INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO INFORMATION

ISO/IEC JTC1/SC29/WG11
MPEG 95/359
November 1995

Source: **RACE/MAVT, University of Hannover, Robert Bosch GmbH, and Deutsche Telekom AG**

Authors: **Peter Gerken, Michael Wollborn, Stefan Schultz (University of Hannover)**

Title: **An object–based layered codec as proposal for MPEG–4 scalability and compression tests – Technical description**

Abstract

An object–based layered codec is submitted for formal testing of object scalability and improved coding efficiency at very low bit rates using class–A sequences. The codec consists of 3 layers. In layer 1, only colour parameters are transmitted. In layer 2, additionally motion parameters are transmitted. In layer 3, additionally to colour and motion parameters shape parameters are transmitted. In order to achieve the highest possible coding efficiency in compression applications, always the most efficient layer is selected according to appropriate switching criteria. In scalability applications, only layer 3 is used. As layers 1 and 2, TMN 5 has been incorporated with its intra– and interframe modes. In layer 3, object–based analysis–synthesis coding as derived from [7] is performed. It consists of 5 main components: image analysis, parameter coding, parameter decoding, image synthesis, and a memory for the object parameters. By the image analysis, the images of a sequence are decomposed into arbitrarily shaped moving objects each described by three sets of parameters defining the shape, the motion, and the colour of the object. The colour parameters consist of the luminance and chrominance values of the object surface. Goal of the approach implemented as layer 3 is to synthesize an image area as large as possible from colour parameters transmitted with previous images using only shape and motion parameters of the current image. Coding of colour parameters is concentrated on a small area, thus the available number of bits per coded picture element is relatively high guaranteeing a good quality in these areas. By the incorporation of TMN 5 as layers 1 and 2, it is guaranteed that the complete coder can not perform less efficiently than TMN 5.

Part of the work was performed within the scope of the European COST 211ter project.

1. Introduction

This paper gives a technical description of the proposal submitted by University of Hannover for the first round of tests in MPEG–4. Proponents are RACE/MAVT, University of Hannover, Robert Bosch GmbH, and Deutsche Telekom. For these first tests the proposal addresses two functionalities, object–based scalability and improved coding efficiency. The technical description is valid for **both** functionalities. For a better understanding of the coding technique, the modules of the **en**coder are described in more detail, the corresponding **de**coder modules are considered to only invert the encoding procedure unless something different is written.

2. Structure of the codec

Fig. 1 shows a simplified block diagram of the proposed codec. The coding scheme has been derived from a general transaction [13], which describes the principle of a layered video coding scheme, and is its first implementation. In this implementation, coder and decoder consist of 3 layers each.

In the first layer, only colour parameters are transmitted. Intraframe transform coding of the incoming image is performed. The quantized transform coefficients of the original samples of the luminance component and of the two chrominance components build the colour parameters.

In the second layer, additionally to colour parameters motion parameters are transmitted. Displacements are estimated block by block and transmitted as motion parameters. Displacement compensating interframe prediction is performed, and transform coding is applied to the prediction errors. The quantized transform coefficients of the prediction errors build the colour parameters.

Today's block–based video coding standards [4][9][10][11] consist of these two layers.

In the third layer, additionally to colour and motion parameters shape parameters are transmitted. Object–based analysis–synthesis coding derived from [7] is applied. Silhouettes of objects, which may be arbitrarily shaped, are estimated and their approximated boundaries are transmitted as shape parameters. Displacements are estimated, but considering the object shapes, and transmitted as motion parameters. After image synthesis with these shape and motion parameters, image areas with significant synthesis error are detected, their total size being much smaller than the size of the whole changed image area. The approximations of their arbitrary boundaries are also transmitted as shape parameters. Transform coding is applied to the synthesis errors only in these areas considering their arbitrary shapes, rather than in the whole changed image area block by block. Quantized transform coefficients of the synthesis errors build the colour parameters.

In compression applications, a layer selector selects the most efficient layer according to appropriate switching criteria. In scalability applications, only layer 3 is used.

Pre– and postprocessing prepares the image signal for coding and display, respectively. Only temporal and spatial resolution is changed for both purposes.

In the following sections, the modules the proposed algorithm consists of are described in more detail.

