The Block–based Coder Mode in an Object–based Analysis–Synthesis Coder

Jörn Ostermann*

AT&T Bell Labs, Room 4E518, 101 Crawfords Corner Rd, Holmdel NJ 07733–3030
Phone (908) 949 6683, email: osterman@big.att.com

Abstract

For object–based analysis–synthesis coding (OBASC) of moving images, each moving object is described by three parameter sets defining its motion, shape and surface color. They are coded using an object–dependent parameter coding. Compared to block–based hybrid coding, OBASC requires the additional transmission of shape parameters. This is only justified, if it increases the overall coding efficiency. The efficiency of the block–based and the object–based coders depends to a large extend on the size of the moving objects. Objects smaller than 256 pels should be coded by a block–based coder whereas larger objects are preferably coded by an object–based coder.

1: Introduction

Block–based hybrid coding according to the standards H.261 and MPEG–1 describes moving objects by motion (2D displacement vectors) and color (DCT–coefficients) parameters of square blocks [8]. This corresponds to a source model "2D square blocks moving parallel to the image plane". At boundaries of moving real objects, this source model cannot describe correctly the motion in the image sequences because there are two different motions within one block. At low data rates, this inefficiency gets visible as mosquito and block artifacts. In contrast to block–based image coding, OBASC [7][12] allows to describe arbitrarily shaped objects by means of motion, shape and color parameters. It requires the additional transmission of shape parameters. Therefore, motion at object boundaries can be described correctly. The source models moving rigid 3D objects [12], moving flexible 3D objects and moving flexible 2D objects [3][4] have been investigated. The latter has been us for coding of image sequences at data rates down to 16 kbit/s. For videophone test sequences, these coders showed an image quality superior to results achieved with block–based hybrid coders like RM8 [1] (H.261) or TMN2 [9]. However, the additional transmission of shape parameters for a moving object is only justified, if the overall coding efficiency using the shape parameters is higher than the efficiency achieved by a block–based description of the moving object.

In order to achieve an efficient coding of each moving object of the scene, the object–dependent parameter coding has to switch between an object–based coder mode and a block–based coder mode. The block–based mode is known from existing and upcoming standards like H.261. Implementations of an object–based mode have been published in the environment of object–based analysis–synthesis coding [7][12][3][4][13][2].

In [2], a switching of coder modes is presented which compares the PSNR (Peak Signal to Noise Ratio) and the bit–rates required in its block–based and object–based modes. In its object–based mode, only motion and shape parameters of the moving objects are transmitted. Color parameters are always coded in the block–based mode [16]. In this contribution, an analysis of the different coding artifacts of an object–based coding mode and of a block–based coding mode as well as the difference in bit–rate required to transmit the parameters of the source models is presented, in order to derive a switching criterion for selecting the appropriate coder mode.

Section 2 reviews the important aspects of OBASC based on the source model of flexible 2D objects. In Section 3, a block–based reference coder is developed which codes images similar to an object–based coder without requiring the transmission of shape parameters for the moving objects. In section 4, the influence of limited bit–rate on coding artifacts is investigated. In section 5, the bit–rates which block–based coders and object–based coders require in order to code an object with good subjective quality are estimated.

*Research was carried out at the Institut für Theoretische Nachrichtentechnik and Informationsverarbeitung, Universität Hannover, Germany.

2: OBASC based on the source model of moving flexible 2D objects

An image coder based on the source model of moving flexible 2D objects describes the shape of each moving object by a 2D silhouette. The motion of a object is described using a displacement vector field. For the displacement vector field, one displacement vector with half pel accuracy is estimated and transmitted for each block of size 16 · 16 pel. Color parameters are extracted from the image sequence.

Inside one object, a dense displacement vector field is computed from the transmitted vectors using bilinear interpolation resulting in a smooth change of motion inside one object and motion discontinuities at object boundaries. Hence, inside moving objects blocking artifacts are avoided. The color parameters of the object are taken from previously transmitted images. Using the motion and shape parameters, image areas which are uncovered due to object motion can be computed. In these uncovered areas, the color parameters of the objects are updated using the current image. Moving objects with these three sets of parameters are named MC–Objects. Due to the shape parameters, it is possible to compute a subjectively correct prediction image. Those image regions, which cannot be described with a sufficient subjective image quality using MC–objects are named MF–objects. They are described using 2D shape parameters and color parameters extracted from the current image. The detection of MF–objects exploits the relevance of errors to the human observer (irrelevancy reduction) and neglects geometrical distortions like small position and shape errors of the moving objects.

