Single-Frame Prediction for High Video Compression

R. Leonardi *
Signals & Communications Lab., Dept. of Electronics for Automation,
University of Brescia, Brescia, I-25123, Italy
Ph: +(39-30) 371-5434 Fax: +(39-30) 380-014
e-mail: leon@bsing.ing.unibs.it

ABSTRACT

In this abstract, we present a novel technique to encode video sequences, that performs a region-based motion compensation of each frame to be encoded so as to generate a predicted frame. The set of regions to be motion compensated for a given frame has been obtained through a quadtree segmentation of the motion field estimated between a single reference frame (representing a typical projection of the scene) and the frame to be encoded. This way, no DPCM loop in the temporal domain is introduced, avoiding the feedback of the quantization errors. Under the assumption that the projection of the scene on the image plane remains nearly constant, only slight deformations of the reference frame occur from one frame to the next, so that very limited information needs to be coded: 1) the segmentation shape; 2) the motion information. Temporal correlation is used to predict both types of information so as to further reduce any left redundancy. As the segmentation may not be perfect, spatial correlation may still exist between neighboring regions. This is used in the strategy designed to encode the motion information.

The motion and segmentation information are estimated on the basis of a two stage process using the frame to be encoded and the reference frame: 1) a hierarchical top-down decomposition, followed by 2) a bottom-up merging strategy. This procedure can be nicely embedded in a quadtree representation, which ensures a computationally efficient but rather robust segmentation strategy.

We show how the proposed method can be used to encode QCIF video sequences with a reasonable quality at a 10 frame/s rate using roughly 20 kbit/s. Different schemes for prediction are compared pointing the advantage of the single reference frame for both prediction and compensation.

1 INTRODUCTION

There is a variety of applications using video information where the available bandwidth remains very limited, typically 19.2 kbit/s. In this context, algorithms derived from traditional video compression schemes are likely not to perform adequately at these bit-rates.

Efforts have been made to modify CCITT recommendation H.261 for video-telephony so as to work at bit-rates lower than 64 kbit/s [1], by changing the video sequence format and adapting the parameters of the coding scheme. H.261 decoding recommendation is based upon a hybrid motion-compensated predictive strategy with DCT coding of the prediction error. The simulation results for low bit-rate targets ($\sim$ 20kbit/s) of the corresponding TMN2 video codec test model are still objectionable at this point especially when significant activity is present in the source material. A still troublesome issue is to design a buffer control strategy at these bit-rates that can manage large amounts of new information. Besides, the efficiency of TMN2 is limited to source material which remains

*This work was supported by the Visual Communications Research Dept. of AT&T Bell Laboratories and the Italian Council for National Research.

0-8194-1766-1/95/$6.00

in the video-telephony context. Even if one considers other state of the art video coding algorithms (such as in MPEG1 or MPEG2), that have been used to encode other classes of video sequences, it is difficult to imagine how these can be modified so as to work at lower bit-rates. In fact the recommendations provided in MPEG1 or MPEG2 require a tremendous amount of overhead information. MPEG1 and MPEG2 provide efficient ways of encoding video information by:

1. improving the quantization strategy of the information to be coded by making such quantization adaptive to the frame content (slice information).

2. offering a better rendering of motion information through the use of motion-compensated interpolation techniques.

Both these coding strategy deal with image sequences in a block-based fashion.

Improving compression for image sequences requires using additional properties than the simple block-based motion information, and underlying stationarity assumption which are commonly considered in the more classical schemes. Some attempts have been made to insert a priori knowledge for classes of video sequences, as in the wireframe models [2] or the head-and-shoulder models, used in the analysis-synthesis proposal of SIMOC [3]. We do not want to restrain the approach here to a limited class of video sequences, but use an often forgotten property of video sequences generated from a natural scene, i.e. the structure information. By structure, we propose to identify the spatial arrangement of pixels that exhibit a common property (texture or motion are two examples), so as to obtain an object based decomposition of each frame, i.e. a segmentation of such frame. Motion compensated prediction can then be applied to each region of the segmentation, by using a previously received *reference frame*.

No matter how sophisticated the motion model, prediction cannot occur in general without introducing errors (e.g., problem of uncovered areas; motion of objects non parallel to the image plane; ...). To avoid the feedback of such errors in a temporal prediction loop, we introduce the idea of using a fixed *reference frame*. As the effect of such errors can only be perceived for each decoded frame with no consequence on the future frames being decoded, quantization can be very large, at most one could decide not to code any error information, which could lead to a substantial gain in compression in the low bit-rate context. Coding is then limited to describe the segmentation shape together with the motion information (from one frame to the next.) For low bit-rate applications, both segmentation and motion must remain as simple as possible, and temporal redundancy should be used to further reduce their coding cost. This is achieved in the proposed scheme by

1. limiting the motion model between any object from the frame to be encoded and the reference frame to a pure 2D translational model;

2. generate the segmentation itself on the basis of motion information so that an additional segmentation map need not be transmitted to identify regions having constant motion field.

Last but not least, a relatively elaborate prediction model has been designed to take into account the temporal correlation of both motion and segmentation in the coding process. Finally, the TM discussed the issue of updating the reference frame so as to increase the performance of this video coding scheme.

The paper is organized as follows: In the next section, we present the different stages of the coding process. We describe then the motion segmentation process using a quadtree decomposition of each frame to be encoded. Section 4 discusses then the coding strategy with an outlook of some quality compression performance trade-offs. Section 5 presents some simulation results, while finally Section 6 provides a summary and a direction for future research.