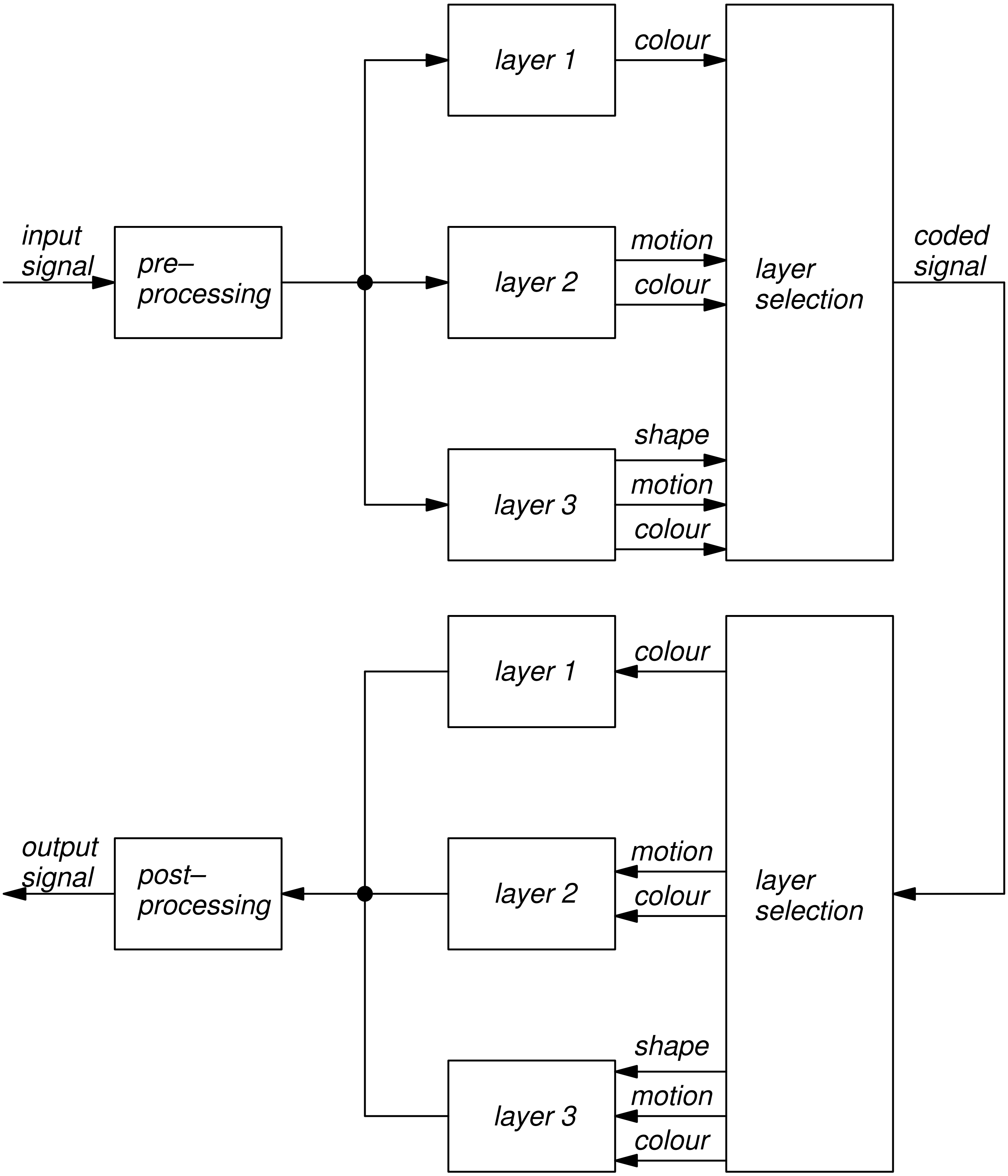

Fig. 1: Simplified block diagram of the proposed object–based layered codec.

3. Pre– and postprocessing

3.1 Conversion from library format to coding format

Temporal resolution is reduced by selecting the frames to be coded. No temporal filtering accompanies the frame rate reduction. Temporal resolution is fixed to the values given in Table 1. This means that these for very low bit–rate applications relatively high values are guaranteed over the whole sequence and no annoying ups and downs of the temporal resolution appears. In the current implementation, these values are set by hand before simulation starts. Automatical calculation of the appropriate temporal resolution of a scene is planned to be implemented in the future.

