Object based coding through multigrid representation

X. Marichal, C. De Vleeschouwer*, T. Delmot and B. Macq

Laboratoire de Télécommunications et Télédétection
Université catholique de Louvain
Bâtiment Stévin - 2, place du Levant
B-1348 Louvain-la-Neuve
Tel.: +32 10 47.41.05 - Fax: +32 10 47.20.89
E-mail: devlees@tele.ucl.ac.be

ABSTRACT

This paper presents a very low bit-rate coding algorithm based on image split in order to represent it through an adaptative multigrid supported by a binary tree structure. Indepedently of its tree representation, the picture is segmented via a watershed procedure and several criteria are combined to automatically extract interesting areas of the image. This object information is not transmitted but used to reduce picture complexity, and therefore the bit-rate, while keeping a good subjective quality. This is achieved by a merge procedure which homogenizes values of the tree subblocks belonging to a same non-interesting object. This treatment affects both intra- and inter-images. For intra-images, the resulting tree structure is entropy coded while its leaves are encoded through a DPCM procedure followed by a multi-huffman coder. For inter-images, a motion field is adaptated by an adaptative block matching algorithm which is a kind of BMA for which blocksize is chosen in order to reach a sufficient level of confidence. Residues, essential to correct motion compensation artifacts, are sent through local intra-trees or, if the bit-rate allows it, through DCT blocks, allowing to reach an arbitrary level of quality. During the reconstruction step, an object oriented approach combined with the use of overlapping functions allows to reduce block artifacts while keeping sharp edges.

Keywords: low bit-rate, tree, video coding, multigrid, merge, interest, object, BMA, overlapping functions.

1 INTRODUCTION: VERY LOW BIT-RATE IMAGE CODING

Very low bit-rate (VLBR) transmission of image sequences remains an essential feature in the framework of MPEG-4. It concerns transmissions under $64 kbits/s$($48 kbits/s$ for video information and $16 kbits/s$ for audio information). Such bit-rates would enable video-telephony,video-conferencing, ...even accross mobile communication channels. To achieve such VLBR transmissions, compression is required; a process that implies the following steps in the transmission chain:

- Compression
  - First, the image is analyzed: the **analysis** may consist in frequential waveforms analysis, in spatial contours and texture analysis, or in an estimation of the motion field between two successive frames.

*Research of C. De Vleeschouwer is supported by a grant of the belgian NFS (FNRS), offered by Alcatel Bell.

- The resulting parameters are then transcribed into the format required by the decoder. This **transcription** may be block-based like in a classical MPEG[1] or H.263[2] encoder, or tree-based, object-based,...
- Finally, the transcription is encoded (**coding**) and sent to the decoder: this coding step certainly implies **source coding**[3] that aims at reducing the bit-rate without corrupting the data and may also include some **channel coding** that introduces special redundancies to overcome the transmission errors.

- DECOMPRESSION
  - After **decoding**, the decoder gets back the transcription generated by the coder.
  - Then two possibilities occur:
    * Either, the transcription is merely reversed in order to obtain a **description** of the images that are close to the original ones (coding-decoding phase is assumed to be lossless).
    * Or a **reconstruction** stage may be added: the transcription is no more simply reversed but specifically treated in order to better take advantage of the available information. Reconstruction is different from *post-processing*, which tries to improve the image quality afterwards, without taking into account the transcription structure.

All these parts of a VLBR coding algorithm may vary from one implementation to another. However, the different steps have to be coherent if one wants the scheme to be efficient. So, the type of analysis performed often guides the rest of the algorithm.

In general, three types of pictures are used by a video coding scheme:

- INTRA-IMAGES: At least the first image has to be encoded on its own basis. With these images, the coder can only exploit the **spatial correlation** inherent in the picture itself. In fact, an intra-image is compressed as an isolated picture (cfr JPEG[4] for example).
- INTER-IMAGES: These images are specific to video sequences. They allow the coder to exploit the **temporal correlation** between the current image and the previous one in the sequence. The classical tool used to achieve compression on such images is motion estimation/compensation which allows to have an estimation of the current image on the basis of the previous one(s).
- RESIDUES: As the estimation/compensation of a inter-image is not perfect, one can compute the difference between the estimate and the original image: the so-called Displaced Frame Difference (DFD) has to be sent on its own basis, i.e. as an intra-image.