2 THE OVERALL CODING MODEL

If translational motion has been extensively used for block-based motion compensation, more elaborate motion models have been rarely considered, so that it is difficult to describe deformations of objects between successive frames. To keep a fixed block size has been another limitation for the correct estimation of motion information, as objects in an image are rarely squares of fixed size; in particular, fixed size blocks may contain 2 or more

noving objects, leading to an incorrect motion compensation, which results in a substantial reconstruction error fter compensation, in the most traditional video compression algorithms (H.261, MPEG1, MPEG2, ...)

This has suggested to some to identify objects in a given sequence by appropriate segmentation techniques hat rely on a joint motion and luminance analysis. These objects can then be tracked over the sequence of rames. This approach allows still for interactive communication, as long as the motion tracking process remains imited to a few frames (typically 3-5 frames).

.1 Segmentation based coding of video

One can at this point consider a "group of frames" (GOF) at the time, and describe it using a 3D segmentation of the spatio-temporal video signal that they define together with a Y-Cr-Cb model of each region in the egmentation. Methods have been proposed to obtain a 3D segmentation [5], but they remain of significant complexity, and have not been considered in the coding context but rather for enhancement or frame rate conversion. 'rediction could be made from one GOF to the next so as to reduce the amount of information to be coded.

.2 Region-based motion compensated video coding

Another solution is to estimate from one frame to the next the motion of objects. As most object shape emains unchanged over time, it is expected that the 2D segmentation of their projection on the image plane an be predicted from one frame to the next according to the motion the objects have [8]. A region based motion ompensation scheme can be easily derived in this case. Square blocks are simply replaced by arbitrarily shaped egions which exhibit uniform motion with respect to a previous reference frame.

Given the 3-dimensional structure of the natural scenes from which the frames are generated, it is clear that ven though the information is treated in an object based fashion, the motion description is not sufficient to enerate from a single frame, any new frame. In effect, there are uncovered areas that represent new information o be transmitted. Transmitting video information requires therefore sending three items [3,9]:

1. the object shape or *region description* resulting from a segmentation process, on a frame by frame basis.
2. the *motion description* of each region.
3. the *error description*, i.e. uncovered background information, or error information resulting from an inappropriate segmentation estimation, or region-based motion compensation.

In this scheme, the first step is to segment a frame into a set of regions having uniform structure. Already at his stage, it is necessary to distiguish intra-frames (no temporal prediction being used) from the predicted frames. ntra-frames are sent as single 2D pictures, and can be coded using traditional coding techniques (JPEG, coding f I frames in MPEG1 or MPEG2 standards), or more sophisticated coding techniques for static images [10,11], can e considered. In this context, segmentation based coding can be used for intra-frames, so as to keep a similar oding strategy (segmentation based) than the one used for the predicted frames. In general, the segmentation is stimated in such a case so as to have slowly varying surfaces in each region that can be easily be modelled using olynomial functions. As an example we show in Figure 1 a quadtree based decomposition of the first frame of he "grandma" sequence in QCIF resolution (144×180).

The second stage of a region based motion compensated video coder is to decompose any future frame into a et of regions that can be motion compensated predicted from some other frame. Motion compensated prediction s applied to the regions defined in the frame to be encoded. Therefore, such regions should have a unique motion haracteristics, which means that they should have been segmented on the basis of such motion information. The second stage should therefore be designed as a joint estimation segmentation process, where one alternates rom the segmentation to the motion estimation stage [6,14]. Figure 2 presents the principle of such motion egmentation operation. If the quality of the region based motion compensated prediction is high enough, the urrent estimate of the segmentation map is retained, together with the region based motion field. Otherwise the egmentation module uses both the error frame and the motion field structure to modify the previous estimate. A ew region based motion estimate can then be computed to obtain another prediction of the current frame. This oint segmentation-motion estimation procedure can stop when the error frame energy becomes sufficiently small. Motion is estimated on a region by region basis by displacing in the reference frame a similarly shaped region of

the current frame segmentation under the motion transformation model. This is a simple extension of the block matching technique to an arbitrarily shaped region. It can easily be extended to elaborate motion models, rather than pure translational displacements. A description of a quadtree based motion estimator/segmenter is provided in Section 3.

It is interesting to notice that regions corresponding to uncovered regions in the current frame are likely to be very small as they can be predicted only by small areas of the previous frame. To avoid creating a dense segmentation of the frame to be encoded, uncovered areas could be identified as such by reverting the direction of the motion estimation process from the reference frame to the frame to be encoded (rather than from the current frame to the reference frame). Areas of the frame to be encoded that are not used in the reverted compensation process define uncovered areas in the frame to be encoded. These can be intra-coded rather than inter-coded in the quantization stage.

The third stage region-based motion compensated video coder is to generate a displaced frame difference signal $e[k,l,n]$ from the region-based motion compensated frame (k and l relate to discrete spatial coordinates while n correspond to the frame number). The advantage of this type of prediction with respect to traditional macroblock based motion compensated prediction lies in the fact that no region contains more than a single motion model. For simplicity, we assume here a simple translational model for each region, but more elaborate models could be considered (affine model, rotation and translation, ...).

