assigned as the predictor of the template $\mathbf{b}$.

Static template prediction

A static template is referred as the commonly used conventional template, i.e., mode 1 in Fig. 12. Let us suppose that the static template (mode 1) is used for prediction. For the first step, that is search for the best approximation of the known pixel values, the matrix A is modified by masking its rows corresponding to the spatial location of the pixels of the area B (the unknown pixel values). A compacted matrix A_c of size $3n^2 \times M$ is obtained. The known input image is compacted in the vector $\mathbf{b}_c$ of $3n^2$ values.

Let $\mathbf{a}_{c_j}$ denote the columns of the compacted dictionary A_c. The template matching algorithm proceeds by calculating $d_j = \mathrm{DIST}\left(\mathbf{b}_c, \mathbf{a}_{c_j}\right)$ for all $j = 1\ldots M$ in order to obtain

$$j_{opt} = \underset{j}{\arg\min}\,\{d_j\}$$

The extrapolated signal $\hat{\mathbf{b}}$ is simply assigned by the sample values of the candidate $\mathbf{a}_{j_{opt}}$ as $\hat{\mathbf{b}} = \mathbf{a}_{j_{opt}}$.

Optimized dynamic templates

The optimum dynamic template is selected among seven pre-defined modes as shown in Fig. 12. The optimization is conducted according to two different criteria: 1. minimization of the prediction residue MSE; 2. minimization of the RD cost function $J = D + \lambda R$, where D is the reconstructed block MSE (after adding the quantized residue when used in the coding scheme), and R is the residue coding cost estimated as $R = \gamma_0 M'$ [17] with M' being defined as the number of non-zero quantized DCT coefficients and $\gamma_0 = 6.5$.

3.2.3 Sparse approximation based spatial prediction

Given $A \in \mathbf{R}^{N \times M}$ and $\mathbf{b} \in \mathbf{R}^N$ with $N << M$ and A is of full rank, the problem of sparse approximation consists in seeking the solution of

$$\min\{\|\mathbf{x}\|_0 \ : \ A\mathbf{x} = \mathbf{b}\},$$

where $\|\mathbf{x}\|_0$ denotes the L_0 norm of $\mathbf{x}$, i.e., the number of non-zero components in $\mathbf{x}$. A is known as the dictionary, its columns $\mathbf{a}_j$ are the atoms, they are assumed to be normalized in Euclidean norm. There are many solutions $\mathbf{x}$ to $A\mathbf{x} = \mathbf{b}$ and the problem is to find the sparsest, i.e., the one for which $\mathbf{x}$ has the fewest non-zero components.

In practice, one actually seeks an approximate solution which satisfies:

$$\min\{\|\mathbf{x}\|_0 \ : \ \|A\mathbf{x} - \mathbf{b}\|_p \leq \rho\},$$

for some $\rho \geq 0$, characterizing an admissible reconstruction error. The norm p is usually 2. Except for the exhaustive combinatorial approach, there is no known method to find the exact solution under general conditions on the dictionary A. Searching for this sparsest representation is hence computationally intractable. MP algorithms have been introduced as heuristic methods which aim at finding approximate solutions to the above problem with tractable complexity.

In the project, the OMP algorithm which offers an iterative optimal solution to the above problem is considered. It generates a sequence of M dimensional vectors $\mathbf{x}_k$ having an increasing number of non-zero components in the following way. At the first iteration $\mathbf{x}_0 = 0$ and an initial residual vector $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = \mathbf{b}$ is computed. At iteration k, the algorithm identifies the atom $\mathbf{a}_{j_k}$ having the maximum correlation with the approximation error. Let A_k denote the matrix containing all the atoms selected in the previous iterations. One then projects $\mathbf{b}$ onto the subspace spanned by the columns of A_k, i.e., one solves

$$\min_{\mathbf{x}} \ \|A_k\mathbf{x} - b\|^2,$$

and the coefficient vector at the kth iteration is given as

$$\mathbf{x}_k = (A_k^T A_k)^{-1} A_k^T \mathbf{b} = A_k^+ \mathbf{b},$$

where A_k^+ is the pseudo-inverse of A_k. Notice that here $\mathbf{x}_k$ is a vector of coefficients. All the coefficients assigned to the selected atoms are recomputed at each step. However,

Dynamic template OMP
(26.48 dB/0.75 bpp)

Dynamic template TM
(25.69 dB/0.85 bpp)

Static template OMP
(24.23 dB/0.84 bpp)

Static template TM
(23.97 dB/0.88 bpp)

Figure 13: Prediction results of Foreman image.

in the experiments reported here, only one iteration has been considered for the sake of comparison with template matching.

The principle of the prediction based on sparse approximation is to first search for a best basis function (atom) which best approximates the known values in C, and keep the same basis function and weighting coefficient to approximate the unknown pixel values in B.

Let the quantity $\mathbf{x}$ denote a vector which will contain the result of the sparse approximation, i.e., the coefficient of the expansion of the vector $\mathbf{b}$ on only one atom. The sparse representation algorithm then proceeds by solving the approximate minimization

$$\mathbf{x}_{opt} = \min_{\mathbf{x}} \|\mathbf{b}_c - A_c\mathbf{x}\|_2^2 \text{ subject to } \|\mathbf{x}\|_0 = 1.$$

To recover the extrapolated signal $\hat{\mathbf{b}}$, the full matrix A is multiplied by $\mathbf{x}_{opt}$ as $\hat{\mathbf{b}} = A\mathbf{x}_{opt}$.

Here, the columns (atoms) $\mathbf{a}_{c_j}$ of the compacted dictionary A_c correspond to the normalized luminance values of texture patches extracted from the search window W.

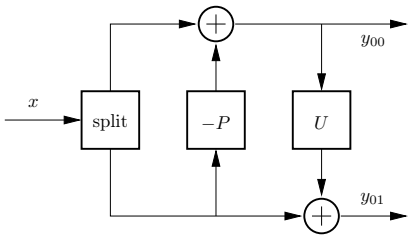


Figure 14: Lifting scheme with a single lifting stage

3.2.4 Linear combination of Template Matching predictors

In this variant of the prediction based on sparse approximation, in the search for the best approximation of the template, one searches for the K most correlated atoms and of their corresponding weight, each time with one iteration. One then computes an averable of the three resulting predictions.

3.3 Adaptive WT and perceptual impact in image coding

Wavelet transform (WT) is a very useful tool for image processing and compression. In particular, the lifting scheme (LS) implementation of WT was originally introduced by Sweldens [27] to design wavelets on complex geometrical surfaces, but at the same time it offers a simple way to build up both classic wavelet transforms and new ones.

The elements composing the lifting scheme are shown in Fig. 14. We call x the input signal, and y_{ij} the wavelet subbands. In particular, the first index determines the decomposition level ($i = 0$ being the first one), and the second index discriminates the channel ($j = 0$ for the low-pass or approximation signal, $j = 1$ for the high pass or detail signal). The input signal x is split into its even and odd samples, respectively called the approximation and the detail signal. Then, a prediction operator P is used in order to predict the odd samples of x from a linear combination of even samples. The prediction is removed from the odd samples in order to reduce their correlation with the even ones. Finally for the third block, the update operator U is chosen in such a way that the approximation signal y_{00} satisfies certain constraints, such as preserving the average of the input or reducing aliasing. It is interesting to notice that, with a proper combination of lifting steps (prediction and update) it is possible to enhance a given transform by imposing new properties on the resulting decomposition (for example, more vanishing moments).

LS are very flexible while preserving the perfect reconstruction property, and this allows to replace linear filters by nonlinear ones. For example, LS with adaptive update [23] or adaptive prediction [9,19] have been proposed in the literature, with the target of avoiding oversmoothing of object contours, and at the same time of exploiting the correlation of homogeneous regions by using long filters on them. The adaptivity makes it possible to use different filters over different parts of image. As a consequence, the resulting transform can be strongly non-isometric. This is a major problem for

compression, since all most successful techniques rely on the distortion estimation in the transform domain, either explicitly like in EBCOT [30], or implicitly, like in the zero-tree based algorithms (EZW, SPIHT [24, 25]. Therefore, in order to efficiently use the adaptive lifting scheme for image compression, we need to estimate correctly the distortion from the transform coefficients. Usevitch showed that the energy of an uncorrelated signal (such as the quantization noise is supposed to be) can be estimated for generic linear wavelet filter banks [34]. We extended this approach to adaptive update LS (AULS) [20], and to adaptive prediction LS (APLS) [21] (in particular those inspired by the paper by Claypoole *et al.* [9]), obtaining satisfying results in term of distortion estimation and of rate-distortion (RD) performance improvement.