According to this, one can say that the scheme presented in figure 1 is typical of most video coders.[5]

Among the existing algorithm, two different philosophies have quickly emerged: the first class is the **block-based** one, where pictures are divided in subblocks according to an *a priori* grid. Thereafter, every subblock may be independently coded according to its own characteristics. These techniques are very efficient in terms of computational cost and realize interesting compression factors. However, they suffer from one major annoyance: the visual blocking artifacts that appear in the decoded sequences. The search for higher compression ratios while preserving essential features of the sequence has led to a second group: the **segmentation-based** or **object-based** one: such schemes[6,7] aim to better preserve essential characteristics: very high compression ratios are obtained by removing small objects and by encoding textures coarserly, while contours are considered as essential features for image description. After a detection of the different pictured regions (objects), every one is separately coded: its shape is first described and is followed by a texture information. The segmentation-based algorithms have a rather high computational cost and do not actually achieve equivalent compression factors. But their visual results get rid of the blocking artifacts. In addition to that, their region-oriented understanding of the pictures allow a wide range of multimedia functionalities.

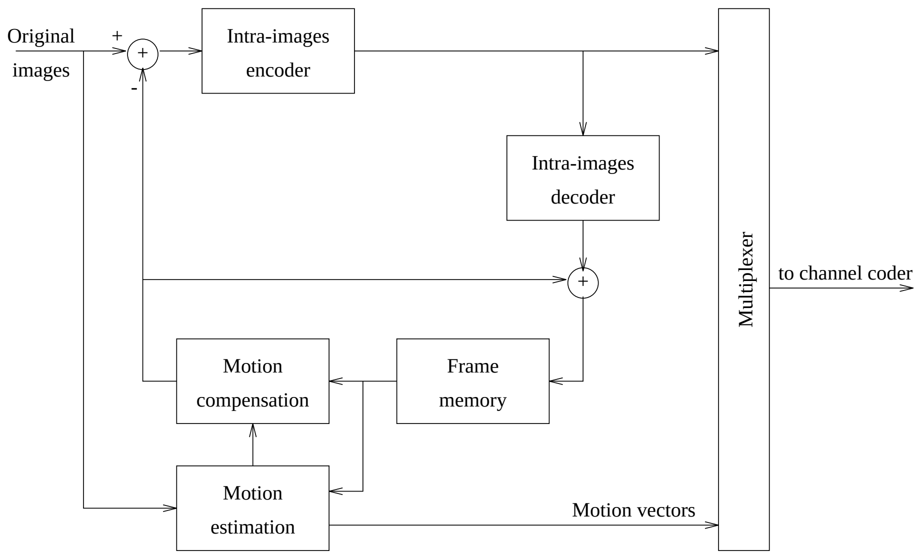


Figure 1: Typical VLBR video coder using intra-images, motion estimation/compensation and residues.

2 ALGORITHM PHILOSOPHY

Initially developed by M.P. Queluz, B. Simon and B. Macq in the context of the European COST 211ter research group, the UCL proposed a new approach for VLBR video coding in which spatial information and temporal changes are encoded using similar tools.[8] In this spatio-temporal integrated approach, adaptive multigrids are used for both predicted images (inter-images) and intra-images. The intra-images are encoded using tree segmentation and transcripted on an adaptive sampling grid (with variable sampling mesh). The inter-images are mainly encoded by a reliable motion field analysis and transcribed in a way similar to the intra transcription. The algorithm received thus the name of COMIS (Coding On Multigrids Image Sequences) and was essentially a spatio-temporal block-based coding technique.

However, it appeared important to add some segmentation-oriented features in order to make the scheme open to interactivity, content-based manipulation and editing, scalability,... The main refit consists in adding an **internal** notion of objects and in a reorganization of the scheme towards an improved coding efficiency.

Figure 2 shows the codec scheme. One can notice that a "region model" box is present in both the encoder and the decoder sides. Therefore, **no object information needs to be transmitted**: the object detection is simultaneously generated on both sides (for instance, via a morphological Watershed procedure[9]), every time the decoder receives additional picture information. Such a combination enables the coder to be very efficient: it retains the cheap picture description provided by the multigrid representation, while enhancing the **analysis** and the **reconstruction** stages with an region-oriented understanding of the pictures.