On a region by region basis (e.g., region $R_i^{(n)}, (i = 1, 2, ..., N_r)$), the displaced frame difference signal $e_i[k,l,n]$ may or may not be coded according to the available bandwidth or the desired quality of the reconstructed frame. Coding the error information can be performed either using a traditional block based DCT coding approach or by selecting a set of r orthonormal basis functions $\{\psi_j^{(i)}[k,l,n], (j = 1, 2, ..., r)\}$ that can efficiently approximate the error signal [3,7] by transformation. The basis functions $\psi_j^{(i)}[k,l,n]$ can be selected as a linear combination of 2D *DCT* basis function. They are made orthonormal on a region by region basis so as to pack the energy of the transformed signal. The orthonormality constraint is equivalent to have:

$$\sum_{(k,l)\in R_i^{(n)}} \psi_j^{(i)}[k,l,n].\psi_{j'}^{(i)}[k,l,n] = \delta_{jj'} \tag{1}$$

where $\delta_{jj'}$ is the Kronecker symbol.

In the last stage coding and quantization are applied to represent the various types of information that need to be sent to a decoder for reconstruction. Two types of information must be coded in lossless fashion: 1) the segmentation map; 2) the motion information. Given the region based approach, it is reasonable to use temporal correlation with respect to previously encoded frames so as to reduce the cost of representing the segmentation map, and the motion estimates. It is suggested here (see Section 4 for a segmentation specific implementation) to use DPCM prediction along the time dimension to encode motion information, and some adaptive entropy coding to further compress the current motion segmentation map representation with respect to the previous frame motion segmentation map.

The transformed error information can be quantized after normalization by an approriately selected quantization matrix with a quantization step size that can depend on the buffer content (A uniform quantization scheme is assumed here).

The three types of coded informations together with some minimum overhead (coded/not coded decision, intra/inter decision, MC/no MC decision on a region by region basis) are finally multiplexed to form a single bit-stream that may be sent to the decoder for inverse quantization and reconstruction. All types of information utilize entropy coding to obtain a further decorrelated bit-stream.

2.3 Scene adaptive segmentation based video coding

Accordingly, we have constructed the following video coding scheme: a) a motion based segmentation algorithm so as to obtain for each frame in the sequence the moving objects and their apparent motion; b) a sophisticated coding procedure to describe the segmentation shape and the motion information associated to each region; c) from the encoded information, a motion-compensated algorithm that allows to reconstruct the video sequence.

However to gain bandwidth, we suggest to eliminate the temporal prediction loop for motion compensation prediction so as to remove the coding of error information, thus any further overhead in the context of very

low bit rate applications. This is made possible by using for motion compensated prediction a single "high quality" reference frame during a certain time interval (a few seconds), during which the scene model remains unchanged. All frames to be encoded are predicted with respect to this scene model. It is possible to imagine to have more than one reference frame, more than one model. For example, for a video-telephony application we could have received as references two frames representing the talking person in a normal situation and while smiling. A 1-bit overhead can be used then to select which reference must be used at the decoder for prediction. In a video-telephony context, a set of reference frames may be transmitted before the communication effectively starts. However if we want to make this concept more general, we can suggest to update the reference frame on a regular basis or whenever the frame being encoded becomes to unpredictable from the currently used reference frame.

On a frame by frame basis, this decision is made from a calculation of the motion-compensated prediction error energy and/or the bit-rate used to encode the current frame. When the available bit-rate is not too limited the buffer content can also be used. If the prediction error and/or the current frame bit-rate become too large, it is reasonable to assume that the current scene model is not adequate enough so that a new reference frame need be transmitted to the decoder. As the quality of such reference needs to be very high a sufficiently small quantizer step size should be selected. Accordingly, the transmission of the new reference frame may take a significant amount of time ($\sim$ 0.5s), which may turn out to be inadequate for coding future frames. Trade-offs will be studied in Section 5, accordingly. With the proposed scheme, it is possible not to freeze the reconstructed sequence while the reference is being trasmitted by splitting the channel bandwidth in two virtual channels one used to transmit the "reference frame" update, the other to obtain a decoded version of the current frame. Again, tradeoffs exist as splitting the available bandwidth can increase the time necessary to reconstruct the new "reference frame".

A final point to mention is related to the buffer fullness. In the proposed scheme, as we suggest not to code the prediction error from one frame to the next, it is essential to provide ways to reduce the output rate of the coder when the buffer tends to get full. Three elements in a region-based motion compensated predictive scheme can be adjusted for this purpose: the displacement accuracy, the segmentation accuracy and the rate at which frames are encoded. Section 4 will discuss some of these issue in detail.

The overall coding procedure of the proposed multiple reference region-based motion compensated video coder is described in the block diagram of Figure 3.

3 SEGMENTATION STRATEGY

3.1 A unique segmentation map

The segmentation process is necessary so as to base the motion compensation process on a region basis, rather than on a macroblock (16×16 block) one.

If this idea is simple in its concept, there are some very delicate issues to be addressed. In particular, sending information according to items 1, 2 and 3 in the previous section may require transmitting 3 different segmentations: The region description one, the motion description one and the error description one. These may differ significantly from each other unless specific constraints are imposed. From traditional segmentation based image coding models [13], the description of the segmentation map is very expensive, typically at least 1 bit per region boundary point. It seems therefore necessary to reduce the number of such maps to be encoded.

To guarantee a unique segmentation map for predicted frames, we suggest to

- base the segmentation solely on motion information. Only when the apparent motion between 2 or more successive frames is insufficient to define accurate region boundaries, luminance information of a single frame may be used.
- compute the segmentation for the frame to be encoded using the past or future frames as references, rather than predict the current frame by displacing regions of the future or past frames. This allows any error information after compensation to be limited within the region boundaries of the motion segmentation. An on/off flag is then sufficient to describe whether or not such error information needs to be coded for any given region of the original segmentation.

Both these selections have been made in the proposed scheme. We propose in what follows a top-down strategy followed by a bottom-up one, similar to a split-and-merge segmentation technique.