The 2D silhouette of each MC–object is coded using a polygon/spline approximation and motion compensated prediction [5]. The quality measure for shape approximation d^{*}_{max} is the maximum allowable deviation between the estimated and the approximated shape. The 2D shape parameters of the MF–objects are coded without temporal prediction using a polygon/spline approximation. Experiments [6][11] have shown that it is more efficient to code shape parameters using the polygon/spline approximation and the color parameters for the areas inside the coded shape than approximating the shape of the MF–objects by square blocks of size 8 · 8 pels and then code the color parameters of the blocks. The gain in bit–rate due to the transmission of these shape parameters can be as high as 50%. Whereas the shape parameters of moving objects are used to describe the physical world, the shape parameters of MF–objects are only used to increase coding efficiency and to avoid mosquito artifacts.

3: Block–based reference coder

The hybrid coder according to the standard H.261 subdivides an image into *macro blocks* of size 16 · 16 pels [8]. For each macro block, one motion vector can be transmitted. For coding of color parameters (DCT coefficients) a macro block is subdivided into 4 *blocks* of size 8 · 8 pels (Fig. 3.1). For each block, it is decided whether color parameters are transmitted. In the next section, *boundary blocks* will be considered in addition to blocks and macro blocks. In a boundary block of size 8 · 8 pels, differently moving objects are present.

Each macro block can be coded in three different modes. In the most simple case, the selection of the appropriate coding mode is based on the measurement of the prediction error. In the CCITT reference model RM8, for each macro block, the energy of the real image (intra mode), the energy of the prediction error without (inter mode) and with motion compensation (MCP–mode) is measured [1]. The mode with the smallest energy is selected. For a optimal coding efficiency, the selection criteria has to select the mode with the smallest rate R given a distortion D:

$$R(D) = Min(R_{Intra}(D), R_{Inter}(D), R_{MCP}(D)). \quad (3.1)$$

Here, R_{intra} and R_{inter} account for the data rates of color parameters and R_{MCP} allows for the rate of color parameters and for the additional rate required for the displacement vector. Generally, the coder uses the MCP–mode in moving areas. In case of estimation errors or limits of the underlying source model, the coder switches "back" to the simpler modes inter or intra.

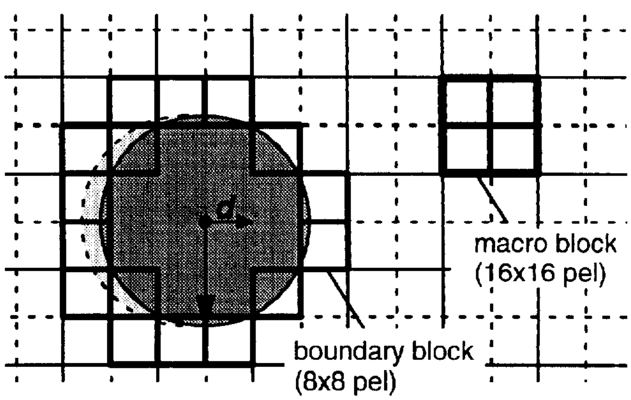

Fig. 3.1 A circle is moving with a displacement *d*. The areas uncovered due to object motion are marked in a light grey. The image is devided into macro blocks consisting of four blocks. The marked blocks on the boundary of the circle are boundary blocks containing two differently moving objects.

The reference coder developed in this section is based on RM8 and will be enhanced in several ways. In order to increase the coding efficiency of the block–based coder and to make it more comparable to the object–based coder, motion compensation will be modified:

1. Displacement vectors are estimated and coded with half pel accuracy [9].
2. For each macro block, it will be transmitted whether the displacement vectors should be interpolated.
3. For each pel of a marked macro block, a displacement vector field with an amplitude resolution of 0.5 pel is calculated using bilinear interpolation of the transmitted vectors [14].

The interpolation of the displacement vector field of OBASC takes into account shape information only at the boundaries of MC–objects. With the extensions 1. to 3., the block–based coder is now able to do a motion–compensated prediction which differs only in boundary blocks from OBASC, hence blocking artifacts are avoided inside moving objects. Like in OBASC, the coder control in non–boundary blocks can consider geometrical distortions as irrelevant to the human observer.

In OBASC, the transmission of shape parameters for MF–objects increases the coding efficiency. Therefore, this method is adapted to the block–based coder:

4. Areas, for which color parameter have to be transmitted, are described using a polygon/spline approximation.
5. Color parameters are transmitted for these arbitrarily–shaped regions only [10][15].

Due to these extensions, coding of color parameters is identical for the block–based and the object–based coder. Furthermore, the block–based and the object–based coder work identical in all but boundary blocks. At object boundaries, the OBASC is able to describe the physical world correctly. As far as hardware complexity is concerned, the two coders differ mainly in image analysis which is much more complex for the OBASC.