3 DESCRIPTION OF THE INVOLVED TOOLS

In fact, according to the MPEG-4 syntactic description,[10] an algorithm may be considered as a collection of tools or elementary boxes that are assembled together in order to obtain the final algorithms. The present sections

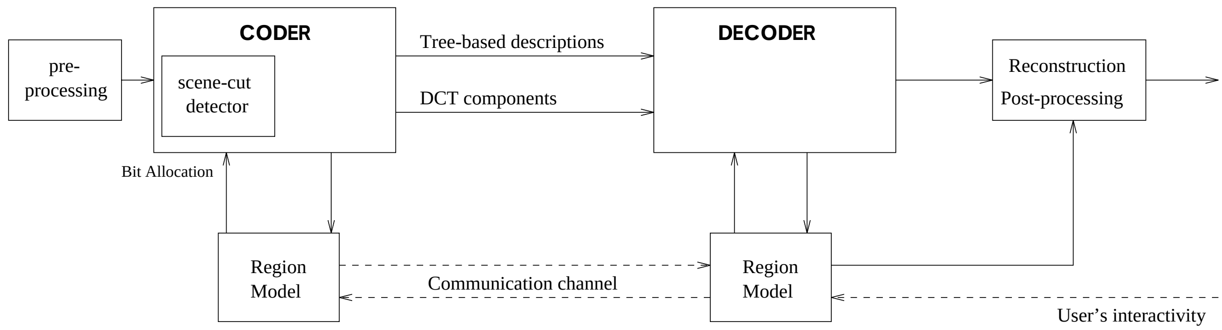


Figure 2: Scheme of the COMIS codec.

reviews the most important tools used in the present coder. For clearness purposes, the tools are described in the order of their appearance in the scheme, following the analysis, transcription and reconstruction phases. Of course, intra-images treatment is described prior to the inter-images one.

3.1 Intra-images

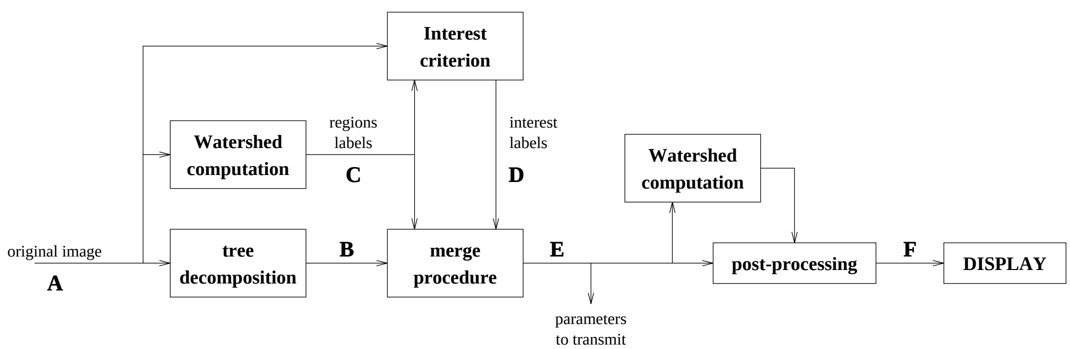


Figure 3: Block diagram to intra-code a picture.

The first image of the sequence must of course be encoded as an intra-image. For the subsequent images, a **scene-cut detector** analyses first if the new image is still coherent with the previous ones. If it is, the image will be inter-coded, otherwise the coder considers that a scene-cut occurred: the new image is so much different than the previous ones that it is better to encode it as an intra-image.

Figure 3 describes the way a picture is treated in the intra-mode. The following subsections emphasize the different tools. All the images representing intermediate results will always refer to some letters (A-F) of the present diagram.

3.1.1 Multigrid representation: tree decomposition

The objective of this representation is to identify the regions with uniform luminance or with random textures. At first, the image is split into non-overlapping 16×16 blocks[1]. Starting from such large blocks, the aim is to obtain homogeneous blocks components by a succession of binary decisions, that can be described using a binary tree. This tree ensures the correspondence with a multigrid representation of the picture (Figure 4).

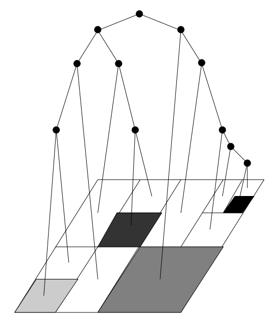


Figure 4: Example of binary tree segmentation.