3.2 Quadtree based motion segmentation

To limit the complexity of the segmentation algorithm, and to have a structured data representation, that may further benefit from a better correlation between segmentation maps of successive frames, we embed the segmentation graph in a tree structure. This seems further attractive, as certain tree structures provide a good trade-off between region and luminance description [15]. For simplicity, we select quadtrees (but more general structures, such as BSP trees, could have been considered), and have constructed a quadtree motion segmentation algorithm, that works in a top-down fashion followed by a bottom-up one.

The top-down part of the algorithm works as following: For any given frame, a multiscale block based motion estimation is iteratively computed with respect to a previous reference frame.

Starting from the entire current frame, blocks are optimally motion-compensated using the reference frame: Whenever the block minimum motion-compensated error energy falls above a dynamic threshold, the block is split into 4 sub-blocks, and the procedure is then iterated. The dynamic threshold is varied according to the block dimension, so as to allow larger error energies for smaller block sizes. The splitting process can be terminated when the block size reaches a lower bound (typically 2×2.) The adaptation of the threshold is essential to obtain a good trade-off between noisy estimation of small size blocks and detection of small moving objects in large size blocks.

At the end of this splitting process, the segmentation result is a quadtree, with a motion description (i.e. a displacement vector) and a block-based prediction error assigned to each leaf node.

The quadtree is then scanned in a bottom-up fashion so as to successively merge 4 children into their parent node, whenever the motion information assigned to the 4 children of a given parent node are very similar. The quadtree representation of the segmentation map is not altered by this merging stage, as we have assumed that only 4 regions can be merged together only if they define the 4 children of a same parent node.

As a result, a quadtree segmentation has been reached, each leaf node in the tree defining a square region. To each leaf node, one can assign both a displacement vector information and a prediction error signal.

3.2.1 Top-down decomposition

A QCIF frame size (144×180 pixels) is embedded in $2^p \times 2^p$ lattice, with p being a positive integer, so that it is easy to construct a quadtree that represent the lattice.

By calling $Q_{k,l}^{(i)}$ the tree node of level i and position k, l (level 0 defines the root node of the quadtree, level 1 corresponds to the 4 children of the root node, and so on; k, l define the coordinates of the top left point of each block), its children are defined by the tree nodes $Q_{k,l}^{(i+1)}$, $Q_{k+2^{p-i},l}^{(i+1)}$, $Q_{k,l+2^{p-i}}^{(i+1)}$, and $Q_{k+2^{p-i},l+2^{p-i}}^{(i+1)}$.

In the top-down approach, an optimum displacement vector is estimated for each $2^{p-i} \times 2^{p-i}$ blocks corresponding to all tree nodes to be processed at a given level i. Let us call $E_{k,l}^{(i)}$ and $\mathbf{d}_{k,l}^{(i)}$, the minimum displacement difference energy and associated displacement vector of one such block at level i with respect to the reference frame, respectively. The motion estimation is iterated at level $i+1$ if

$$E_{k,l}^{(i)} > T_i = T_{i-1}/2 \tag{2}$$

with T_{p-1} being specified for 1×1 region, typically a value of 30 is accurate enough for a typical 8-bit video-conferencing sequence.

3.2.2 Bottom-up merging

In the bottom-up part of the algorithm, children nodes $Q_{k,l}^{(i+1)}$, $Q_{k+2^{p-i},l}^{(i+1)}$, $Q_{k,l+2^{p-i}}^{(i+1)}$, and $Q_{k+2^{p-i},l+2^{p-i}}^{(i+1)}$ are merged into their parent node $Q_{k,l}^{(i)}$ if $\mathbf{d}_{k,l}^{(i+1)}$, $\mathbf{d}_{k+2^{p-i},l}^{(i+1)}$, $\mathbf{d}_{k,l+2^{p-i}}^{(i+1)}$, and $\mathbf{d}_{k+2^{p-i},l+2^{p-i}}^{(i+1)}$ are within some small fraction from each other (typically ± 1 pel for each component). In case of merging, the parent node is assigned the displacement $\mathbf{d}_{k,l}^{(i)}$, which corresponds to the integer rounded value of the four children displacement vector average, i.e.

$$\begin{aligned} \mathbf{d}_{k,l}^{(i)} \;=\; & \frac{1}{4}[\mathbf{d}_{k,l}^{(i+1)} + \mathbf{d}_{k+2^{p-i},l}^{(i+1)} + \\ & \mathbf{d}_{k,l+2^{p-i}}^{(i+1)} + \mathbf{d}_{k+2^{p-i},l+2^{p-i}}^{(i+1)}] \end{aligned} \tag{3}$$

Given the randomness of the displacement estimates for small size blocks (large values of i), a better merging strategy is under investigation. This is obtained by selecting for each set of 4 children of a given parent, one of the 4 displacement vectors found for each one of them and use it to displace by the corresponding amount their parent node. If the new prediction error energy does not increase significantly, the 4 children are merged into one. The problem lies in selecting in an automatic fashion the value of the increase in prediction error energy with respect to the optimal one.

If one does not mind departing from a quadtree segmentation map of the final segmentation result, a merging strategy can be considered. Alternatively a reduced complexity strategy that proposes an original ordering of the dissimilarity measures between neighboring size regions was recently proposed by Ebrahimi [6]. We have not considered these approaches as we want to represent the segmentation map and associated motion information for each non reference frame using a quadtree addressing, as exposed in the next section.