When non-isometric linear analysis is used, Usevitch [34] showed that the MSE in the original domain D is related to the MSE's D_{ij} of the wavelet subbands y_{ij} by the linear relation $D = \sum_{ij} w_{ij} D_{ij}$. The weight w_{ij} is computed as norm of the reconstruction polyphase matrix columns for subband y_{ij}.

However APLSs (as well as AULSs) are nonlinear systems, therefore no polyphase representation of them exist. Our contribution in previous papers [20, 21] was to extended this approach to adaptive LS, and to show how to compute the weights w_{ij}. The error D is still obtained as a weighted sum of the subband errors, but now the weights depend on the input image, since the transform itself depends on it. In conclusion the proposed approach shows how to compute, in the transform domain, a metric which estimates the quantization noise MSE. This objective, non-perceptual distortion metric is then expressed as:

$$D_1 = \sum_{ij} w_{ij} d_{ij} \tag{2}$$

where d_{ij} is the MSE in the subband ij:

$$d_{ij} = \sum_{n,m} \left[y_{ij}(n,m) - \hat{y}_{ij}(n,m) \right]^2 . \tag{3}$$

3.3.1 Perceptual quality evaluation

Even though in our previous work the MSE estimation was quite reliable, we did not take into account the perceptual quality of the compressed image. Actually, we provided just a tool for estimating the MSE between two images in the wavelet domain, which was not possible before for non-linear (*i.e.* adaptive) wavelet transforms. Now, it is well known that MSE is not satisfactory for perceptual quality evaluation. In this work we propose a method for estimating the perceptual quality of an image compressed with an adaptive LS (with particular focus on APLS since they have by far the best performance). We make use of saliency maps in order to evaluate the different contributions of wavelet coefficients affecting different areas of the image; moreover we use the weights proposed in our previous works [21] in order to correctly compare different subbands.

We take inspiration from the quality metrics based on the saliency of specific areas in an image or a video. Let x be the original image, $\hat{x}$ the distorted (or compressed) one,

and n, m the spatial coordinates for pixels. The perceptual distortion is a weigthed sum of errors:

$$D_2 = \sum_{n,m} \mu(n,m) \left[x(n,m) - \hat{x}(n,m)\right]^p \tag{4}$$

where μ is a suitable saliency map. For example, in [8], it is proposed to take into account three phenomena: the image contrast (on a frame-by-frame basis), the global activity and the local activity (on a temporal basis). The image contrast mask, inspired on the work by Kutter and Winkler [15] uses a non-decimated WT of the image. Let W^{LL} be the LL band of undecimated wavelet transform of x. The contrast saliency map is defined as:

$$\alpha(n,m) = T[C_0(n,m)] \cdot W^{\mathrm{LL}}(n,m) \tag{5}$$

where

$$T[C_0] = \begin{cases} C_T & \text{if } C_0 < C_M \\ C_T \left(\frac{C_0}{C_M}\right)^{\epsilon} & \text{otherwise} \end{cases}$$

$$C_0(n,m) = \sqrt{2} \cdot \frac{\sqrt{|W^{\mathrm{HH}}(n,m)|^2 + |W^{\mathrm{HL}}(n,m)|^2 + |W^{\mathrm{LH}}(n,m)|^2}}{W^{\mathrm{LL}}(n,m)}$$

The parameters C_T, C_M, ϵ can be assigned according to the observations made in the paper [15]. This map does not take into account temporal effects in a video, and could be used if only fixed images are to be considered. However, one can make up for it by adding a further contribution accounting for global activity and based on motion vector norms.

In conclusion, we end up with a single masking function α which is higher where the observer is less sensible to errors, such that we can use $\mu = 1/\alpha$ in Eq. (4).

The problem is that this distortion metric should be computed in the spatial domain, which is a quite large impairment for compression algorithms, as already noted in the first section.

3.3.2 Proposed metric

Based on the previous work [20, 21] and inspired on the perceptual metrics used in [8, 15], we propose a new metric which would allow to evaluate the perceptual effect of quantization (and actually of any other degradation) performed in the transformed domain. In other words, we want to make it possible to evaluate the perceptual quality of a compressed image directly from its transformed coefficients, when *adaptive* and highly non-linear transforms are used.

The proposed metric is based on subband energy weighting (to make it possible to use adaptive filters) and on the perceptual saliency described in the previous section. The weighting allows to compare wavelet subbands having different orientations and resolutions; the spatial masking allows to evaluate the impact of each WT coefficient according to the spatial region it will affect in the reconstructed image. However, since

the different subbands have different resolutions, the mask α must be adapted to it. To this end, we define the mask value $\alpha_i(n,m)$ at resolution level i as the average of mask values in the positions associated to the coefficient (n,m):

$$\alpha_i(n,m) = \frac{1}{4^i} \sum_{k=2^i n}^{2^i n + 2^i - 1} \sum_{\ell = 2^i m}^{2^i m + 2^i - 1} \alpha(k,\ell) \tag{6}$$

Now we can define the distortion evaluation in the transform domain. The new metric is similar to the one in Eq. (2):

$$D_3 = \sum_{ij} w_{ij} d'_{ij} \tag{7}$$

since the weights (computed as defined in [20,21]) are necessary to compare the distortion in different subbands. The innovation stands in the term d'_{ij}, defined as follows:

$$d'_{ij} = \sum_{n,m} \mu_i(n,m) \left[y_{ij}(n,m) - \hat{y}_{ij}(n,m) \right]^2 \tag{8}$$

This equation is similar to the perceptual metric in Eq. (4); however here we use $\mu_i = 1/\alpha_i$. In its turn, α_i is defined in Eq. (6), and any saliency mask can be used in principle, even though in a first moment we propose the one suggested in [8].

3.4 Examplar-based inpainting based on local geometry

3.4.1 Introduction

Inpainting methods play an important role in a wide range of applications. Removing text and advertisements (such as logos), removing undesired objects, noise reduction [37] and image reconstruction from incomplete data are the key applications of inpainting methods. There are two algorithm categories: PDE (Partial Derivative Equation)-based schemes [32] and examplar-based schemes [10]. The former uses diffusion schemes in order to propagate structure in a given direction. The drawback is the introduction of blur due to diffusion. The latter relies on the sampling and the copying of texture from the known parts of the picture.
In this paper, we propose a novel inpainting algorithm combining the advantages of both aforementionned methods. As in [10], the proposed method involves two steps: first, a filling order is defined to favor the propagation of structure in the isophote direction. Second, a template matching is performed in order to find the best candidates to fill in the hole. Compared to previous approaches, the main contributions are fourfold: the first one concerns the use of structure tensors to define the filling order instead of field gradients. Second is to use a hierarchical approach to be less dependent on the singularities of local orientation. Third is related to constraining the template matching to search for best candidates in the isophote directions. Fourth is a K-nearest neighbor approach to compute the final candidate. The number K depends on the local context centered on the patch to be filled in.

This part is organized as follows. Section 2 describes the proposed method. Section 3 presents the performance of the method and a comparison with existing approaches. Some conclusions are dranw in this subsection.

3.4.2 Algorithm description

The goal of the proposed approach is to fill in the unknown areas of a picture I by using a patch-based method. As in [10], the inpainting is carried out in two steps: (i) determining the filling order; (ii) propagating the texture. We use almost the same notations as in [10]. They are briefly summarized below:

- the input picture noted I. Let $I : \Omega \rightarrow \mathcal{R}^n$ be a vector-valued data set and I_i represents its i-th component;
- the source region noted $\boldsymbol{\varphi}$, ($\boldsymbol{\varphi} = I - \Omega$);
- the region to be filled noted Ω;
- a square block noted ψ_p, centered at the point $\mathbf{p}$ located near the front line.

The differences between the approach in [10] and the proposed one are on the one hand the use of structure tensors and in the other hand the use of a hierarchical approach. In the following, we first describe the algorithm for a unique level of the hierarchical decomposition and then we describe how the hierarchical approach is used.

For one level of the hierarchical decomposition

As previously mentioned, the proposed approach follows the approach in [10] int the way that the inpainting is made in two steps. In a first step, a filling priority is computed for each patch to be filled. The second step consists in looking for the best candidate to fill in the unknown areas in decreasing order of priority. These two steps are described in the following.