	10 kbit/s	24 kbit/s	48 kbit/s
Akiyo	10 Hz	10 Hz	15 Hz
Mother & Daughter	7.5 Hz	10 Hz	10 Hz
Hall monitor	7.5 Hz	10 Hz	10 Hz
Container ship	7.5 Hz	10 Hz	10 Hz
Sean	–	–	10 Hz

Table 1: Temporal resolution (frame rate) of each sequence for each bit rate

Spatial resolution is reduced by sampling frequency reduction. After selecting a centered window of 704x480 picture elements out of the full CCIR 601 4:2:2 format by omitting picture elements at the frame boundaries, the signal within this window is low–pass filtered and subsampled to QCIF picture format in one step. The coefficients of the lowpass filter are chosen depending on the relation of the input image size and the output image size so that an ideal low–pass filter is best approximated. For this, an si–function is used and Hamming–windowed. For details please refer to [6][14]. QCIF picture format was used for all sequences at all bit–rates. For Akiyo at 48 kbit/s, CIF picture format with 10 Hz shows even a better result than the submitted one. However, since for time reasons the decoder could have still not been verified in CIF mode, the simulation with QCIF picture format and 15 Hz temporal resolution was preferred for submission.

3.2 Conversion from coding format to display format

Temporal resolution is enhanced from the values in Table 1 to 30 Hz by frame repetition. No other kind of temporal filtering is applied.

Spatial resolution is enhanced by upsampling and low–pass filtering. First, vertical resolution of the chrominance components is doubled. Then, vertical and horizontal resolutions of all components are doubled. Finally, a mid–level grey border is put around the picture to form a full CCIR 601 4:2:2 image with the decoded and upsampled picture in its center. For all filter operations, the filter proposed by COST 211ter is used [5]:

$$(-1\ \ 0\ \ 9\ \ 16\ \ 9\ \ 0\ \ -1)\ /\ 16$$

4. Layers 1 and 2

As layers 1 and 2, TMN 5 has been used. TMN 5 intraframe modus is layer 1. TMN 5 MC–interframe modus is layer 2. By this choice it is guaranteed that in cases where the third layer performs worse the image quality cannot fall below what is provided by the existing standard, H.263 [11]. Furthermore, compatibility to existing equipment is guaranteed, and scalability with respect to the hardware complexity of the receiver is introduced. As decoder the public domain software of Telenor, Norway, is used.

5. Layer 3

Fig. 2 shows a block diagram of layer 3. The incoming images are decomposed by an object–based image analysis into arbitrarily shaped moving objects. Each object is described by three sets of parameters defining its shape, motion, and colour. Colour parameters are the luminance and chrominance values of the object surface. They are forwarded via the parameter coding unit to the layer selector. Decoded parameters are stored in a parameter memory from where they are used for predictive coding of the parameters of the successice image and for the synthesis of the current image. The synthesized image is then displayed and used for the analysis of the successive incoming image. Image analysis is based on a source model. Here, the source model of flexible 2–dimensional objects which translationally move in the image plane is applied.