For small size blocks, the motion estimate may not relate at all to a physical displacement. In fact, small blocks may correspond to uncovered areas in the current frame. The displacement vector allows then by compensation to predict the best match of a block of the reference frame within a certain spatial range of a same size block in the current frame. The set of all possible blocks of same size from the reference frame can be seen as a locally adapted dictionary of a vector quantizer, the displaced blocks defining codewords of the dictionary. This can lead to the speculation that even though uncovered areas of the current frame are not present in the reference frame, there still are codewords of the dictionary that are good matches for small size blocks. This way, it is unlikely that the non-transmission of error information (to reconstruct non reference frame) leads to perceptually unacceptable errors.

3.3 Segmentation results

In our simulation, the search range was kept the same for each block size, and was set to ± 7 pixel, with an integer pel accuracy. The threshold on the displaced block difference energy was set at 30 for a region size corresponding to one pel in the splitting stage of the top-down quadtree construction process. The last level of the decomposition process in that part of the segmentation algorithm was limited to blocks of size 2×2.

Figure 5 presents the final segmentation result, with corresponding motion field, and displaced frame difference, for frames 75 and 80 of the sequence "Miss America" shown in Figure 4.

4 CODING STRATEGY

A multiscale spatio-temporal coding algorithm has been designed to describe the motion information jointly with the quadtree segmentation map. While describing the quadtree structure, a spatio-temporal prediction of the motion information is performed with very little overhead.

In the event there is enough bandwidth to encode some of the prediction error information, a further merging stage is applied to the quadtree segmentation to obtain the segmentation error map. In the segmentation error map, regions are formed by merging all contiguous squares exhibiting the same displacement value. This information can be computed independently at the transmitter's and the receiver's end.

Each relabelled region for which the error signal is too high can then be encoded in a region based fashion, using a set of basis functions that is made orthonormal with respect to the support of the region [4,7], as described in Section 2 (See Figure 1 and Figure 3).

In what follows, we target our simulation results to work at very low bit-rates so that no error information can be transmitted at the receiver. Only motion and shape information will be provided to the receiver with respect to the single reference frame. Contrarily to what has been described in Figure 3, given the early stage of our simulation, any update of the reference frame has not been made using motion-compensated prediction. Rather, each new reference frame is provided in intra-frame coding mode.

4.1 Intra-frame coding model

At the beginning of the transmission and after each scene cut, a new reference frame must be generated. Any still picture coding strategy could have been used in this context, e.g. JPEG [16].

Alternatively a spatial segmentation coding technique could have been selected. This has the advantage of keeping a similar coding strategy to the one used for inter-frame prediction.

Accordingly, we used a quadtree based coding technique similar to the one shown in [12]. The segmentation map coding is similar to what is presented in the next sub-section. Luminance and chrominance information of the quadtree leaf nodes are spatially predicted from their neighbours. The prediction strategy is equivalent to the "s" prediction mode used for motion prediction in the next section.

Figure 1b shows the reconstructed picture quality for frame 0 of the "grandma" QCIF sequence, using roughly 4000 bits.

4.2 Spatio-temporal prediction of motion information

To encode the quadtree segmentation map with associated motion information, the tree is traversed in a top-down fashion. At each level, all nodes are successively analyzed, so as to specify whether they have descendants or they define leaf nodes. In the former case ("the traversed node is a parent node with 4 children"), a single bit set to 0 is transmitted. In the latter case ("the traversed node is a leaf of the quadtree") a bit set to 1 is transmitted.

If 0 has been transmitted, the 4 children of the current node are processed when the next level of the tree is reached. If bit set to 1 has been transmitted, the motion information assigned to the corresponding leaf node is encoded. For this purpose, we select a spatio-temporal prediction mode for estimating the motion information. The corresponding mode is one element of the 3 symbol alphabet: $\{s, t, T\}$.

- If "T" is selected, the motion vector can be predicted at no additional cost as the average of the previously encoded frame motion vectors that have been temporally extrapolated to the current frame and fall within the block corresponding to the leaf node being currently encoded (assuming constant velocity of the moving objects).

  Consider the segmentation S_1 of a previously transmitted frame F_1 which occurred at time t_1. Consider as well the instant t_0 in which the reference frame F_0 occurred. Each region in segmentation S_1 has an assigned motion vector $\mathbf{d}_{i,1}$, which measures the displacement of a particular region $R_{i,1}$ from the reference frame F_0 to the previously transmitted frame F_1. Let us call the current frame F_2, the time at which it occurred t_2, its segmentation S_2. The current motion estimate $\tilde{\mathbf{d}}_{j,2}$ of a region $R_{j,2}$ is given by

$$\tilde{\mathbf{d}}_{j,2} = \frac{t_2 - t_0}{(t_1 - t_0) \sum_{i \in C_{i,j}} N_{i,j}} \sum_{i \in C_{i,j}} N_{i,j} \mathbf{d}_{i,1} \tag{4}$$

  where $N_{i,j}$ corresponds to the number of pixels of all regions $R_{i,1}$ which are displaced into region $R_{j,2}$ by reverting and scaling the displacement vectors $\mathbf{d}_{i,1}$ of each pixel in $R_{i,1}$, so that each such pixel is displaced into frame F_2 by an amount $-\mathbf{d}_{i,1}\frac{t_2-t_1}{t_1-t_0}$. To determine which regions $R_{i,1}$ fall after compensation into $R_{j,2}$, a compensated version of S_1 is generated using the $-\mathbf{d}_{i,1}\frac{t_2-t_1}{t_1-t_0}$ displacement estimates. The intersection of these displaced regions into F_2 defines the set of indices i that belong to $C_{i,j}$, i.e. all regions $R_{i,1}$ that are moved into regions $R_{j,2}$ after compensation.