Computing patch priorities

Given a patch ψ_p centered at the point $\mathbf{p}$ (unknown pixel) located near the front line, the filling order (also called priority) is defined as the product of two terms: $P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p})$.
The first term, called the confidence, is the same as in [10]. It is given by:

$$C(\mathbf{p}) = \frac{\sum_{q \in \psi_p \cap (I-\Omega)} C(\mathbf{q})}{|\psi_p|} \tag{9}$$

where $|\psi_p|$ is the area of ψ_p. This term is used to favor patches having the highest number of known pixels (At the first iteration, $C(\mathbf{p}) = 1$ $\forall \mathbf{p} \in \Omega$ and $C(\mathbf{p}) = 0$ $\forall \mathbf{p} \in I - \Omega$).
The second term, called the data term, is different from [10]. The definition of this term is inspired by PDE regularization methods acting on multivalued images [31]. The most efficient PDE-based schemes rely on the use of a structure tensor from which the

local geometry can be computed. As the input is a multivalued image, the structure tensor, also called Di Zenzo matrix [12], is given by:

$$\mathbf{J} = \sum_{i=1}^{n} \nabla I_i \nabla I_i^T \tag{10}$$

$\mathbf{J}$ is the sum of the scalar structure tensors $\nabla I_i \nabla I_i^T$ of each image channel I_i (R,G,B). The structure tensor gives information on orientation and magnitudes of structures of the image, as the gradient would do. However, as stated by Brox et al. [2], there are several advantages to use a structure tensor field rather than a gradient field. The tensor can be smoothed without cancellation effects : $\mathbf{J}_\sigma = \mathbf{J} * G_\sigma$ where $G_\sigma = \frac{1}{2\pi\sigma^2} exp(-\frac{x^2+y^2}{2\sigma^2})$, with standard deviation σ. In this paper, the standard deviation of the Gaussian distribution is equal to 1.0.
The Gaussian convolution of the structure tensor provides more coherent local vector geometry. This smoothing improves the robustness to noise and local orientation singularities. Another benefit of using a structure tensor is that a structure coherence indicator can be deduced from its eigenvalues. Based on the discrepancy of the eigenvalues, this kind of measure indicates the degree of anisotropy of a local region. The local vector geometry is computed from the structure tensor $\mathbf{J}_\sigma$. Its eigenvectors $\mathbf{v}_{1,2}$ ($\mathbf{v}_i \in R^n$) define an oriented orthogonal basis and its eigenvalues $\lambda_{1,2}$ define the amount of structure variation. $\mathbf{v}_1$ is the orientation with the highest fluctuations (orthogonal to the image contours), and $\mathbf{v}_2$ gives the preferred local orientation. This eigenvector (having the smallest eigenvalue) indicates the isophote orientation. A data term D is then defined as [36]:

$$D(\mathbf{p}) \quad = \quad \alpha + (1-\alpha) exp(-\frac{C}{(\lambda_1 - \lambda_2)^2}) \tag{11}$$

where C is a positive value and $\alpha \in [0,1]$ ($C = 8$ and $\alpha = 0.01$). On flat regions ($\lambda_1 \approx \lambda_2$), any direction is favored for the propagation (isotropic filling order). The data term is important in presence of edges ($\lambda_1 >> \lambda_2$).
Figure 15 shows the isophote directions (a) and the value of the coherence norm $(\frac{\lambda_1-\lambda_2}{\lambda_1+\lambda_2})^2$ (b). Black areas correspond to areas for which there is no dominant direction.

Propagating texture and structure information

Once the priority P has been computed for all unknown pixels $\mathbf{p}$ located near the front line, pixels are processed in decreasing order of priority. This filling order is called percentile priority-based concentric filling (PPCF). PPCF order is different from Criminisi's approach. Criminisi et al. [10] updated the priority term after filling a patch and systematically used the pixel having the highest priority. The advantage is to propagate the structure throughout the hole to fill. However, this advantage is in a number of cases a weakness. Indeed, the risk, especially when the hole to fill is rather big, is to propagate too much the image structures. The PPCF approach allows us to start filling by the $L\%$ pixels having the highest priority. The propagation of image structures in the isophote direction is still preserved but to a lesser extent than

(a) (b) (c) (d)

Figure 15: (a) direction of the isophotes;(b) coherence norm: black areas correspond to areas for which there is no dominant direction; (c) Filling with the best candidate (K=1); (d) Filling with the best 10 candidates.

in [10]. Once the pixel having the highest priority is found, a template matching based on the sum of squared differences (SSD) is applied to find a plausible candidate. SSD is computed between this candidate (entirely contained in $\boldsymbol{\varphi}$) and the already filled or known pixels of ψ_p. Finally, the best candidate is chosen by the following formula:

$$\psi_{\widehat{q}} = \arg \min_{\psi_q \in \mathcal{W}} d(\psi_{\widehat{p}}, \psi_q) \tag{12}$$

where $d(.,.)$ is the SSD. Note that a search window $\mathcal{W}$ centered on p is used to perform the matching.
Finding the best candidate is fundamental for different reasons. The filling process must ensure that there is a good matching between the known parts of ψ_p and a similar patch in $\boldsymbol{\varphi}$ in order to fill the unknown parts of ψ_p. The metric used to evaluate the similarity between patches is then important to propagate the texture and the structure in a coherent manner. Moreover, as the algorithm is iterative, the chosen candidate will influence significantly the result that will be obtained at the next iteration. An error leading to the apparition of a new structure can be propagated throughout the image. In order to improve the search for the best candidate, the previous strategy is modified as follows:

$$\psi_{\widehat{q}} = \arg \min_{\psi_q \in \boldsymbol{\varphi}} d(\psi_{\widehat{p}}, \psi_q) + (\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2})^2 \times f(p,q) \tag{13}$$

where the first term $d(.,.)$ is still the SSD and the second term is used to favor candidates in the isophote direction, if any. Indeed, the term $(\frac{\lambda_1-\lambda_2}{\lambda_1+\lambda_2})^2$ is a measure of the anisotropy at a given position (as explained in section 3.4.2). On flat areas, this term tends to 0. The function $f(p,q)$ is given by:

$$f(p,q) = \frac{1}{\epsilon + \frac{|\mathbf{v}_2 \cdot \mathbf{v}_{pq}|}{\|\mathbf{v}_{pq}\|}} \tag{14}$$

where $\mathbf{v}_{pq}$ is the vector between the centre p of patch ψ_p and the centre q of a candidate patch ψ_q. ϵ is a small constant value, set to 0.001. If the vector $\mathbf{v}_{pq}$ is not collinear to the isophote direction (assessed by computing the scalar product $\mathbf{v}_2 \cdot \mathbf{v}_{pq}$),

this candidate is penalized. In the worst case (the two vectors are orthogonal), the penalization is equal to $1/\epsilon$. When the two directions are collinear, the function $f(p,q)$ tends to one.
A K nearest neighbour search algorithm is also used to compute the final candidate to improve the robustness. We follow Wexler et al.'s proposition [38] by taking into account that all candidate patches are not equally reliable (see equation 3 of [38]). An inpainting pixel $\widehat{c}$ is given by (c_i are the pixels of the selected candidates):

$$\widehat{c} = \frac{\sum_i s_i c_i}{\sum_i s_i} \tag{15}$$

where s_i is the similarity measure deduced from the distance (see equation 2 of [38]). Most of the time, the number of candidates K is fixed. This solution is not well adapted. Indeed, on stochastic or fine textured regions, as soon as K is greater than one, the linear combination systematically induces blur. One solution to deal with that is to locally adapt the value K. In this approach we compute the variance σ_W^2 on the window search. K is given by the function $a + \frac{b}{1+\sigma_W^2/T}$ (in our implementation we use $a = 1$, $b = 9$ and $T = 100$. It means that we can use up to 10 candidates to fill in the holes). Figure 15 (c) and (b) shows the rendering of a fine texture with the best and the best ten candidates. For this example, good rendering quality is achieved by taking into account only the best candidate.

Hierarchical decomposition

Previous sections described the proposed approach for a given pyramid level. One limitation concerns the computation of the gradient ∇I used to define the structure tensor. Indeed, as the pixels belonging to the hole to fill are initialized to a given value (0 for instance), it is required to compute the gradient only on the known part of the patch ψ_p. This constraint can undermine the final quality. To overcome this limitation, a hierarchical decomposition is used in order to propagate throughout the pyramid levels an approximation of the structure tensor. A Gaussian pyramid is then built with successive low-pass filtering and downsampling by 2 in each dimension leading to nL levels. At the coarsest level $\mathcal{L}_0$, the algorithm described in the previous section is applied. For a next pyramid level $\mathcal{L}_n$, a linear combination between the structure tensors of level $\mathcal{L}_n$ and $\mathcal{L}_{n-1}$(after upsampling) is performed:

$$\mathbf{J}_h^{\mathcal{L}_n} = \nu \times \mathbf{J}^{\mathcal{L}_n} + (1-\nu) \times \uparrow 2(\mathbf{J}^{\mathcal{L}_{n-1}}) \tag{16}$$

where $\mathbf{J}_h$ is a structure tensor computed from a hierarchical approach. $\uparrow 2$ is the upsampling operator. In our implementation, ν is fixed and set to 0.6. This hierarchical approach makes the inpainting algorithm more robust. At the coarsest level, the local structure tensor is a good approximation of the local dominant direction. Propagating such information throughout the pyramid decreases the sensitivity to local orientation singularities and noise. By default, nL is set to 3.