The splitting process is repeated until reaching homogeneous blocks or a block of predefined minimal size. For a $i \times j$ block, let m be the luminance mean of the block . Let α be the percentage of pixels in the block with a luminance value l, such that $|l - m| \geq T_1$, with T_1 an *a priori* parameter (typical value = 10). The block is decided to be split if $\alpha \geq \alpha_T$, with α_T another *a priori* parameter (typical value = 0.05). Once the block is considered as homogeneous, the luminance of all its pixels is set to the mean of the luminance values on the block and is attached to the appropriate leave of the tree.

Another criterion is added to prevent from splitting regions with random textures, that are not perceptually important: if v is the variance of the $i \times j$ block and $v_{r,s}$ the variance of the block (r, s) contained in that block. If $v \geq T_2$ and $(|v_{r,s} - v|/v) \leq T_3$ for every block (r, s), the block is considered as possessing a random texture and is not split (T_2 and T_3, are *a priori* parameters: typical values = 3000 and 0.3).

Figure 5.a and b illustrates the principle and the result of the algorithm. Another advantage of the algorithm is the use it can make of the correlation between luminance and chrominance components: for color images, only one single tree is used to describe the three components decompositions [2].

3.1.2 Split and merge

If the reader has a close look at the tree decomposition of "Foreman" image at figure 5.g, he/she will notice the image is much too complex to be transmitted over a VLBR channel. A way to reduce this complexity is to identify coherent areas of the picture and to merge the subblocks of the tree decomposition belonging to a same region (or object). Indeed, the tree description of the image (cf. section 3.1.1) can be considered as a split procedure. A classical way of dealing with such split algorithms is to combine them with a merge procedure that aims at homogenize their result and correcting their errors due to local traps.

[1] Any power of two size would be convenient: if 16×16 is used, it is because analysis of test images has shown that larger blocks are always unhomogeneous in a QCIF (144×176 pixels) picture.

[2] Generally, a color image may be described by a combination of three chromatic stimuli. For instance the color television uses the red, green and blue primary colors. Another possibility is to use the Y luminance, plus two difference chrominance signals.

First, the merge procedure took as arguments the result of the split and a Watershed analysis of the original image. Results obtained are really unpleasant. Although it was convenient to simplify the background, the merger of the face is much too coarse. It is thus time to introduce a region-oriented understanding of the picture. Once the regions will be classified, the algorithm will be able to preserve a good quality on the interesting regions (speaker head,...) while being coarse on the other regions.

3.1.3 Interesting regions determination

An interest criterion is set up. It can automatically decide what is important and what is not on an original image. Three criteria are combined in order to provide the final classification:

- the **image border criterion**: based on the facts that the most important part of an image in a video sequence is correctly centered and that the eye precision is better in the center of the vision area than in the extremities, it rejects the regions which have a huge amount of their pixels along the picture's border.
- the **interactive criterion**: it can reinforce the previous criterion or fight against it. Its principle is to eliminate regions with no pixels inside a pre-defined window of the image. This criterion is the interactive criterion par excellence as the user can easily select another "interest window".
- the **face texture criterion**: mainly designed for videophone or video-conference, this criterion rejects all the regions whose chrominance components do not coincide with a set of skin samples.

These criteria result in a classification of the image regions that is used in the split and merge algorithm to prevent the merging of interesting areas. On Figure 5.f and h , we observe that in spite of the drastic simplification of the tree decomposition complexity, a good subjective quality is kept for the corresponding picture.

3.1.4 Coding

These parameters have now to be transmitted to the decoder in order to allow this one to reconstruct the picture. Two elements are necessary to fully describe the image: the binary tree structure and the attached leaves values.

The tree is entropy coded with the M-Coder[11],[12] which is a set of Truncated Run Length (TRL[13]) codes parametrized by their value of mode m. The bits are multiplexed in the coder according to an efficient probability estimation model. In order to track the changes in local statistics and to have a generic coder, the tables used by the probability estimation scheme are constantly updated according to the last encoded tree.

The leaves obtained in the analysis-transcription phase are first decorrelated before coding. This is done by optimum linear prediction, using the nearest neighbors in a causal neighborhood. The resulting prediction errors to be encoded are, therefore, linearly decorrelated and statistically weak. These prediction errors are encoded using the Universal Variable Length Coding (UVLC[14]). The UVLC is based on the fact that quantized values coming out of a prediction process are likely to be close to zero. Therefore, the probability for a bit of the quantized value to be one depends on the values of the most significant bits of the value. The UVLC makes also use of the TRL codes, via a set of Adaptive TRL (ATRL[15]) codes. One of the main advantages using such TRL based coders is that it leads to very simple decoders, implemented by counters.