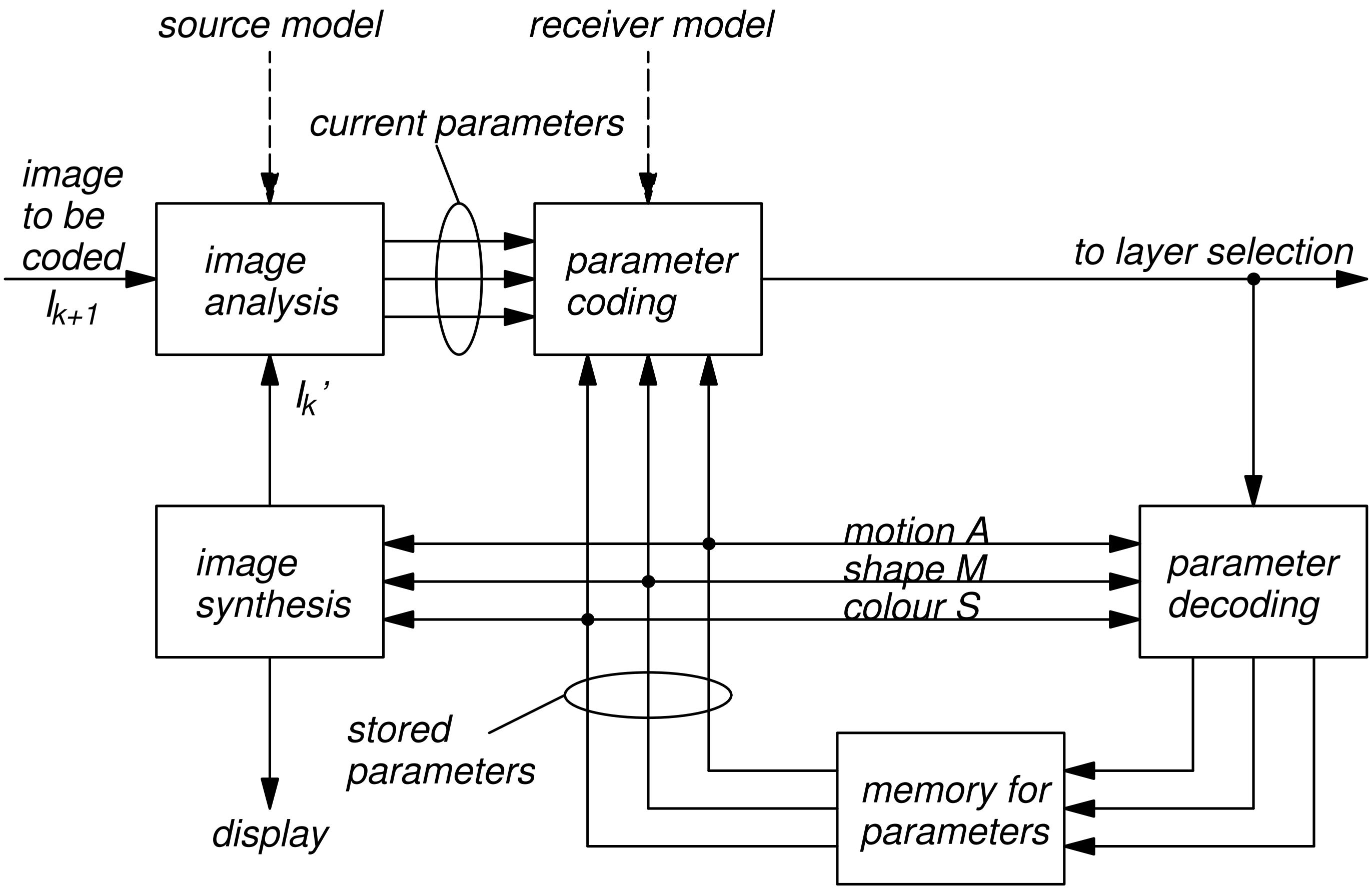

Fig. 2: Block diagram of layer 3.

5.1 Image analysis

Image analysis consists of four steps. Shapes of moving objects are to be analyzed by means of segmentation, and motion of the objects must be analyzed by means of motion estimation. After an intermediate image synthesis, regions where this synthesis does not lead to a sufficient quality are still to be detected. These regions are called model failure (MF–) regions because they do not comply with the source model upon which the image analysis is based.

5.1.1 Shape analysis

Shape analysis is completely **automatical**. It is applied to the images in coding format. Thus, this approach is also applicable in low–cost environments when a cheap camera does not provide full CCIR 601 4:2:2 quality.

According to the applied object model of 2–dimensional objects, shapes of the objects are given by their silhouettes. Since the shapes of **moving** objects are looked for, static objects e.g. drawn on a wall–paper shall not be segmented. Segmentation is applied to the difference between incoming image and a specially synthesized image where decoded values are substituted by original input values. It consists of three modules: change detection, adaptation to spatial gradients, and removal of uncovered background.

Change detection calculates a binary change detection mask from the multi–level difference image. First, a threshold operation is applied to the difference image using a spatially varying threshold according to the approach in [1][2]. In order to isolate regions, dilation and erosion are applied to the so obtained mask. For smoothing, isolated regions with a size smaller than a threshold are eliminated. In order to get temporally consistent object masks, a memory for the moving objects is applied. Each pel which has been once determined to belong to a moving object is marked in the memory. It is deleted from the memory when it does not belong to a moving object for a certain number of successive frames.

Adaptation to spatial gradients is performed after applying a Sobel operator to the incoming and the last synthesized image. Boundaries between changed and unchanged regions in the change detection mask are shifted towards the closest of the highest local gradients in one of the images. By this approach, the boundaries become more accurate. In order to remove sharp corners, which may still remain, median filtering is applied.

Removal of uncovered background is performed after motion analysis, which is explained in the following subsection, has finished. Estimated motion parameters are exploited to detect which parts of the adapted change detection mask coincides with the motion compensated position of the moving object in the previous image. These parts are deemed to belong to the position of the moving objects also in the current image. Other parts are deemed to belong to parts of the background which are uncovered due to the motion of the objects and are removed from the changed area, thus generating a moving area mask.