3.5 Visual attention modeling: relationship between visual salience and visual importance

We have already explained in the Chapter 4 of the D3.1 deliverable, how saliency maps can be used to drive decisions when coding images. This new study [35] aims at comparing two types of maps, obtained through psychophysical experiments.

Exactly two mechanisms of visual attention are at work when humans look at an image: bottom-up and top-down. Bottom-up refers to a mechanism driven by only low level features, such as color, luminance, contrast. Top-down refers to a mechanism which is more about to the meaning of the scene. According to these two mechanisms, the research about (bottom-up) visual saliency and (top-down) visual importance, which is also called ROI, can provide important insights into how biological visual systems address the image-analysis problem. However, despite the difference in the way visual salience and visual importance are determined in terms of visual processing, both salience and importance have traditionally been considered synonymous in the signal-processing community: They are both believed to denote the most visually "relevant" parts of the scene. In the study, we present the result of two psychophysical experiments and an associated computational analysis designed to quantify the relationship between visual salience and visual importance. In the first experiment, importance maps were collected by asking human subjects to rate the relative visual importance of each object within a database of hand-segmented images. In the second experiment, experimental saliency maps were computed from visual gaze pattern measured for these same images by using an eye-tracker and task-free viewing. The results of the experiment revealed that the relationship between visual salience and visual importance is strongest in the first two second of the 15-second observation interval, which implies that top-down mechanisms dominate eye movements during this period.From the psychophysical point of view, these results suggest a possible strategy for human visual coding. If the human visual system can so rapidly identify the main subject(s) in a scene, such information can be used to prime lower-level neurons to better encode the visual-level tasks such as scene categorization. Several researchers have suggested that rapid visual priming might be achieved via feedback and/or lateral interactions between groups of neurons after the "gist" of the scene is determined. The results of the study provide psychophysical evidence that lends support to a gist-based strategy and a possible role for the feedback connections that are so prevalent in mammalian visual systemsIn the study, we have attempted to quantify the similarities and differences between bottom-up visual salience and top-down visual importance. The implications of these initial findings for image processing are quite important. As we know, several algorithms have been published which can successfully predict gaze patterns [16]. Our results suggest that these predicted patterns can be used to predict importance maps when coupled with a segmentation scheme. In turn, the importance maps can then be used to perform importance-based processing such as compression, auto-cropping, enhancement, unequal error protection, and quality assessment.

3.6 Taking into account the geometric distortion

An efficient estimation and encoding of motion/disparity information is a crucial step in 2D and 3D video codecs. In particular, the displacement information can be at the origin very rich, and therefore very costly to represent. For this reason, displacement information is always in some way degraded in order to be efficiently sent. In order to justify these affirmation one can think to the "original" or "actual" motion vector field[2] as to a dense (i.e. one vector per pixel), infinite-precision (i.e. not limited to integer, half or quarter pixel) field. When we encode this field we degradate it:

- Using only one vector per block of pixel,
- Reducing the precision to integer, half or quarter pixel precision

Usually, this information degradation is performed in such a way to optimize a rate-distortion cost function. This is for example when the best macroblock partition is selected in H.264 with a RD-optimization approach [26,39]. In this traditional context, the mode selection procedure can be interpreted as an implicit segmentation of the motion vector field in which the shape of the objects is constrained by the partitions of the encoder (e.g. the 7 partition modes of H.264, [42]).

This operation can be seen in the following way. We are corrupting an "optimal" MVF in order to reduce the coding cost. The representation of this "corrupted" MVF is driven by a rate-distortion cost function: the mode which minimize the functional $J = D + \lambda R$ is choosen, where R is the coding rate and D is usually a quanity related to the MSE.

We notice that in this operation there is no consideration for the perceptual impact of corrupting motion vectors. The basic idea proposed here stems from the consideration that we *should* take into account what is the effect of changing motion vectors on the perceived quality of the reconstructed image.

	lena	peppers
PSNR	27.9	27.9
GM	4.11	4.28

Table 2: Geometrical distortion measure

At this end, we borrow the framework of geometric distortion estimation recently proposed by D'Angelo et al. [11]. In that paper, the authors introduced an objective perceptual and full-reference metric which allows to estimate the perceived quality of an image affected by geometric distortion. The geometric distortion is represented as a MVF $\mathbf{v}$ applied to the original image X to give the distorted image Y

$$Y(\mathbf{p}) = X(\mathbf{p} + \mathbf{v})$$

[2]For simplicity, we refer in the following to motion vectors, but all these considerations hold for disparity vectors as well

Figure 16: "Lena" and "peppers" after the same geometrical distortion. The left image is visibly corrupted (look at the vertical structure on the left), while the right one looks perfect. The PSNR is not able to catch the perceptual quality difference of these images, while the GM is, as shown in Table 2

A couple of examples is shown in Fig. 16. The motion vector field is shown in Fig. 17. It amounts to only horizontal displacements, with a displacement amplitude varying along the vertical direction. We can argue that, even though the amplitude of the displacement field is exactly the same for the two images, the perceived quality is different, since "lena" has much more vertical contours and structures that are disrupted by the horizontal shift of pixels. PSNR is unfit to catch the effect of geometrical distortion. The metric proposed by D'Angelo et al. [11] seems better adapted to this end, as shown in Tab 2.

3.6.1 The original distortion measure

This metric (for short, GM, as for geometry-based metric) is based on computing the geometric features of the image directed along a given direction θ. Then, for each motion vector, authods consider its component along direction θ, be it $\mathbf{v}_\theta$, and finally the gradient of $\mathbf{v}_\theta$ in the direction orthogonal to θ. This gradient gives the amount of degradation with respect to the structures alligned with θ; a Gabor filtering of the image allows to find the amount of energy of the structures having the same orientation θ for the considered pixel. Then, the degradation estimation takes into account both the displacement (gradient of $\mathbf{v}_\theta$ in the direction orthogonal to θ) and the importance of the structures (Gabor filter). A product of suitable real powers of these contribution assesses the geometrical estimation for a pixel and for a given direction. The global geometrical distortion measure is given by averaging over all directions and all pixels.

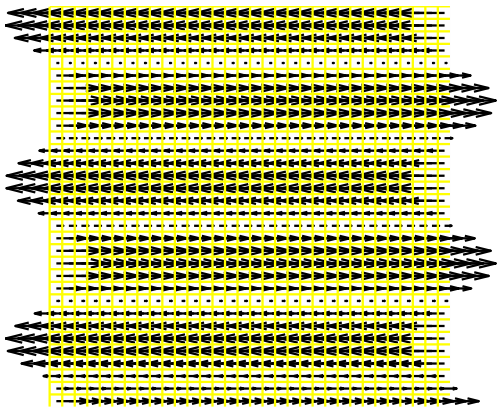


Figure 17: The motion vector field used for generating the geometrical distortion

Tests show that considering two directions, horizontal and vertical, is already sufficient for collecting the most of the relevant information.

3.6.2 Proposed methodology

The proposed methodology is based on the following assumption: there is some optimal dense motion vector field, and we are able to estimate it[3]. Let $\mathbf{v}^*$ be this MVF. Every time we do not use this MVf it is like we are imposing a geometrical distortion through an error field $\delta\mathbf{v}$:

$$\delta\mathbf{v} = \mathbf{v}^* - \mathbf{v},$$

where $\mathbf{v}$ is the MVF we are actually using. Therefore we can change the cost function of $\mathbf{v}$, from

$$J(\mathbf{v}) = D(\mathbf{v}) + \lambda R(\mathbf{v})$$

to

$$J(\mathbf{v}) = GM(\delta\mathbf{v}) + \lambda R(\mathbf{v})$$

. Finally we can try to use this new cost function whenever the old one was used:

- Motion estimation
- Motion vector encoding
- Mode selection
- Disparity estimation
- Disparity/depth encoding

[3]Some methods for dense MVF estimation are presented in this document and in D5.2

The last two issues are related to the 3D video coding, which can immediately benefit from this technique. In particular, when one is encoding disparity maps or depth maps, the effect of this compression is a modification of the depth, and therefore an additive error field δ. As for the 2D case, the effect of this error field is a geometric distortion, but this time this distortion affects the synthetized view.

We observe that the proposed metric lends itself well to the considered framework. In particular, it suggests the use of a closed-loop, two-passes encoder.

- This metric is a full reference one, which perfectly matches the closed-loop video encoding paradigm.
- It demands a pre-analysis of images, then providing a map of relevant structures (according to some orientation). This also fit well with a two-passes video encoder, which in a first pass produce the information of all relevant oriented structures, and then, using these maps, can estimate the perceptual impact of motion/disparity information encoding.