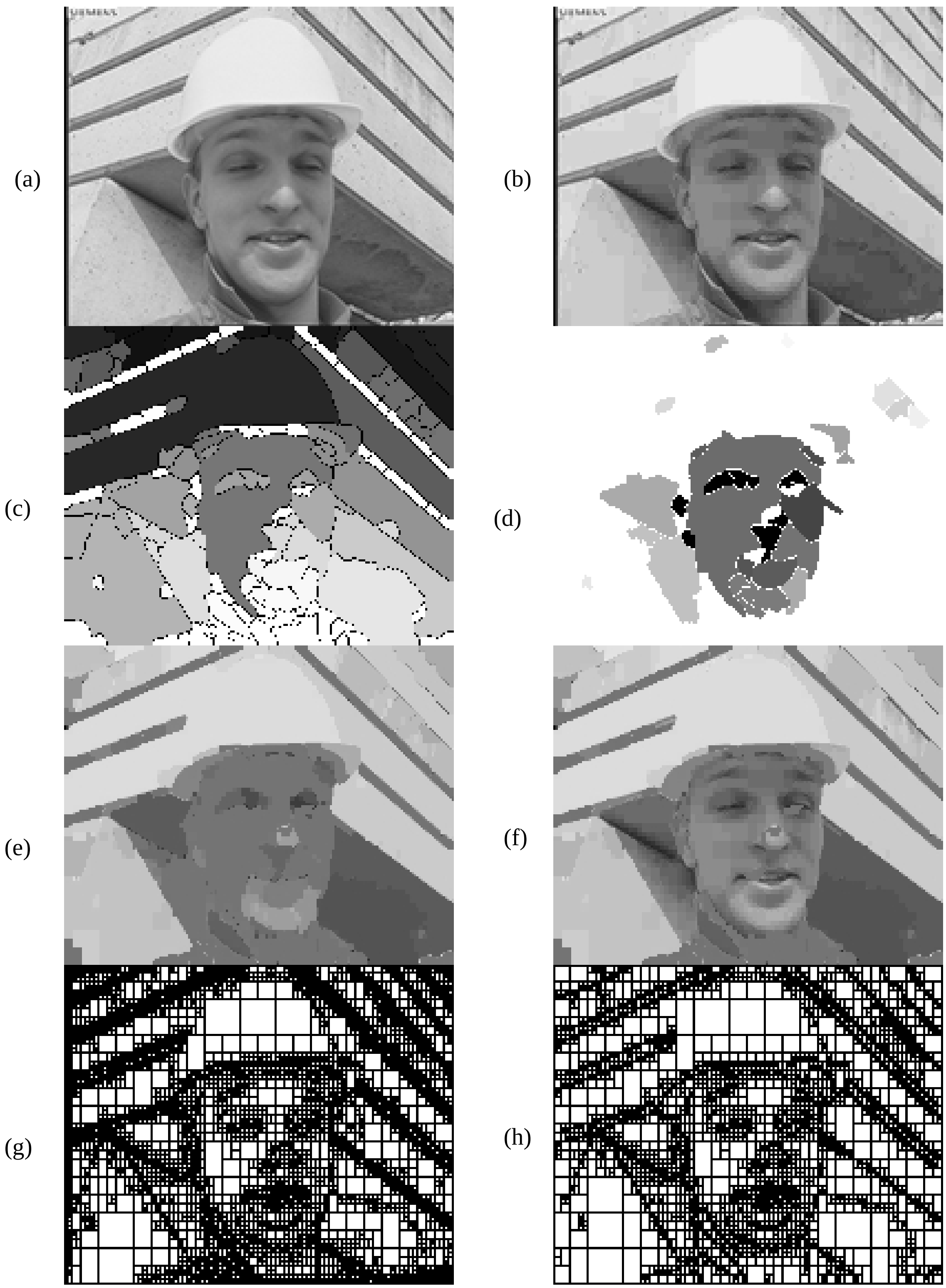
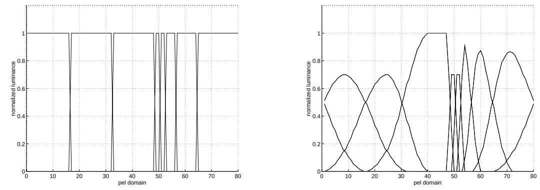