For the first image, an artificial moving area mask is generated. In order to avoid reading a hand–generated mask from outside, automatical segmentation is applied as described. Since for the tests the simulations had to start at frame 0 and this kind of segmentation needs at least one previous image, the first 10 images of a sequence have been mirrored to get a mask for frame 0.

5.1.2 Motion analysis

Since displacements shall be estimated which represent the physically true motion rather than only minimizing a prediction error criterion, hierarchical block matching [3][7] is applied. Searching is performed in 3 hierarchy levels starting with a large measurement window and low–pass filtered versions of the current original and the specially synthesized image as used for shape analysis. Rough but with respect to the true physical motion reliable displacements are estimated by logarithmic search in this first level. In the second level, these displacements are refined using a smaller measurement window and non–filtered versions of the involved images. The third level provides half–pel accuracy of the displacement vector field. Displacement estimation is performed on the luminance components only. The precise algorithmic parameters are listed in Table 2.

For low–pass filtering in the first hierarchy level, a simple mean value filter with a block size of 3x3 picture elements is used. The missing luminance values of non–grid positions for the half–pel accurate search in the third hierarchy level are calculated by bilinear interpolation. Only luminance values inside one object are exploited.

Hierarchy level	1	2	3
Max. update displacement	+/– 3 pel	+/– 1 pel	+/– 0.5 pel
Measurement window size	32x32 pel	16x16 pel	16x16 pel
Signal bandwidth reduction factor	1/2	1/1	1/1
Spatial vector field resolution	16x16 pel	16x16 pel	16x16 pel

Table 2: Adjustment of the algorithmic parameters for hierarchical block matching.

5.1.3 Intermediate synthesis

The estimated shape and motion parameters are forwarded to the parameter coding unit. Decoded parameters can be used in both coder and decoder for an intermediate synthesis. Intermediate synthesis is done by taking the colour parameters from the last synthesized image, which is stored in a frame memory, considering the decoded shape and motion parameters. Displacements for positions which do not belong to the vector field grid are calculated by bilinear interpolation.

5.1.4 Model failure detection

A synthesis error image is calculated by subtracting the image obtained after intermediate synthesis from the current original one. A method similar to change detection is applied but with a fixed threshold and without memory and adaptation to spatial gradients. The value of the threshold is derived from the target quality value which is input by hand before the simulation in order to arrive at a target bit rate. This value is constant for a whole simulation and also used for quantizer adjustment in colour parameter coding. Automatical calculation of the target quality value frame by frame is planned as future implementation.

5.2 Parameter coding/decoding

5.2.1 Shape parameter coding/decoding

Each connected region in the moving area mask is regarded to be a moving object. The boundaries of the moving objects and the MF–regions are transmitted as shape information. In order to reduce the amount of bits necessary for coding the shape parameters, shape approximation is performed. Combined polygon/spline representation of the shapes is applied. For details of this tool the reader is referred to the tool description [8] which has been submitted jointly with this algorithmic description. For moving objects, maximum deviations of the approximation has been set to 2.9 picture elements to the outside of an object and 1.45 picture elements to its inside. It has been found to be a good compromise between bit rate necessary for shape parameter coding and synthesis quality near object boundaries. For MF–regions, the maximum deviation has been fixed to 10 picture elements. This values is set relatively high, in order to keep the bit rate necessary for shape parameter coding low, but not infinite, in order to cope also with "pathologically" shaped regions.

The approximated shapes are characterized by the vertices of the approximation and the approximation function, polygon piece or cubic B–spline piece, selected between each pair of vertices. The vertices are represented by their coordinates, the approximation function by a binary value. For moving object shapes, it is first tested whether motion compensating interframe prediction can be applied. Displacements valid for the previous image at all vertex positions are applied for linear extrapolation of motion. It is looked whether the so predicted vertices fall into the tolerance range of the exact shapes in the current image. If not, they are rejected from the approximation. If more than two vertices are maintained, this approximation is used as initialization for the refinement process described in [8]. In this case, four ensembles must be coded:

- the runlengths of maintained and rejected vertices,
- the runlengths of maintained vertices and vertices inserted during the refinement process,
- the coordinates of the inserted vertices relative to the respectively last maintained or inserted vertex,
- the runlengths of the approximation functions.

In case prediction is not possible, the following two ensembles must be coded instead:

- the coordinates of all vertices relative to the respectively last vertex,
- the runlengths of the approximation functions.

The latter also holds for shapes of MF–regions.