In conclusion, the methodology proposed in this report is promising since it goes in the direction of assessing the perceptual effect of information degradation due to motion information compression. A perceptual encoder must take this phenomenon into account in order to perform an optimal rate-quality trade-off.

4 Performance evaluation: first results

4.1 Dense motion vector estimation with Cafforio-Rocca Algorithm

We considered several video sequences with different motion content, (from low motion content to irregular, high motion sequences) and we performed two kind of experiments on them. In a first set of experiments, described in section 4.1.1, we only considered the effectiveness of CR motion vector improvement in the framework of the H.264 codec. In the second set of experiments, described in section 4.1.4, we considered the introduction of the CRA within the H.264 codec, and we evaluated the RD performances with respect to the variations of all relevant parameters.

4.1.1 Tests on Motion Estimation

In this section we consider tests made with the goal of validating the CR ME. We will not consider the impact over the global RD performances of the encoder.

4.1.2 Comparison with a Block Matching Algorithm

In a first set of experiments we compared the CR MVs with a MVF obtained by classical block-matching algorithms. We considered several input video sequences, and for each of them several pairs of current-reference frames (not necessarily consecutive images). For each pair of current-reference images, we computed the prediction error energy with respect to the full-search BMA MVF, and the prediction error energy in the case of CR ME. In this case we use the original (*i.e.* not quantized images), and the computation has been repeated for several values of the Lagrangian parameter λ.

The results are shown in Fig. 18. For all the test sequences we find a similar behavior: but for very small values of λ, the CR vectors guarantee a better prediction with respect of the BMA (green dashed line). For increasing values of λ the MSE decreases quickly, reaches a minimum and then increases very slowly towards a limit value, corresponding to $\lambda = \infty$. The latter case corresponds to a null refinement: all the improvement is due to the validation test, and corresponds with difference between the green and blue dashed lines in Fig. 18. The minimum value of the MSE is obtained when besides the validation test, the vectors are modified by the refinement step. We also remark that the CRA is quite robust wrt the choice of λ; the difference between the minimum and the asymptotical value of the black curve is the contribution of the refinement step.

4.1.3 Motion estimation with quantized images

The first experiment shows us that the CRA has the potential to improve the BMA motion vectors. However comparing only the prediction MSE is not fair since the cost of encoding the vectors is not taken into account. Of course, in this case, it would be extremely costly to encode the CR vectors, since there is a vector per pixel. The RD comparison would be definitely favorable to the classical ME algorithm. This is

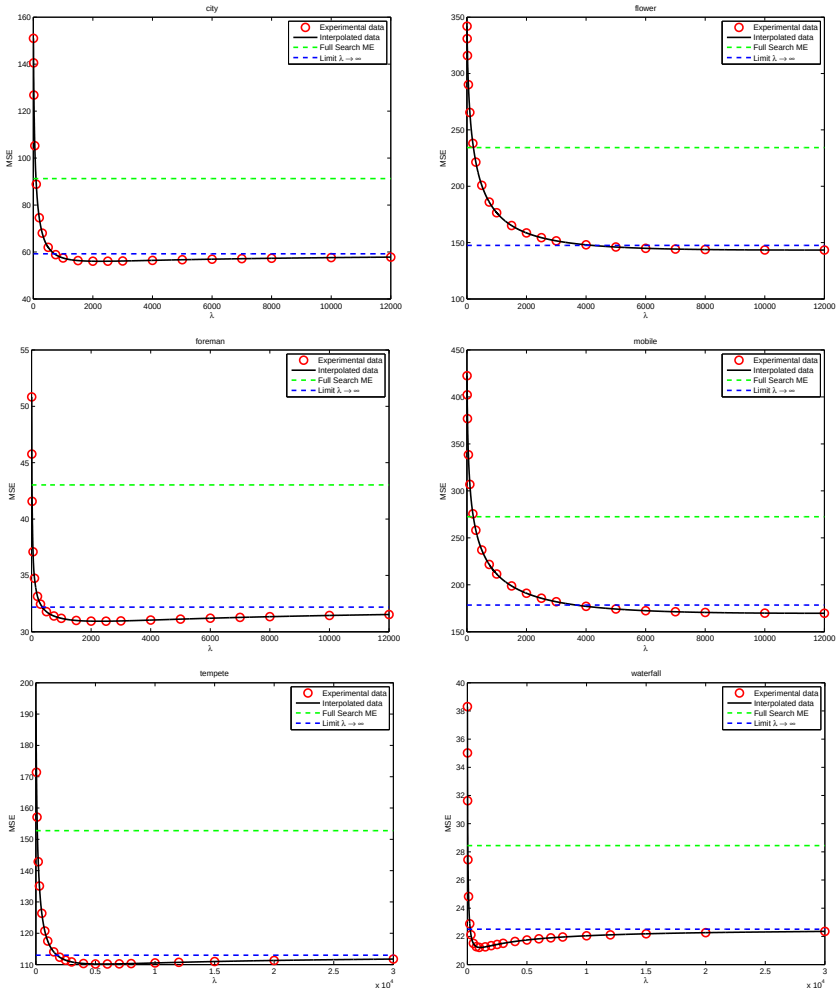


Figure 18: Motion Estimation performances of the CRA algorithm, different sequences

Method	MSE
H.264 Vectors	54.64
CR MSE $\lambda = 10$	95.48
CR MSE $\lambda = 100$	60.23
CR MSE $\lambda = 1000$	53.22
CR MSE $\lambda = 10000$	53.06
CR MSE $\lambda = 100000$	53.17
CR MSE $\lambda = \infty$	53.17
H.264 Coding MSE	6.08

Table 3: Prediction error energy, "foreman" sequence.

Method	MSE
H.264 vectors	90.79
CR $\lambda = 10$	571.02
CR $\lambda = 100$	237.34
CR $\lambda = 1000$	92.09
CR $\lambda = 10000$	72.86
CR $\lambda = 100000$	71.68
CR $\lambda = \infty$	71.69
H.264 MSE	3.49

Table 4: Prediction error energy, "mobile" sequence.

however why the CRA is not used in hybrid video coders at least in its classical form. On the contrary, the in the modified CRA the cost of MV coding is zero, but this is obtained by sacrificing the accuracy of validation and refinement, which must be performed using quantized data.

We designed the a second kind of tests in order to evaluate the potential gains of the CRA in the framework of a H.264-like coder. Unlike the previous case, we did not used the original images to perform the CRA, but we used those available to a H.264 decoder. More precisely, we first used H.264 to produce: the MVF between two images in a video sequence (indicated by $\mathbf{v}$, and the decoded current and reference image, indicated as $\hat{I}_k$ and $\hat{I}_h$ (*e.g.* $h = k - 1$).

Then the following quantities were computed:

H.264 Vectors MSE : The prediction error mean squared value for the H.264 vectors.

$\mathbf{v}_{\mathrm{CR}}$: The CR motion vectors, obtained by using $\hat{I}_k$ and $\hat{I}_h$ and $\mathbf{v}$.

CR MSE : The prediction error mean squared value for the CR vectors.

Some results are summarized in table 3 and 4. We performed the same test over many other sequences, and we obtained similar results.

We observe that in this case as well, the CR method is able to potentially improve the performance: the prediction produced with the CR vectors has (but for small λ

	Δ PSNR	Rate reduction
bus	0.01	-0.02%
city	0.01	-0.11%
coastguard	0.01	-0.07%
flower	0.04	-0.76%
football	0.02	-0.35%
mobile	0.02	-0.42%
paris	0.02	-0.55%
tempete	0.01	-0.13%

Table 5: Rate distortion performances improvement when introducing the new coding mode into H.264.

values) a smaller error than the original H.264 vectors, even if the CRA is run over quantized images. Moreover in this case the comparison is fair in the sense that the CR vectors do not cost any bit more than the original ones.

The last line of the table reports the MSE of the decoded macroblock in H.264. Of course it is much smaller of the prediction error energy, since it benefits from the encoded residual information. As a conclusion, it is critical to keep an efficient residual coding, since it is responsible for a large amount of the distortion reduction.

4.1.4 RD performances

In this section we comment about the performance of the modified H.264 coder. In order to assess the effectiveness of the proposed method, we have implemented it within the JM H.264 codec. We considered several design choice: the effects of the Lagrangian parameter, of the threshold and of the initialization method. Finally we compare the RD comparison to the original H.264 coder.

The proposed method seems not to be too affected by the value of the threshold γ provided that it is not too small (usually $\gamma > 10$ works well). For the sake of brevity we do not report here all the experimental results. Likewise, we found that setting $\lambda = 10^4$ works fairly well in all the test sequences. In the following, these values of the parameters are kept.