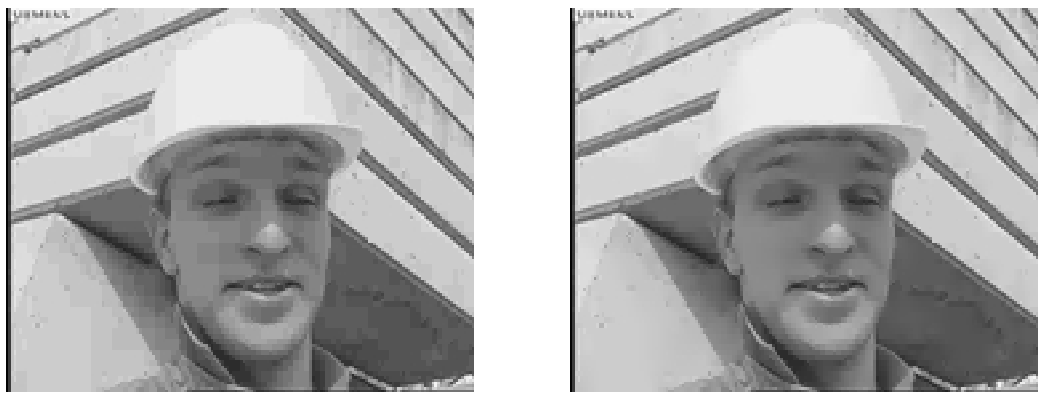
Figure 5: Intra treatment on "Foreman": (a) Original (A in fig 3), (b) split image (B in fig 3), (c) watershed (=location of coherent areas = C in fig 3), (d) interesting areas (D in fig 3), (e) full merge, (f) merge with protection of interesting areas (E in fig 3), (g) multigrid of original split, (h) multigrid structure after merging.



(a) normal impulse response

(b) longer and adaptive impulse response

Figure 6: Longer impulse response for overlapping reconstruction.



(a)

(b)

Figure 7: Effectiveness of reconstruction: split image (a), reconstructed image (b) (F in figure 3).

3.1.5 Reconstruction

Describing tree segmented images as the juxtaposition of variable size blocks (with one single mean value associated to every block) leads to large blocking artifacts, as one can see on the pictures of figure 5.b and figure 7.a. So, the description would better not be the mere reverse operation of the transcription.
The blocking artifacts may be eliminated by the use of overlapping functions in a reconstruction process. These functions have to respect transitions between small and large blocks but will also have to be spread out in low resolution areas. The solution consists in replacing the step impulse response by a longer impulse response but only locally, in order to respect transitions in the image. This locally longer impulse response permits an overlap between interpolation functions; and the local character of the insertion of that function allows a control in the resulting overlap, as shown in figure 6. Low resolution areas receive an important overlap while high resolution areas need to keep the interpolation functions disjoint. Object information, obtained from the decoded picture, is also used to reduce smooth effects along regions' borders in comparison with smoothing inside the object.

The improvement offered by this reconstruction technique is depicted on figure 7 and is clearly important for human visual perception.

3.2 Inter-images

As mentioned in the introduction, the aim of the inter-images is to exploit the temporal correlation between successive images. It is possible to inter-code an image every time it possesses a reference image.

3.2.1 The Adaptive Block-Matching Algorithm

The motion estimation used in COMIS is based on the well known block matching algorithm (BMA).
Two substantial improvements have been added.
First, on the basis of a tree based split procedure, a hierarchical structure of block sizes is considered (with pre-defined minimum size), which allows to overcome the problem of having different moving objects in the same block.
Secondly, a merge procedure is applied to correctly propagate the motion vectors from blocks with reliable motion to blocks with uncertain motion. The motion vector field is further smoothed by a non-linear filtering operation that preserves discontinuities.
Another feature of the scheme is a two stage global/local motion estimation.

The used scheme is thus called an Adaptive Block Matching Algorithm (ABMA,[16]). The algorithm, whose result is presented in figure 8, with comparison to classical BMA, mainly consists in six steps:

1. a global motion estimation: this step aims at determining the motion parameters resulting from the camera pan and zoom.
2. a detection of changing regions that have a local motion (independent from the camera motion).
3. a local motion estimation for all these regions, via a BMA using $m \times m$ pixels blocks.
4. a computation of the motion certitude for every block.
5. a nonlinear filtering of the resulting motion field in order to homogenize it with a propagation of the certain vectors on the uncertain ones, while preserving the essential motion discontinuities.
6. a further splitting of the blocks whose motion is still not certain enough. For these blocks, the algorithm returns to step 3 with $m = \frac{m}{2}$.