All ensembles undergo a 2–dimensional adaptive arithmetic coding. Coding starts with an equal probability distribution.

After arithmetic decoding, shapes are reconstructed from their parameters. Filling the regions inside the shapes yields approximated moving area and MF–regions mask which are then used to control decoding of the other parameter sets.

5.2.2 Motion parameter coding/decoding

For each moving object, all displacement vectors inside the object and the first ones outside are coded. Intraframe DPCM is applied using the left neighbour as prediction. Zero components are collected to runlengths appended by the next non–zero value. Thus, the same 2–dimensional adaptive arithmetic coding/decoding procedure as for shape parameters can be applied.

5.2.3 Colour parameter coding/decoding

MF–regions

Shape adaptive DCT [12] of the synthesis errors is used for colour parameter coding in MF–regions. In cases where a full block of 8x8 picture elements falls into an MF–region, the approach switches back to conventional block–based DCT. Since block–based DCT is also used in layers 1 and 2, only one colour parameter coding algorithm must be implemented later in an eventual hardware solution, thus keeping the complexity reduced. Uniform quantization is applied. Quantizer stepsize is input by hand and is constant for a whole sequence. Automatic quantizer adaptation frame by frame is planned as future implementation. Zero values are collected to runlengths appended by the next non–zero value. Thus, the same 2–dimensional adaptive arithmetic coding/decoding procedure as for shape parameters can be applied again.

Uncovered background regions

Adaptive prediction of colour parameters is performed in uncovered background regions. Intraframe prediction values are obtained from the neighbouring static background. At horizontal or vertical boundaries between static and uncovered background, luminance and chrominance values of the static background are mirrored. At diagonal boundaries, horizontally and vertically mirrored values are averaged. Prediction errors are calculated. The same coding technique as in MF–regions is applied to the prediction errors. If picture elements which belong to the uncovered background were already visible before in the sequence, the decoded colour parameters are available from a special background memory for interframe prediction. If both kinds of prediction are available, the mean absolute prediction errors of the image are calculated for both kinds of prediction. That kind with the lower mean absolute prediction error is selected. This information is transmitted with 1 bit per image.

5.3 Image synthesis

After all parameters have been decoded, final image synthesis can be performed. For this, the result of the intermediate synthesis is taken and the decoded colour parameters are inserted at the appropriate positions, in the uncovered background after prediction.

6. Layer selection

In order to achieve a high coding efficiency, always the most efficient layer is selected. Since layers 1 and 2 are realized in form of TMN 5, the selection between these layers is performed blockwise. Therefore, block by block that layer out of these two is chosen which yields the lower mean prediction error, taking into account the bitrate needed for motion vector coding in layer 2. The performance of the selection so far is then compared with the performance of layer 3. For this decision, a new criterion was developed. Since shape parameter coding is not efficient for less than a whole frame, the final selection is performed framewise.

Normally, the easiest decision would be to keep the bitrate the same and to select according to the highest resulting image quality, or vice versa to keep the quality the same and to select according to the lowest resulting bit rate. But due to the different control of the incorporated coding techniques, in general they do not have exactly the same bit rate or the same quality. Therefore, in a rate/distortion (RD–) sense, there are two possibilities, as shown in Fig. 3a.

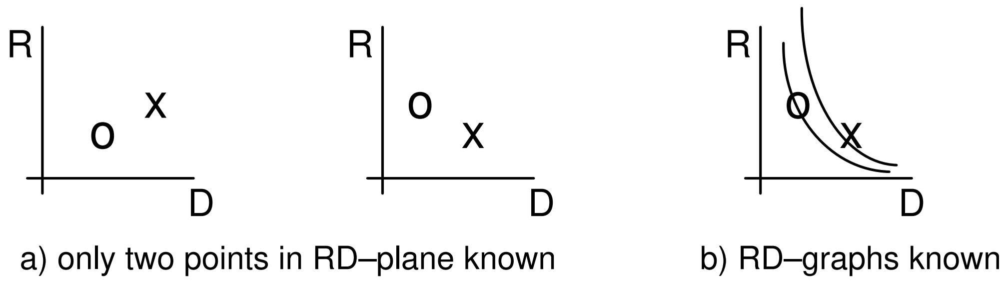

Fig. 3: Assessment of two different coding techniques from only two points in the RD–plane

In the one case, as in the left graph, rate **and** distortion of one coding technique are higher or equal than those of the other one. Here, a clear decision is made for that technique with lower rate and lower distortion. In the other case, i.e. one technique results in a higher bit rate, but in a lower distortion, as shown in the right graph of Fig. 3a., no clear decision can be made. Only if the whole RD–graph was known, as shown in Fig. 3b, a clear decision would be possible. Thus, the developed criterion estimates the RD–graph of each coding technique. As rate the amount of bits R_f for the whole frame is used. The distortion is measured as the variance σ'^2 of the coding error for the whole image except for regions of geometric distortions where higher coding errors are intentionally allowed. A mask for these regions is available from layer 3.