4: Coding artifacts

Fig. 3.1 shows for a moving object the problems for selecting the appropriate coding mode in order to achieve a good image quality. In order to describe the current image, OBASC requires in addition to the color parameters transmitted for previous images the current motion and shape update parameters as well as the color parameters of the uncovered areas. The block–based coder has to transmit motion parameters and color parameters for each boundary block, since a block–based motion description cannot be correct for boundary blocks. Assuming identical previously coded images, differences in picture quality show up in boundary blocks only.

For OBASC, image quality of the boundary blocks depends on

- the accuracy of the estimated and transmitted shape and motion parameters
- the coding error of the color parameters transmitted for the uncovered areas.

For videophone test sequences, it has been shown that the estimation of motion and shape parameters is possible with sufficient accuracy and the bit–rate for coding the color parameter of the uncovered areas is high enough to ensure good image quality [4][12].

For the block–based reference coder, image quality of boundary blocks depends on

- the inherently wrong motion compensation
- the coding error of the color parameters transmitted for the boundary blocks.

Since the block–based coder does not know the shape of the moving objects, motion compensation creates a high frequency prediction error signal which has to be coded for all boundary blocks. Due to the limited data rate, a relatively high coding error has to be accepted. As long as the boundary of a moving object is moving within the same boundary block, color parameters of the complete boundary block will be coded in consecutive frames. Therefore, subjective image quality is not only influenced by the coding error of the color parameters but also by the temporal unrest and edge busyness caused by the repeated coding of the boundary block. This coding artifacts become visible as soon as the coding error is higher than 42 dB. Since an object–based coder is able to compensate motion in boundary blocks correctly and transmits color parameters only for the uncovered areas and not for the complete boundary blocks, no temporal unrest or edge busyness occurs. Therefore, acceptable image quality is achieved with coding errors of the color parameters as high as 36 dB. Hence, for the same subjective image quality, the coding error in the object–based coding mode may be up to 6 dB higher compared to block–based coding.

5: Data rates

In the following paragraphs, the data rates required for coding of boundary blocks is estimated in order to develop a switching criterion for selecting the appropriate coding mode for each object in a scene. Since the object–based and the block–based coder transmit the same motion parameters and hence require the same data rate for their coding, the rate for motion information will not influence the switching criterion and will be neglected in the following investigations.

For block–based coding, the required data rate for each boundary block of a macro block is given by

$$R^B = \sum_{boundary\ blocks} R_C^B, \quad (5.1)$$

where R_C^B gives the average rate for coding the color parameters of a boundary block.

For object-based coding, the data rate for each boundary block is given by

$$R^O = \sum_{boundary\ blocks} (R_{F,MC}^O + R_{C,UA}^O), \quad (5.2)$$

where $R_{F,MC}^O$ gives the average rate for coding the shape parameters of the MC–object and $R_{C,UA}^O$ the average bit–rate for coding the color parameters of the uncovered areas. Assuming identical subjective image quality, the appropriate coding mode is selected according to

$$coding\ mode = \begin{cases} block-based & \text{if } R^B < R^O \\ object-based & else \end{cases} \quad (5.3)$$

In order to get an estimate of the rates R_C^B, $R_{F,MC}^O$ and $R_{C,UA}^O$, the test sequences *Claire* and *Miss America* were analyzed.

In order to measure the rate $R_{F,MC}^O$, 3D model objects of the persons in the sequence were generated [12]. By moving these model objects in a 3D model world and projecting them into the image plane, the silhouettes of the objects were obtained in different sizes. Fig. 5.1 shows the bit rate $R_{F,MC}^O$ required for coding the shape parameters of a MC–object with a quality measure of $d^*_{\max} = 1.4$ pel [12] as a function the object size represented by the number of boundary blocks. For small objects, this rate is higher than for large objects due to the higher curvature of the object silhouette. The rate $R_{F,MC}^O$ can be approximated by

$$R_{F,MC}^O(n_R) \approx M \cdot (e^{-0.1 \cdot (n_R - 30)} + 5.8) \text{ [bit/boundary block]} \quad \text{for } n_R \geq 4, \quad (5.4)$$

with the number of boundary blocks n_R and the scaling factor M. For non–predictive coding M equals 1, for motion–compensated predictive coding using $d^*_{\max} = 1.4$ pel, M decreases to 0.5. Eq. (5.4) is not valid for $n_R<4$, because then the assumption of estimating the shape of the moving objects correctly is not valid anymore.

The rate $R_{C,UA}^O$ for coding the color parameters of the uncovered areas is given by

$$R_{C,UA}^O = n_{UA} \cdot r_{C,UA}, \quad (5.5)$$

with n_{UA} the number of pels of the uncovered areas and $r_{C,UA}$ the average rate required for transmitting the color parameters of 1 pel.