  As we assume that this prediction mode is most frequently used, a single bit set to 0 is used to code it.

- If "t" is selected, the motion vector is still temporally predicted, but there is a residual error with respect to the exact motion vector. Coding of the difference value with respect to the temporal motion estimate $\tilde{\mathbf{d}}_{j,2}$ is further performed component-wise, using Huffmann tables provided for recommendation H.261.

  The selection of the "t" mode with respect to the "s" mode is done when the temporal prediction results in a better estimate with respect to the spatial prediction provided by the "s" mode.

  Coding of the "t" prediction mode is performed using a 10 2-bit word.

- If "s" is selected the current displacement $\mathbf{d}_{2,j}$ is spatially predicted from the displacement of the largest leaf node which was contiguous in space to the current node. If there are 2 or more leaf nodes corresponding to the largest neighboring nodes of the currently processed node, the first one encountered in the tree traversal procedure is used. To guarantee that all nodes of a given tree can be used for prediction the tree traversal algorithm scans each node one level at the time from the root node to the current level.

The use of the largest neighboring node for spatial prediction of motion may not be the optimal one, but it is reasonable (spatial prediction may work better when performed from a larger block than from a smaller one) and can be used by the decoder at no additional cost. The difference value with respect to the current block displacement value is further encoded component-wise using again the Huffmann tables provided in recommendation H.261.

Coding of the "s" prediction mode is performed using a 11 2-bit word.

The selection of the "T" or "t" modes is considered only for blocks larger than a certain size (typically the smallest considered size), and is used to handle the constant velocity of most objects over time. As small size blocks may have unreliable motion estimates, it is unlikely that a temporal prediction may work in this case. In such a case, only a spatial prediction is made, with no need therefore to transmit the "T", "t" or "s" symbols (1 or 2 bit overhead).

Performance of this coding approach is discussed in the next section.

5 SIMULATION RESULTS

Using few high quality reference frames (one every 2 seconds), QCIF video-telephony sequences have been encoded at bit-rates ranging from 10kbit/s to 48kbit/s with a fixed frame rate ranging from 4 to 10 frames/s. For a given bit-rate the frame rate was adjusted so as to provide acceptable quality results. (The cost for encoding the high quality reference frames has been neglected at this point). In the segmentation process, the error threshold on the absolute frame difference energy was varied from 30 to 100 for a 1×1 block size. By setting this threshold, one obtains a trade-off in the number of regions of the final segmentation and the quality of the motion compensated prediction.

To prove the effect of error propagation on P frames when no error coding is added to the motion compensated error signal, we present in Figure 6 a set of motion-compensated error images for the sequence "carphone". The splitting stage of the top-down part of the segmentation algorithm was limited to 8×8 blocks. As it can be noted in the reconstructed image, all detailed information has been lost.

In the test of Figure 6, it was also noticed that even worse artefacts occur in the chrominance domain, when no error information is added to correct the motion compensated prediction. This is normal considering that in the experiment motion information was detected only on the basis of luminance information.

If a single reference frame is used instead, the prediction becomes much more accurate as long as the motion model can handle the important displacements that can occur between 2 frames separated by a large time interval. Unfortunately, in the current experimental result, the motion was limited to pure translation within a range of ± 15 pixels. Figure 7 shows the problem for some frames of the "Miss America" sequence. A way to solve the problem is to start the search for any new frame around the previous reconstructed frame motion estimate. By selecting the search range differentially with respect to the previous frame motion estimate, it is reasonable to assume that the extent of differential displacement will remain more or less within range of ± 15 pixels.

In this case instead, there is a much better compensation of the chrominance images as no error feedbacks in the prediction loop. The problems that have been presented for Figure 6 can be noticeably reduced by decreasing the size of the smallest size block in the splitting stage of the motion segmentation process.

Table 1 shows the bit-rate versus frame-rate that can be obtained having an acceptable quality of the decoded sequence for the "carphone" and "Miss America"sequences (the bit-rate associated with the coding of the intra-frame was discarded in the experiment. The intraframe was assumed to have been coded in a lossless fashion. No motion compensated prediction error information was encoded but alternatively we allowed for a smaller size of each block in the decomposition stage of the segmentation). For results around 10kbit/s for the "carphone" sequence, our approach seems to lead to a better reconstruction quality than that provided by the test model TMN2 provided in the ITU core experiments for low bit rate video. At higher bit rates, better performance are achieved by TMN2. For the "Miss America" sequence, similar performance has been obtained.

6 CONCLUSION AND FUTURE DIRECTIONS

Sequence	Frame (Hz)	Bitrate (bps)	Pred.	Quality
Miss	10	30k	P	perf. YCrCb
Miss	7.5	16k	P	perf. Y
carph.	5	8k	P	satisf. Y
carph.	4	25k	m.ref.	good YCrCb

Table 1: Coding performance summary

We have described a quadtree based motion predictive scheme to encode video sequences at low bit-rates. The novelty of the approach lies both in the joint motion estimation and segmentation procedure, and in the joint coding of the motion information and segmentation map. We have suggested to predict spatially or temporally the motion information for each square region in the quadtree using neighboring size blocks or the average value of motion information extrapolated from previously encoded frames. A rigorous strategy is adopted to obtain the proper motion estimate at no additional cost in terms of overhead.