Using these measurements, the RD–graph of layers 1 and 2 and that of layer 3 can be described by the asymptotic function:

$$R_f\,(\sigma'^2) = \frac{K}{\sigma'^2} \tag{1}$$

This allows to estimate the two RD–graphs. The parameter K determines the efficiency of the respective layer. The lower K, the higher is the efficiency, and vice versa. This results in a selection criterion, where the layer with the lower value of K is chosen for a frame:

$$K_{I/II} = R_{f,\,I/II} \cdot \sigma'^2_{I/II} \qquad \text{and} \qquad K_{III} = R_{f,\,III} \cdot \sigma'^2_{III} \tag{2}$$

7. Performance

7.1 Efficacy

The submitted algorithm addresses improved coding efficiency and content–based functionalities. Due to the inclusion of TMN 5 as fallback coder it is guaranteed that the image quality can not fall below that achieved with TMN 5, even in situations where object–based coding can not sufficiently be applied. Content–based functionalities are provided whereby any contents can be addressed with arbitrary shapes. Error robustness is under investigation. One algorithm and one tool for this functionality are also submitted, see Subsection 7.3.4.

7.2 Added value

Due to the inclusion of TMN 5, compatibility is guaranteed to the existing standard [11].

7.3 Adaptability

7.3.1 to different input

The submitted algorithm is for general use. If the scene material is appropriate for object–based coding, the layer selector will select layer 3. If the scene can not be sufficiently coded with object–based coding, layers 1 and 2 forming TMN 5 will be selected, which can handle also general scene contents due to its block–based processing. Layer selection is done automatically frame by frame.

7.3.2 to (number of) objects

There is no difference for the coder to code one or more objects. Also the kind of objects is of no importance. Any object can be coded by object–based coding. If coding of an object happens not to be efficient in layer 3, coding efficiency of layers 1 and 2 is a lower bound.

7.3.3 to different/variable bit rates

Since all layers can handle different bit rates, also the complete codec can handle different bit rates. This is shown by the submission. Handling of bit rates which would be strongly variable from frame to frame still needs improvement for applications in CBR channels. Currently, a quality criterion fixed over a whole sequence controls the coder. Automatic coder control regulated from buffer occupancy is envisaged for future implementation.

7.3.4 to different error patterns

Error robustness is under current investigation. An algorithm for error robustness in layers 1 and 2 is proposed by Robert Bosch GmbH. A tool for error robustness in all layers is proposed by Siemens AG [15].

7.3.5 to different temporal/spatial resolution

The codec can handle different temporal resolutions. This is shown by the submission, see Table 1. The codec can also handle different spatial resolutions. CIF and QCIF modes are implemented and show to work well. However, since due to time reasons the decoder CIF mode could not have been verified before submission, only simulation results achieved in QCIF mode have been submitted.

7.4 Possibilities for improvement

Besides general optimization of all tools within the proposed algorithm, extensions of the source model are possible for certain applications. E.g. in head–and–shoulder sequences, 3–D object models can be used instead of 2–D object models.

Also the use of representative objects available from an object library is imaginable for cases where the presence of certain objects, e.g. human faces, are known. This would lead to knowledge–based coding founding a 4th layer according to the notation in [13].

A very practical improvement, which is planned for future implementation, is the extension of the object model towards occluding 2–D objects. By this, image synthesis can lead to a better quality in situations where occlusions occur.

The implemented a–posteriori layer selection is intended to be substituted by an a–priori selection thus reducing the necessary computational power.

7.5 Complexity

7.5.1 Delay

Since the coder works with a constant (high) frame rate, the maximum delay can be kept as low as 1 frame.

7.5.2 Memory

The required memory is mainly influenced by the number of images and masks that are used during the encoding process and by the number of images and masks that are used as memories, which may therefore not be used for other tasks. A rough estimation gives about 30 required images and/or masks for the encoding process and as memories. Assuming an additional overhead of 50% for the local memory of the separate modules (e.g. shape estimation or motion estimation), the absolute maximum required number of images and masks should be below 50. This leads for QCIF sequences to a maximum required memory of about 1.2 MB, for CIF sequences of about 5 MB.