A larger impact on the performance is due to the validation step. Introducing a third candidate vector for validation allows a fast motion vector recover when passing from one object to another within the same block.

Finally we compared an H.264 coder where the new coding mode was implemented with the original one. The results are shown in table 5, where we report, for each test sequence, the improvement in PSNR and the bit-rate reduction computed with the Bjöntegard metric [1] over four QP values. We observe that the improvement are quite small, in part because the CR mode is rarely selected rarely (usually only for 10% of the blocks).

It is worth noting that the coding time with the modified coder are very close to those of the original one: we usually observed an increase less than 2%.

4.2 Experimental Results of the proposed Intra mode

4.2.1 Sparse prediction versus template matching

The proposed sparse spatial prediction method has been assessed competitively to the template matching based prediction using both a static and optimized dynamic templates. It has then been assessed in a still image coding scheme in which the residue blocks are encoded with an algorithm similar to JPEG.

In order to initialize the prediction, the top 3 rows and left 3 columns of blocks of size 8×8 are intra coded with JPEG algorithm. When a block has been predicted, the residue is quantized and encoded similar to JPEG. In this coding structure, a uniform quantization matrix of step size equal to 16 is weighted by a quality factor (QF). The QF is increased from 10 to 90 with a step size of 10. The reconstructed image is obtained by adding the quantized residue to the prediction. The best approximation support (if dynamic template selection is used) is Huffman encoded when transmitted to the decoder. For the sake of the comparison with template matching, the OMP algorithm is iterated only once.

Fig. 13 shows the predicted images of Foreman[1] at QF=30 where the dynamic template selection criterion is the minimization of the prediction residue MSE. Fig. 19 demonstrates the corresponding prediction performance curves for Foreman (CIF) and Barbara (512×512) images. The quality of the predicted signal, both visually and in terms of PSNR/bit-rate, is significantly improved when compared with the conventional template matching based prediction.

Fig. 20 shows the coding performance curves of the proposed sparse image prediction algorithm in comparison to the template matching based prediction using both a static and RD optimized dynamic templates. One can observe that the proposed method with dynamic template selection significantly improves the coding performance with respect to the conventional template matching prediction. A gain of up to 3 dB has been achieved in both prediction and coding.

4.2.2 Prediction based on a linear combination of template matching predictors

The intra prediction method, based on a linear combination of template matching predictors, has been integrated so that it competes with all other intra prediction modes existing in the TMuC 0.9. However, the algorithm briefly described in the previous section had to be adapted in order to take into account:

- The block size and the template size,
- The number and the size of the search windows and the number of atoms,
- A RD Cost threshold,
- A new signalling.

[1] The first frame in the Foreman (CIF) sequence is used.

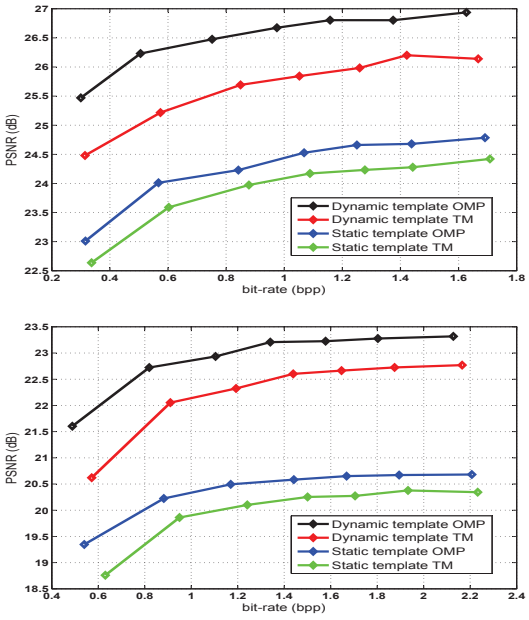


Figure 19: Prediction performance curves for (up) Foreman and (down) Barbara images.

These updates and related characteristics are detailed in the following.

Block size and template size

This new intra prediction is implemented for the following block size: 4x4, 8x8, 16x16 and 32x32. However, its activation depends on the chosen profile. When the High Efficiency profile is selected, the new intra prediction is used for all block sizes except for 4x4 blocks for the class A and for the 32x32 blocks for the class C and D. When the Low Complexity profile is selected, it is used only for 4x4 and 8x8 blocks whatever the video class. For each block size, the template area surrounding the block to be predicted has a L shape and is four pixels large whatever the size of the block.

Search windows and number of atoms

The concept of prediction unit (PU) introduces two possible sizes during the prediction process: NxN and 2Nx2N. Depending on whether the cases NxN or 2Nx2N are processed, the number of search windows is respectively 3 or 2. The number of atoms added in the dictionary is determined according to the number of windows and their width and height.

The syntax of the TMuC bit stream has been modified in order to specify when this new intra prediction mode is used. Four cases are taken into account: 4x4, 8x8, 16x16

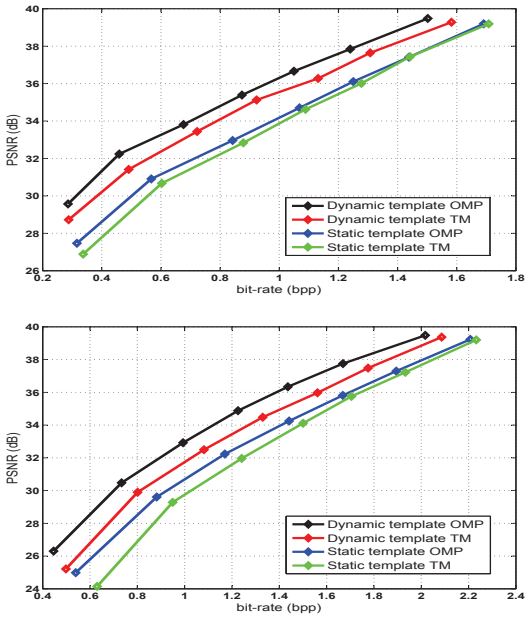


Figure 20: Coding performance curves for (up) Foreman and (down) Barbara images.

and 32x32 intra prediction modes. The mode "4", (IntraVertical - 4 [3]), has been extended with one flag the "weighted template matching" flag (wtm_flag). If this flag is set, that means that the new intra prediction mode is used; otherwise the classic mode "4" prediction is used.

We noticed that:

- the new prediction method always improves the prediction of TMuC 0.9,
- the average BD rate gain is -0.8
- although the new intra prediction is not used for 16x16 and 32x32 blocks with the Low Complexity profile, it performs better than when it is used with the High Efficiency profile. However, it is too time demanding and only the High Efficiency profile should be taken into account.

4.3 Experimental results for adaptive wavelet compression

The proposed metric can be used to compare the perceptual quality of images, so one can easily conceive a battery of tests devoted to inspect the correlation between the proposed metric and a subjective measure.

However we introduced the metric in Eq. (7) in order to improve resource allocation for image and video coding. Therefore, a more significant set of experiments would

Image	Δ_1^{SSIM}	Δ_2^{SSIM}
barbara	4.750	0.692
baboon	6.748	1.835
bottle	2.277	0.043
cameraman	5.459	0.095
couple	4.309	0.141
crowd	3.264	0.048
einst	5.443	0.111
house	2.697	0.009
lena	3.109	0.194
man	4.613	0.235
plane	2.835	0.174
spring	3.259	0.298
truck	2.398	0.233
woman1	3.858	0.184

Table 6: Average SSIM gains, percent values.

consist in using Eq. (7) to drive any resource allocation algorithm, be it a simple uniform quantization of WT coefficients (the resource allocation would decide the quantization step for each subband) or more efficient techniques such as EBCOT.

A first set of experiments is conducted as follows: for a given image, the saliency map μ is computed as specified in the previous Section. Then the image is transformed using the adaptive wavelet transform proposed by Claypoole. The we considered three methods to allocate coding resources to coefficients coding blocks: a traditional method based on coefficient variances; a weighted method using MSE-minimizing weights proposed in [21], and a perceptual method using the weigths and the average value of the saliency map in the locations corresponding to the code block.

Then, the image was coded by simple uniform quantization and entropic coding, using the rates which in turn take into account:

1. only the variances;
2. variances and the normalizing weights;
3. the variances, the normalizing weights and the saliency map.

For each technique, all the images were coded at several coding rates, ranging from 0.1 to 2 bpp. Then we evaluated the quality of the reconstructed image using PSNR and SSIM. Finally, we computed the difference in PSNR and SSIM (indicated with $\Delta^{\mathrm{PSNR}}$and Δ^{SSIM}) between techniques 1) and 2) and between 2) and 3). The first set of differences measures the impact of correct subband weighting from an objective (Δ_1^{PSNR}) and subjective (Δ_1^{SSIM}) with respect to classical coding. The second set of measures indicates the objective (Δ_2^{PSNR}) and subjective (Δ_2^{SSIM}) impact of saliency maps.