Extensions of this encoding scheme are considered so as to include error information when necessary, especially to update any new reference frame. This requires to design a sophisticated buffer control strategy as described in section 5 that allows to share the channel bandwidth between an update of the reference frame and a prediction of the currently being encoded frame. It is also necessary in this context to define the conditions under which one must tell the decoder that the next frame may be used later as a temporal reference.

The use of a set of multiple reference frames remains still under investigation, to judge whether it provides a real advantage over traditional prediction. In this context, it is necessary to start the motion estimation and description from the previously decoded motion field (i.e., the motion field sent for a previous frame.) Also, only preliminary experiments were developped to study the effect of the splitting of the channel bandwitdh (to provide for an update of the reference frame).

The use of time correlation is being investigated as well to predict the change of the segmentation map (using the quadtree representation) over time.

We think that a two stage motion compensation process should be added to the presented scheme: a global motion compensation that can better handle zoom and pan motion of the camera; a local motion compensation of the object displacement that can possibly include rotation (to handle tilt of objects in a scene).

Finally, in our view it is clear than region-based video coding cannot provide significant improvement over traditional block based techniques unless may be the shape of objects is tracked in time. In the current system, we tried to handle this only in the coding stage (especially for motion information). It is necessary to have it also in the definition of the segmentation of each frame (i.e. use the previous frame segmentation as an initial solution or even a constraint to the current frame segmentation result).

7 REFERENCES

[1] *Video Codec Test Model, TMN2*, CCITT SGXV WPXV/1: Expert's Group on Very Low Bitrate Visual Telephony, 1994.

[2] K. Aizawa, "Model Based Coding", to be published in the *Handbook of Visual Communications*, Ed.s H.-M. Hang and J. Woods.

[3] H. H. Chen, C. Horne, and B. G. Haskell, "A Region-Based Approach to Very Low Bitrate Video Coding", AT&T Technical Memorandum BL0113550-931207-TM, 24 pages, 1993.

[4] H. H. Chen, M. R. Civanlar, and B. G. Haskell, "A block transform coder for arbitrarily-shaped image segments", in *Proc. of the ICIP94 Conference, Austin, TX*, I:85-89, Nov. 1994.

[5] E. Dubois, and J. Konrad, "Estimation of 2-D Motion Fields from Image Sequences with Application to Motion-Compensated Processing", in *Motion Analysis and Image Sequence Processing, M. I. Sezan, and R.L. Lagendijk*, Kluwer Academic Publ., 3:53-87, 1993.

[6] T. Ebrahimi, "A New Technique for Motion Field Segmentation for Very Low Bitrate Video Coding Applications", in *Proc. of the 1994 ICIP Conference, Austin, TX*, II:433-437, Nov. 1994.

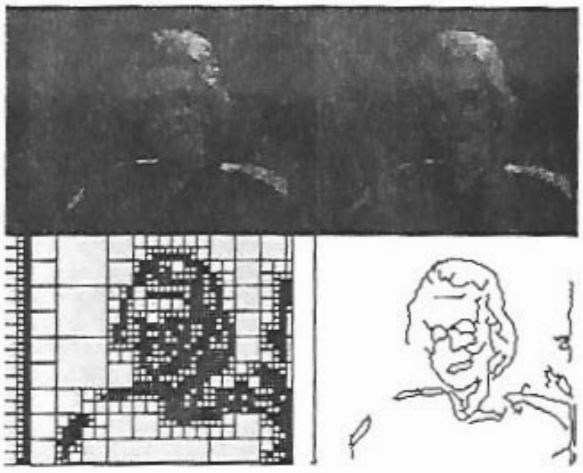

Figure 1: Quadtree based coding of the first frame of the "grandma" sequence; a) original frame; b) coded reconstruction (~ 4000bits); c) segmentation shape; d) contours of the original frame.

[7] M. Gilge, T. Engelhardt, and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform", *Signal Processing: Image Communication*, 1(2): 153-180, Oct. 1989.

[8] C. Gu and M. Kunt, "Contour Image Sequence Coding by Motion Compensation and Morphological Filters", in *Proc. of the VLBV94 workshop, University of Essex, Colchester (UK)*, paper 7.1, Apr. 1994.

[9] M. Hotter, "Object Oriented Analysis Synthesis Coding Based on Moving Two Dimensional Objects", *Signal Processing: Image Communication*, 2:409-428, 1990.

[10] M. Kunt, A. Ikonomopoulos, & M. Kocher: "Second-Generation Image-Coding Techniques", *Proceedings of the IEEE*, 73(4):549-574, Apr. 1985.

[11] A.E. Jacquin, "Image Coding Based in a Fractal Theory of Iterated Contractive Image Transformations", *IEEE Transactions on Image Processing*, 1(1):18-30, Jan. 1992.

[12] R. Leonardi, "Segmentation adaptative pour le codage d'images", Ph.D. Thesis No. 691, Dept. of Electrical Engineering, Swiss Federal Institute of Technology, Lausanne, Switzerland, June 1987.

[13] R. Leonardi, M. Eden, and M. Kocher: *Coding a Contour Graph with No Address Assignments*, AT&T Bell Laboratories Technical Memorandum, Doc. No. 11355-901115-12TM, Nov. 1990.

[14] R. Leonardi, and H. Chen, "Tree Based Motion Compensated Video Coding", in *Proc. of the 1994 ICIP Conference, Austin, TX*, II:438-442, Nov. 1994.

[15] J. Ostermann, "SIMOC: A European Initiative towards MPEG-4 by COST 211", ISO/IEC JTC1/SC29/WG11, MPEG-4 seminar, Paris, Mar. 1994.