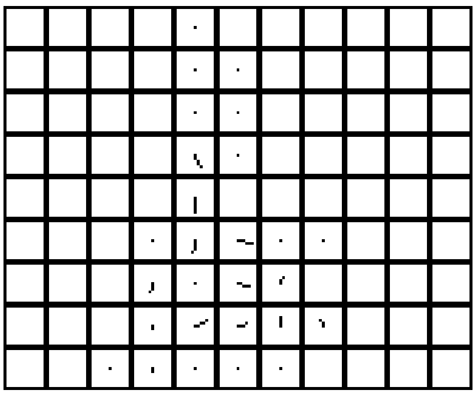
3.2.2 Pre-treatment before motion estimation

To simplify the new coming image to the same level of complexity than the previously decoded ones, a pre-treatment is applied to the image before it is motion estimated. The image is treated as an intra-image. But, once arriving at point E of figure 3, the parameters are not sent. The resulting image is used for the motion estimation itself. Figure 9 depicts the COMIS scheme for inter-coding.

During intra treatement of the picture, an interesting regions determination (cf. section 3.1.3) is needed. Two criteria complete this tool for inter-images. A **motion criterion**: based on a succession of motion estimations between the images at the highest frequency (even if the sequence is coded at $8Hz$, the criterion will compute three mere ABMA between images at $25Hz$), it eliminates regions with no motion and very small gradient (the Human Visual System do not focus on this areas) as well as the regions with very important motion and high gradient (that are to noisy to be correctly perceived). A **continuity criterion** uses the mask obtained for the previous image to give some temporal stability to the final criterion.

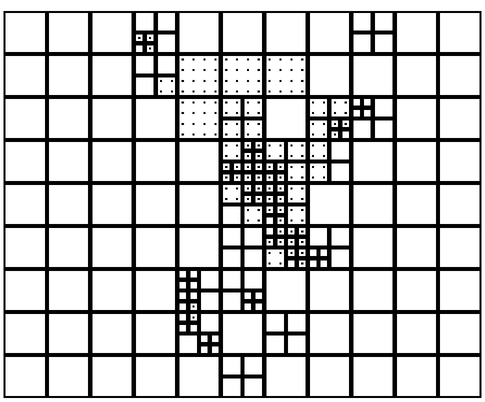
(a)

(b)



(c)

(d)



(e)

(f)

Figure 8: Comparison of motion estimation/compensation using the BMA with 16×16 blocks and using the ABMA, starting with 16×16 blocks, stopping at 4×4 blocks: (a) Original image at time $t-1$, (b) Original image at time t, (c) Estimated motion field (one vector per block) for BMA, (d) Resulting compensated image, (e) Estimated motion field for ABMA and (f) Resulting compensated image.

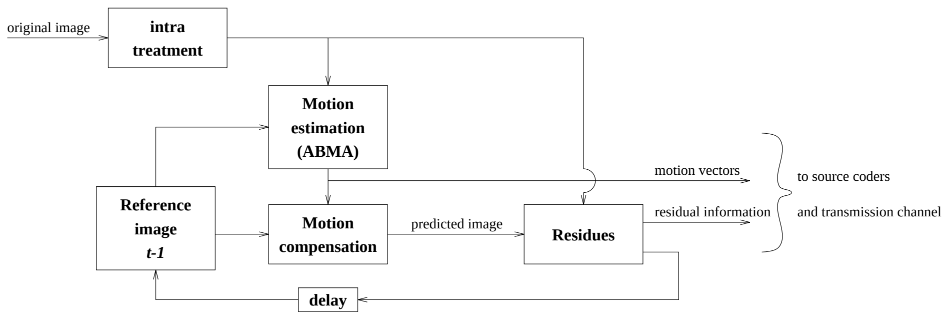


Figure 9: Block diagram to inter-code a picture

3.2.3 Motion transcription and coding

Since the final goal is compression, the description of the motion information is a key problem and efficient ways of transmitting the displacement vectors must be found. The high level of smoothness provided by the ABMA method justifies the use of contour/content coding of the motion vector field. In order to take advantage of what is done in the intra mode, the same kind of transcription and coding is used: a binary-tree decomposition is applied to the estimated motion field, every leaf of the tree corresponding to a region with the same motion vectors. The entropy coders already used in the intra coding (cf. section 3.1.4) are also applied to encode the inter tree descriptions.