Analytical results are shown in Tables 6 and 7. In the first one, we show the quantities $\Delta_1^{\mathrm{SSIM}}$and Δ_2^{SSIM}. Looking at the Δ_1^{SSIM} column, we see that the correct weighting of the transform subband has a beneficial impact over subjective quality, allowing to improve the SSIM up to 6.7%. A further improvement, which is smaller

Image	Δ_1^{PSNR}	Δ_2^{PSNR}
barbara	4.265	0.222
baboon	3.474	0.126
bottle	4.567	-0.182
cameraman	3.810	-0.186
couple	3.237	-0.105
crowd	2.356	0.033
einst	4.180	0.001
house	4.179	0.068
lena	3.583	-0.103
man	3.115	0.028
plane	2.995	-0.037
spring	3.187	0.080
truck	2.649	0.058
woman1	3.168	0.071

Table 7: Average PSNR gains, in dB.

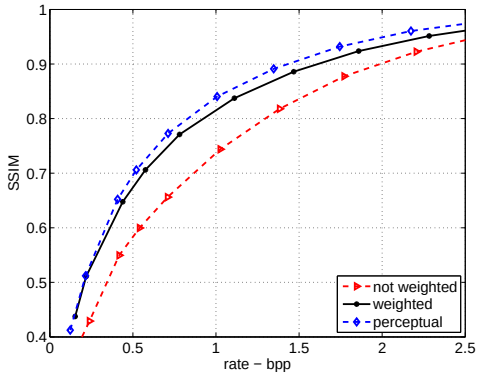


Figure 21: SSIM for the image "baboon" at several coding rates

but not negligible, is possible when the saliency information is used, since the values of the Δ_2^{SSIM} column are always positive. The performance gain of our contribution can globally be assessed as the sum of the two deltas.

The second tables shows us that correct weights do always improve the image PSNR. This is exactly what we expected, since this weighting was conceived to improve the objective quality of the image. We also notice that taking into account saliency does not always improve the PSNR[4]. This result is not surprising, since it is known that PSNR is not perfectly correlated to subjective quality.

We conclude that the proposed metric, used in the transformed domain, allows to improve the SSIM of decoded images, simply by altering the coding resource allocation

[4]If the quantization noise was a perfectly uncorrelated random process, the expected value of PSNR obtained with the weights would be larger than any other. However the facts that the quantization noise is not white and that we can only compute the average PSNR and not its expected value, makes it possible for some positive Δ_2^{PSNR}'s to appear.

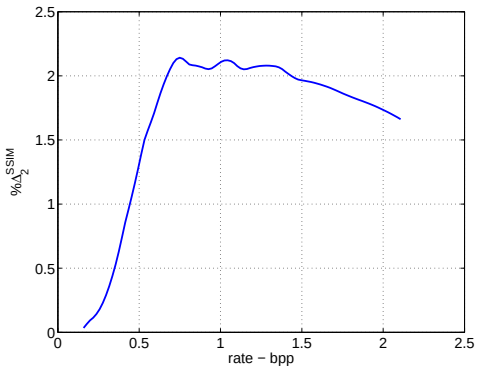


Figure 22: SSIM differences Δ_2^{SSIM} for the image "baboon"

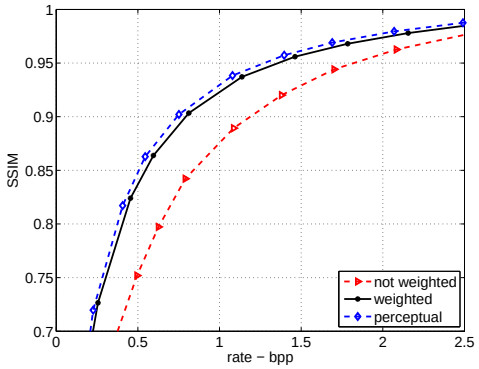


Figure 23: SSIM for the image "barbara" at several coding rates

between coding blocks. This is obtained in spite of the occasional reduction of PSNR.

We also report some more detailed results for a couple of images. In Fig. 21 we show the SSIM as a function of the coding rate for the three considered techniques and for the test image "baboon". We see that the improvement with respect to the basic technique (red curve) is consistent for all the coding rates. Moreover, in Fig. 22 we show the Δ_2^{SSIM} for this image (using interpolated values for computing the SSIM difference). We observe that SSIM improvements are relevant above all at the medium coding rates. It is worth nothing that for coding rates below 0.5 bpp the quality of the decoded image is not satisfactory, whatever the coding technique is. In Fig. 23 we report the SSIM behavior for another test image, "barbara". Similar conclusions (with respect to the previous case) can be drawn.

Finally, in Fig. 24, 25, and 26 we show the decoded "baboon" images for the three techniques. The coding rates are approximately the same (0.7 bits per pixel), but the visual quality are far different. In the first image, neither the weights nor the saliency

Figure 24: Decoded image, no weighting, Rate 0.71bpp PSNR 22.53 dB SSIM 0.656

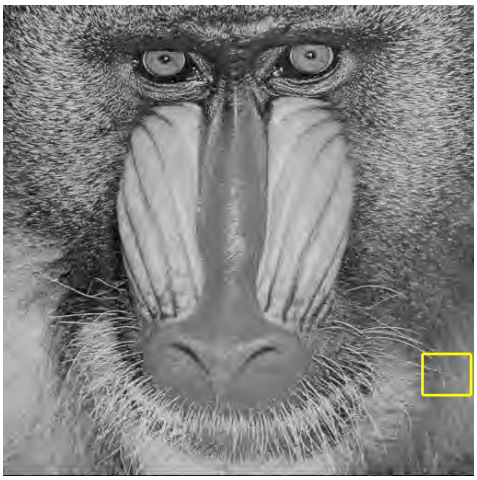


Figure 25: Decoded image, weights, Rate 0.72bpp PSNR 25.88 dB SSIM 0.750

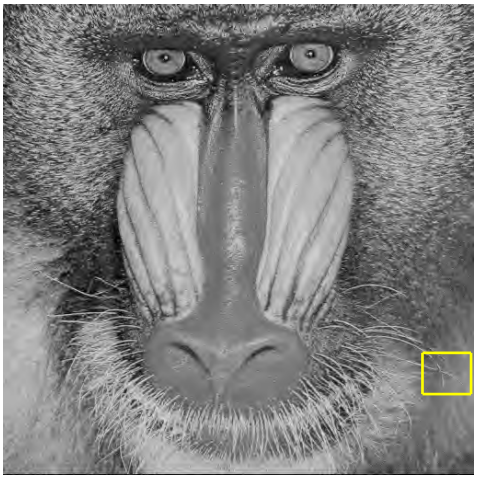


Figure 26: Decoded image, perceptual coding, Rate 0.71bpp PSNR 25.85 SSIM 0.773

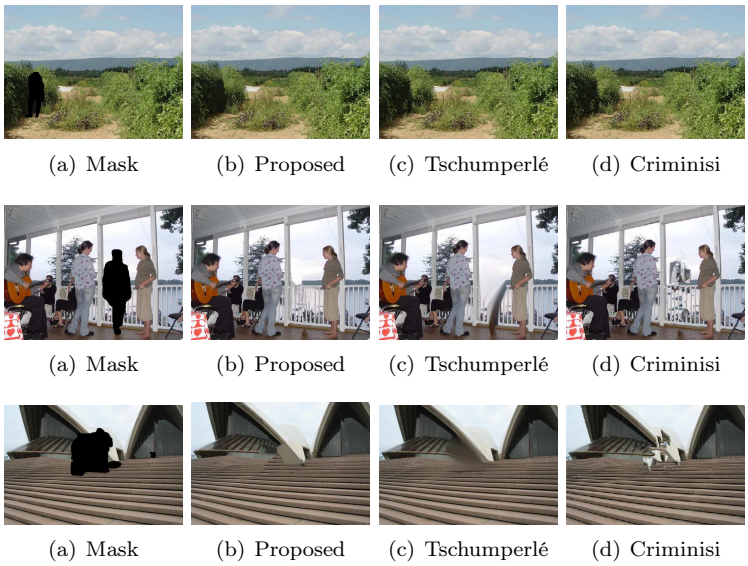
map have been taken into account. This explains the poor visual and objective (PSNR) quality of the decoded image. In the second image, relative subband importance has been take into account, in order to maximise the PSNR. This results in an improved visual quality with respect to the non-weighted case. However, the best perceptual quality (measured by SSIM) appears to be in the third image, where, at the cost of a very small loss in PSNR, we have an improved SSIM ($\Delta_2^{\text{SSIM}} = 2.3\ \%$) and we are able to keep some fine details the we would loose with the MSE-oriented technique. For example, we can remark that the nose contours are sharper, and that some detail (like the baboon's hairs in the highlighted box) are kept only when using the perceptual approach.