[16], W.B. Pennebaker, and J. L. Mitchell: "JPEG: Still Image Data Compression Standard", *Van Nostrand Reinhold*, 1993.

[17] H. Radha, R. Leonardi, and M. Vetterli: "Coding Images Using BSP Trees", submitted to *IEEE Transactions on Image Processing*.

Figure 4: Original "Miss America" sequence: a) frame 75; b) frame 80.

Figure 5: Final segmentation result (On the left: Displaced frame difference; on the right: horizontal and vertical components of the motion field, the needle diagram, and the final segmentation map).

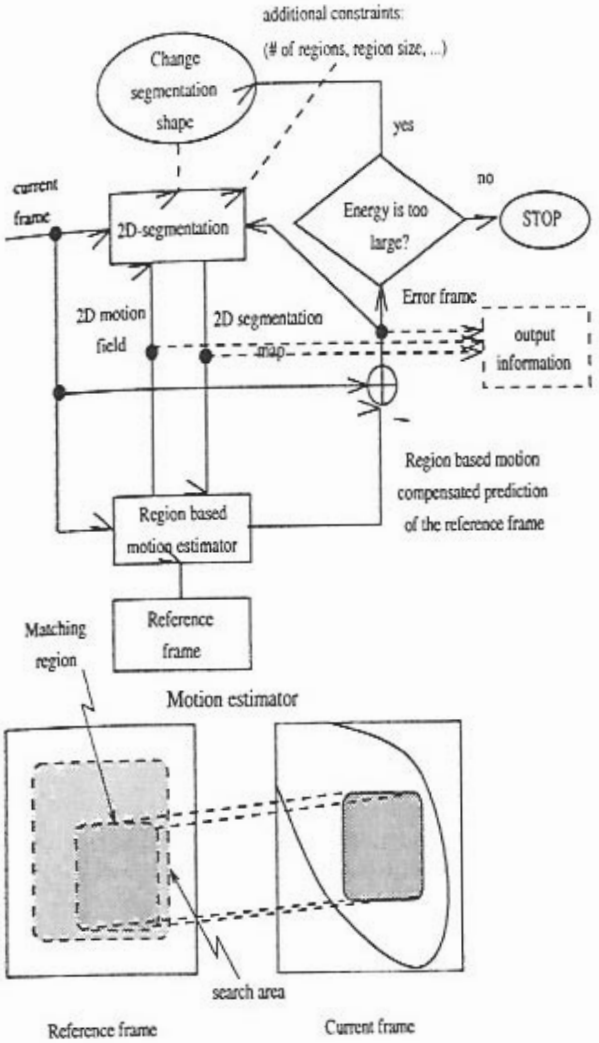

Figure 2: Motion segmentation unit.

Figure 6: "Carphone" sequence coded normal prediction (P frames); smallest block size: 8x8, maximum displacement: +/- 7, error threshold: 30, bit-rate: 12000 bps, frame rate: 4Hz; Horizontally: a) original frame; b) decoded frame; c) error frame - Vertically: 1) frame 54; 2) frame 84; 3) frame 96.

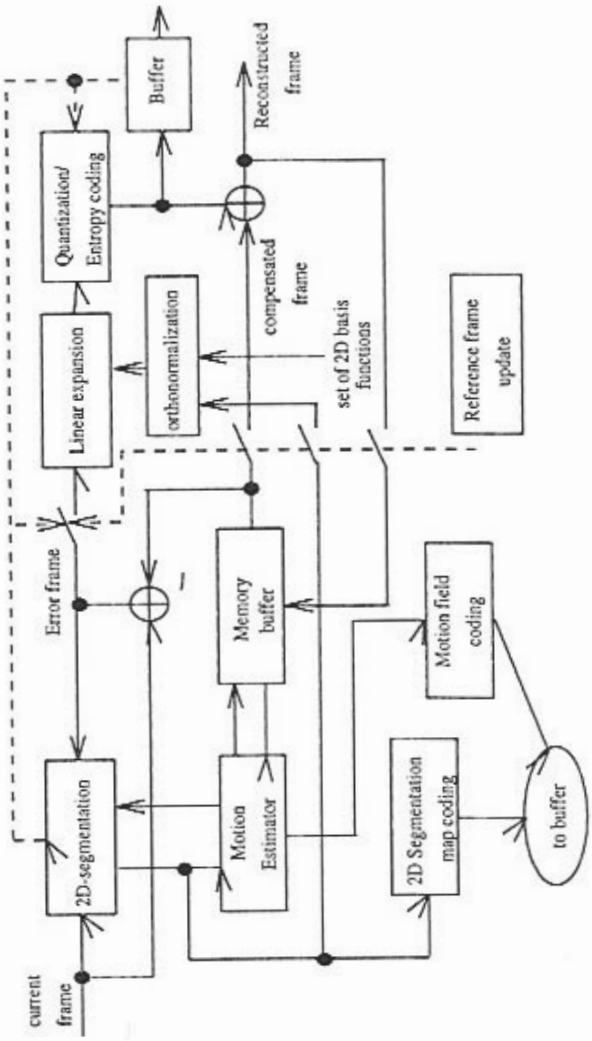

Figure 3: Scene adaptive region-based motion compensated video coding algorithm.

Figure 7: "Miss America" sequence coded using as single reference frame frame 0, smallest block size: 4x4, maximum displacement +/- 7, error threshold: 30, bit-rate: 30000 bps, frame rate: 10Hz; Horizontally: a) frame 61; b) frame 91; c) frame 121- Vertically: 1) original frame; 2) decoded frame.