3.2.4 Motion prediction failure areas: the residues

Several schemes are actually under study. Two possibilities have been developed.
The first one is to send the DFD as an intra image. The DFD is the difference between the compensated picture at time t and the original one already intra-treated. This DFD will then be intra-segmented and merged before being sent. The advantage of such a global DFD will be to avoid to obtain too many blocking artifacts.
The second possibility is to send only localized refreshes on selected blocks. A mini intra tree is sent for pre-defined size blocks which belong to crucial part of the picture.
The main drawback of sending such intra-segmented residues is that the scheme will never be able to design fine textures, high frequency details,... and recover the quality of the original images. Such elements would have to be transmitted if the bit-rate allows it. In this case, other residues would be sent using a classical Discrete Cosine Tranform (DCT[4]) with appropriate quantization matrix and run-length codes.

3.3 Buffer regulation

Although this part of the scheme, still under development, won't be longer discuted, one note three possibilities of regulating the bit-rate via a buffer (what is a classical way to regulate the bit-rate, cf.[1]):

- changing the intra quantization step and the minimal size of the blocks;
- increasing/decreasing the percentage of the image that is kept in plain resolution through the merge process;
- increasing/decreasing the maximum number of blocks that can be refreshed.

4 RESULTS AND CONCLUSION

The present paper introduced a scheme for very-low bit-rate image coding. Although the scheme is yet not completely implemented, its first results are promising and the range of $48 - 64kbits/s$ is reached with a pretty good quality. One of the major feature of the scheme is its intention to take into account the subjectivity of the viewer. This has been achieved by an automatic detection of the regions that may appear as important and a focus of the bit-rate on these areas. The next step that is under development by now is to take into account the behavior of the Human Visual System (HVS) to a moving scene. Let consider two images that are slightly different. While looking separately to every image the HVS will notice the difference. Once the images are consecutive in a video sequence, the differences cause a "flash" effect that is particularly unpleasant to see if it does not correspond to a moving area. Such effects are still present in the COMIS scheme due to the intra pre-processing and are going to be eliminated.

5 REFERENCES

[1] D. Le Gall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[2] Telecommunication Standardization Sector of ITU. Draft ITU-T Recommendation H.263. *ITU Recommendations*, July 1995.

[3] R.G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., New-York, 1968.

[4] G.K. Wallace. The JPEG Still Picture Compression Standard. *Communications of the ACM*, 34(4):30–44, April 1991.

[5] C. Chen and T.R. Hsing. Review: Digital coding techniques for visual communications. pages 1–15, 1991.

[6] MORPHECO. Morphological segmentation-based coding of image sequences. *COST 211ter European Workshop on New Techniques for Coding of Video Signals at Very Low Bitrates - Hannover*, pages 2.2.1–2.2.8, December 1993.

[7] H.G. Musmann, M. Hotter, and J. Ostermann. Object-oriented analysis-synthesis coding of moving images. *Image Communication*, 1(2):117–138, October 1989.

[8] B. Macq, M.P. Queluz, and B. Simon. Very low bit-rate image coding on adaptive multigrids. *Signal Processing: Image Communication*, 1995. Accepted for publication.

[9] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.

[10] AOE SubGroup. Proposal package description (ppd)- revision 1.0. *MPEG 4 meeting*, November 1994.

[11] X. Marichal, B. Macq, and M.P. Queluz. Generic coder for binary sources: the m-coder. *IEE Electronics Letters*, 31(7):544–545, March 1995.

[12] M.P. Queluz, X. Marichal, and B. Macq. Efficient entropy coding of tree data structures. pages 478–481, Sacramento, September 1994. Picture Coding Symposium.

[13] F. Ono, R. Ohnishi, and Y. Ueno. Optimization of Facsimile Data Compression. *Conf. Record of Nat. Telecomms. Conf.*, pages 49.1.1–49.1.6, November 1977.

[14] B. Macq. A universal entropy coder for transform or hybrid coding. *Picture Coding Symposium*, pages 12.1.1–12.1.2, 1990.

[15] H. Tanaka and A. Leon-Garcia. Efficient Run Length Encoding. *IEEE Transactions on Information Theory*, 28(6):880–890, November 1982.

[16] M.P. Queluz and B. Macq. Signal-adapted motion compensation for video compression. Paris, September 1992. ISSES, URSI.