4.4 Results of the inpainting techniques

Figures 27 and 28 show the performance of the proposed method for inpainting. A comparison with Criminisi et al.'s[5] and Tschumperlé et al.'s approach [32] is performed. Figure 27 depicts the results. The approach in [32]preserves quite well the images structures but the apparition of blur is annoying when filling large areas. Regarding Criminisi's approach, results of both approaches are similar on the first picture. On the latter two, the proposed approach outperforms it. For instance, the roof as well as the steps of the third picture are much more natural than those obtained by Criminisi's method. The use of tensor and hierarchical approach brings a considerable gain[6]. Figure 28 shows results on pictures belonging to Kawai et al.'s database [14][7]. Compared to previous assessment, these pictures have a smaller resolution (200×200 pixels) than

[5]Matlab implementation available on `http://www.cc.gatech.edu/~sooraj/inpainting/`.
[6]see `http://www.irisa.fr/temics/staff/lemeur/` to have more results.
[7]`http://yokoya.naist.jp/research/inpainting/`.

those used previously (512×384). As illustrated by the figure, the unknown regions have been coherently reconstructed. Except for the last picture, structures are well propagated without loss of texture information.
Next studies will focus on stochastic or inhomogeneous textures, for which repetitions of structure are absent. In this case, template matching fails to replicate this kind of texture in a coherent manner. Instead of using an exemplar-based method, it would be probably better to synthesise such texture by using stochastic-based texture models. Pictures used in this paper as well as software are available on `http://www.irisa.fr/temics/staff/lemeur/` .



(a) Mask (b) Proposed (c) Tschumperlé (d) Criminisi

(a) Mask (b) Proposed (c) Tschumperlé (d) Criminisi

(a) Mask (b) Proposed (c) Tschumperlé (d) Criminisi

Figure 27: Comparison of the proposed approach with the approaches [10, 32].

(a) Mask (b) Proposed (a) Mask (b) Proposed (a) Mask (b) Proposed (a) Mask (b) Proposed

Figure 28: Results of the proposed approach on pictures proposed by [14].

References

[1] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *VCEG Meeting*, Austin, USA, Apr. 2001.

[2] T. Brox, J. Weickert, B. Burgeth, and P. Mrázk, "Nonlinear structure tensors," *Image and Vision Computing*, vol. 24, pp. 41–55, 2006.

[3] C. Cafforio and F. Rocca, "The differential method for motion estimation," in *Image Sequence Processing and Dynamic Scene Analysis*, T. S. Huang, Ed., 1983, pp. 104–124.

[4] C. Cafforio, F. Rocca, and S. Tubaro, "Motion compensated image interpolation," *IEEE Trans. Commun.*, vol. 38, no. 2, pp. 215–222, Feb. 1990.

[5] C. Cafforio and F. Rocca, "Methods for measuring small displacements of television images," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 5, pp. 573–579, Sep. 1976.

[6] M. Cagnazzo, T. Maugey, and B. Pesquet-Popescu, "A differential motion estimation method for image interpolation in distributed video coding," in *Proceed. of IEEE Intern. Conf. Acoust., Speech and Sign. Proc.*, vol. 1, Taiwan, 2009, pp. 1861–1864.

[7] M. Cagnazzo, W. Miled, T. Maugey, and B. Pesquet-Popescu, "Image interpolation with edge-preserving differential motion refinement," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Cairo, Egypt, 2009.

[8] R. Caldelli, A. De Rosa, P. Campisi, M. Carli, and A. Neri, "Perceptual aspect exploitation in video data hiding," in *Proceed. of Intern. Worksh. on Video Proc. and Quality Metrics*, Scottsdale, AZ, U.S.A., Jan. 2006.

[9] R. L. Claypoole, G. M. Davis, W. Sweldens, and R. G. Baraniuk, "Nonlinear wavelet transforms for image coding via lifting," *IEEE Trans. Image Processing*, vol. 12, no. 12, pp. 1449ï¿½â€“1459, Dec. 2003.

[10] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by examplar(based image inpainting," *IEEE Trans. On Image Processing*, vol. 13, pp. 1200–1212, 2004.

[11] A. D'Angelo, L. Zhaoping, and M. Barni, "A full-reference quality metric for geometrically distorted images," *IEEE Trans. Image Processing*, vol. 19, no. 4, pp. 867–881, Apr. 2010.

[12] S. Di Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, vol. 33, pp. 116–125, 1986.

[13] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[14] N. Kawai, T. Sato, and N. Yokoya, "Image inpainting considering brightness change and spatial locality of textures and its evaluation," in *PSIVT2009*, 2009, pp. 271–282.

[15] M. Kutter and S. Winkler, "A vision-based masking model for spread-spectrum image watermarking," *IEEE Trans. Image Processing*, vol. 11, no. 1, pp. 16–25, Jan. 2002.

[16] O. Le Meur and P. Le Callet, "What we see is most likely to be what matters: visual attention and applications." in *Proc. IEEE International Conference on Image Processing (ICIP)*. Cairo, Egypt., 2009.

[17] S. Mallat and F. Falzon, "Analysis of low bit rate image transform coding," *IEEE Trans. on Signal Processing*, vol. 46-4, pp. 1027–1042, Apr. 1998.

[18] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. on Signal Processing*, vol. 41-12, pp. 3397–3415, Dec. 1993.

[19] N. Mehrseresht and D. Taubman, "Spatially continuous orientation adaptive discrete packet wavelet decomposition for image compression," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Atlanta, GA (USA), Oct. 2006, pp. 1593–1596.

[20] S. Parrilli, M. Cagnazzo, and B. Pesquet-Popescu, "Distortion evaluation in transform domain for adaptive lifting schemes," in *Proceed. of IEEE Worksh. Multim. Sign. Proc.*, Cairns, Australia, 2008, pp. 200–205.

[21] ——, "Estimation of quantization noise for adaptive-prediction lifting schemes," in *Proceed. of IEEE Worksh. Multim. Sign. Proc.*, Rio de Janeiro, Brazil, Oct. 2009.

[22] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. of the 27th Annual Asilomar Conf. on Sig. Sys. and Compt.*, 1993, pp. 40–44.

[23] G. Piella, B. Pesquet-Popescu, and H. J. A. M. Heijmans, "Gradient-driven update lifting for adaptive wavelets," *Signal Proc.: Image Comm. (Elsevier Science)*, vol. 20, no. 9-10, pp. 813–831, Oct.-Nov. 2005.

[24] A. Said and W. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.

[25] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[26] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.

[27] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.

[28] T. Tan, C. Boon, and Y. Suzuki, "Intra prediction by template matching," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2006, pp. 1693–1696.

[29] ——, "Intra prediction by averaged template matching predictors," in *IEEE Conf. on Consumer Communications and Networking (CCNC)*, 2007, pp. 405–409.

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

[31] D. Tschumperlé, "Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's," *Int. Jounral of Comp. Vision*, vol. 68, no. 1, pp. 65–82, 2006.

[32] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with pdes: a common framework for different applications," *IEEE Trans. on PAMI*, vol. 27, no. 4, pp. 506–517, April 2005.

[33] M. Turkan and C. Guillemot, "Sparse approximation with adaptive dictionary for image prediction," in *IEEE Int. Conf. on Image Processing (ICIP)*, 2009, pp. 25–28.

[34] B. Usevitch, "Optimal bit allocation for biorthogonal wavelet coding," in *Proceed. of Data Comp. Conf*, Snowbird, USA, Mar. 1996, pp. 387–395.

[35] J. Wang, D. M. Chandler, and P. Le Callet, "Quantifying the relationship between visual salience and visual importance." in *Proc. SPIE Electronic Imaging.* San Jose, CA, USA., 2010.

[36] J. Weickert, "Coherence-enhancing diffusion filtering," *International Journal of Computer Vision*, vol. 32, pp. 111–127, 1999.

[37] J. Weickert and H. Scharr, "An anisotropic diffusion algorithm with optimized rotation invariance," *Journal of Visual Communication and Image Representation*, vol. 13, no. 1-2, pp. 103–118, 2002.

[38] Y. Wexler, E. Shechtman, and E. Irani, "Space-time completion of video," *IEEE Trans. On PAMI*, vol. 29, no. 3, pp. 463–476, 2007.

[39] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[40] T. Wiegand, G. Sullivan, B. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13-7, pp. 560–576, July 2003.

[41] T. Wiegand, G. Sullivan, and A. Luthra, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 / ISO/IEC 14496-10 AVC)*, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Geneva, CH, May 2003, Doc. JVT-G050r1.

[42] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.