7.5.3 Computational power

The computational power required for the encoder is approximately 40 MOPS/frame for QCIF and 600 MOPS/frame for CIF, leading to values of 300–600 MOPS (QCIF) and 450–900 MOPS (CIF) at the used frame rates. The decoder requires approxiamtely half the computational power of the encoder. This leads to computational requirements for a whole codec of at maximum 1 GOPS for QCIF and 1.5 GOPS for CIF sequences.

7.5.4 Complexity scalability (quality vs. h/w costs)

Scalability at both encoder and decoder sites are possible. If the decoder as low–cost version provides either only layers 1 and 2 (H.263 decoder) or only layer 3 instead of all three layers, the decoder is still compatible to the encoder. Also complexity scalability within the single layers are possible to a certain extent. The only necessity is that during the start–up phase of communication the decoder can tell the encoder his configuration.

8. Special results for object scalability

The percentages of the bit rates used for coding of the objects which have been specially selected for the object scalability tests are given in Table 3.

Test sequence	Percentage
Akiyo	93 %
Sean	85 %
Hall monitor	30 %

Table 3: Percentage of the bit rate for the selected objects with respect to the total bit rate.

Acknowledgment

The software has been developed using the image processing system Xlim which has been provided by Centre de Morphologie Mathématique (CMM), Fontainebleau, France. Part of the routines has been provided by Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. Public domain software for TMN 5 from Telenor, Oslo, Norway, has partly been used. This is gratefully acknowledged.

9. References

[1] T. Aach, A. Kaup, R. Mester, ”Statistical model–based change detection in moving video”, *Signal Processing,* Vol. 31, No. 2, March 1993, pp. 165–180.

[2] T. Aach, A. Kaup, R. Mester, ”Change detection in image sequences using Gibbs random fields: a Bayesian approach”, *Proc. Int. Workshop on Intelligent Signal Processing and Communication Systems,* Sendai, Japan, October 1993, pp. 56–61.

[3] M. Bierling, ”Displacement estimation by hierarchical blockmatching”, *3rd SPIE Symposium on Visual Communications and Image Processing,* Cambridge, MA, November 1988, pp. 942–951.

[4] CCITT, ”Video codec for audiovisual services at p * 64 kbit/s”, *Recommendation H.261,* December 1989.

[5] COST 211ter, ”SIMOC 1 (final version)”, *Doc. SIM(95)08,* February 1995.

[6] R.E. Crochiere, L.R. Rabiner, ”Interpolation and decimation of digital dignals – A tutorial review”, *Proceedings of the IEEE,* Vol. 69, No. 3, March 1981, pp. 300–331.

[7] P. Gerken, ”Object–based analysis–synthesis coding of image sequences at very low bit rates”, *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on Very Low Bit Rate Video Coding,* Vol. 4, No. 3, June 1994, pp. 228–235.

[8] P. Gerken, M. Wollborn, S. Schultz, ”Polygon/spline approximation of arbitrary image region shapes as proposal for MPEG–4 tool evaluation – Technical description”, *Doc. ISO/IEC JTC1/SC29/WG11 MPEG95/360,* November 1995.

[9] ISO/IEC, ”Information technology – Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s”, *International Standard IS 11172,* August 1993.

[10] ISO/IEC, ”Information technology – Generic coding of moving pictures and associated audio information”, *International Standard IS 13818,* 1995.

[11] ITU–T, ”Video coding for low bitrate communication”, *Draft Recommendation H.263,* July 1995.

[12] E. Jensen, K. Rijkse, I. Lagendijk, P. van Beek, ”Coding of arbitrarily shaped image segments”, *Proc. Workshop on Image Analysis and Synthesis in Image Coding,* Berlin, Germany, October 1994, pp. E2.1–E2.4.

[13] H.G.Musmann, ”A layered coding system for very low bit rate video coding”, accepted for publication in *Signal Processing: Image Communication, Special Issue on Very Low Bit–rate Video.*

[14] L.R. Rabiner, E. Gold, *Digital Signal Processing*. Prentice–Hall, Englewood Cliffs, NJ, 1975.

[15] F. Seytter, ”Hybrid ARQ type–II error protection tool”, *Doc. ISO/IEC JTC1/SC29/WG11 MPEG95/xxx,* November 1995.