For the test sequences, an average $n_{UA}=7$ pel/boundary block and $n_{UA}=10$ pel/boundary block were measured. The rate $r_{C,UA}$ depends to a large extend on the image signal and the efficiency of a background memory, which can easily be incorporated into a object–based coding scheme [3]. The measurements for a coding error of PSNR=36 dB gave $0 \leq r_{C,UB} \leq 1.0$ bit/pel. Fig. 5.2 shows the data rate for coding the boundary blocks of an object for different n_{UA} and $r_{C,UA}$. Fig. 5.3 gives the same rate normalized to bit/boundary block. If the block–based coder requires a bit–rate below the lower curve in Fig. 5.2 and 5.3, it will outperform a object–based coder. Similarly, if the block–based coder requires data rates above the upper curve, an object–based coder will outperform the block–based coder. If the rate is

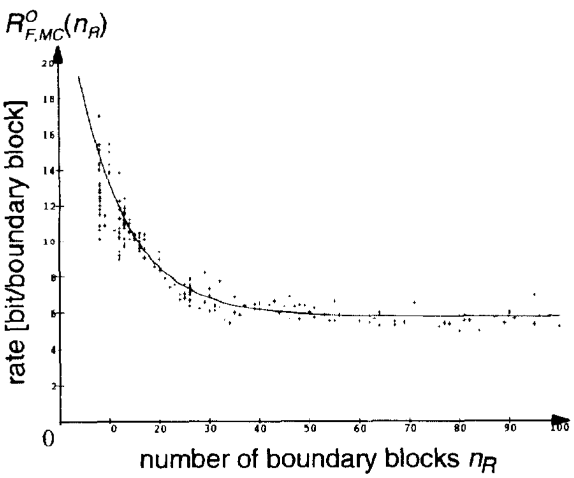

Fig. 5.1 Data rate in bit/boundary block for coding the shape parameters of a object during its first occurence (intra mode) as function of the object size represented by the number of boundary blocks n_R. The solid curve is the model curve according to Eq. (5.4) with M=1. In case of predicitve coding the data rates have to be diveded by two (M=.5). For the test sequences *Claire* and *Miss America*, the no. of boundary blocks is n_R=87 and n_R=95, respectively.

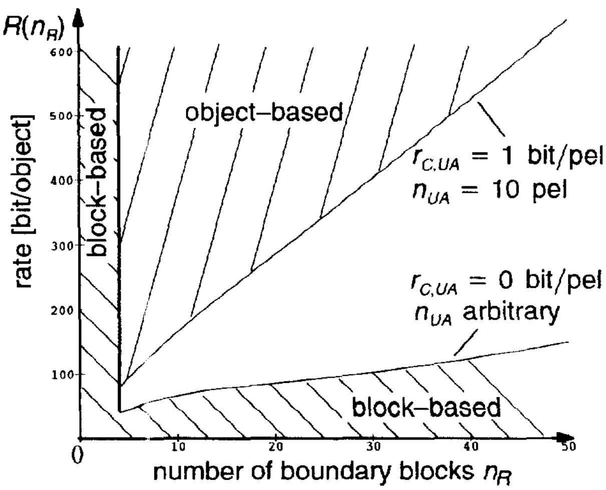

Fig. 5.2 Data rate $R^O(n_r)$ in bit/object. Shape parameters of MC–objects are predictively coded (M=0.5). Color parameters of uncovered areas are coded with different rates $r_{C,UA}$ in bit/pel.

between the two curves, the size of the uncovered area has to be determined in order to make a correct decision according to Eq. (5.3). The test sequences (CIF, 10 Hz) were coded at a data rate of 64 kbit/s. In Fig. 5.3, the data rates spent for coding the boundary blocks are given for RM8 and the object–based coder. Although the block–based coder spends approximated 25 % of the bit rate in the boundary blocks for the test sequence *Claire*, it is still not possible to achieve a subjectively acceptable image quality, whereas the object–based coder provides sharper images without blocking artifacts. Experiments conducted so fare indicate that moving objects covering more than four boundary block or objects larger than 256 pels should be coded in a object–based mode.

6: Conclusions

In order to compare the bit–rates required for coding a moving object in a block–based and in a object–based mode, a block–based reference coder is assumed which uses features like interpolation of the displacement vector field and color coding for arbitrarily shaped regions developed for object–based coding. The transmission of shape parameters for the moving objects is not required. Hence, the coders mainly differ in their ability to describe motion discontinuities properly. Experiments indicate that objects larger than 256 pels are preferably coded in an object–based mode whereas smaller objects are more efficiently coded in a block–based mode.

7: Literature

[1] CCITT SG XV, Doc. #525, Description of Reference Model 8 (RM8), 1989.

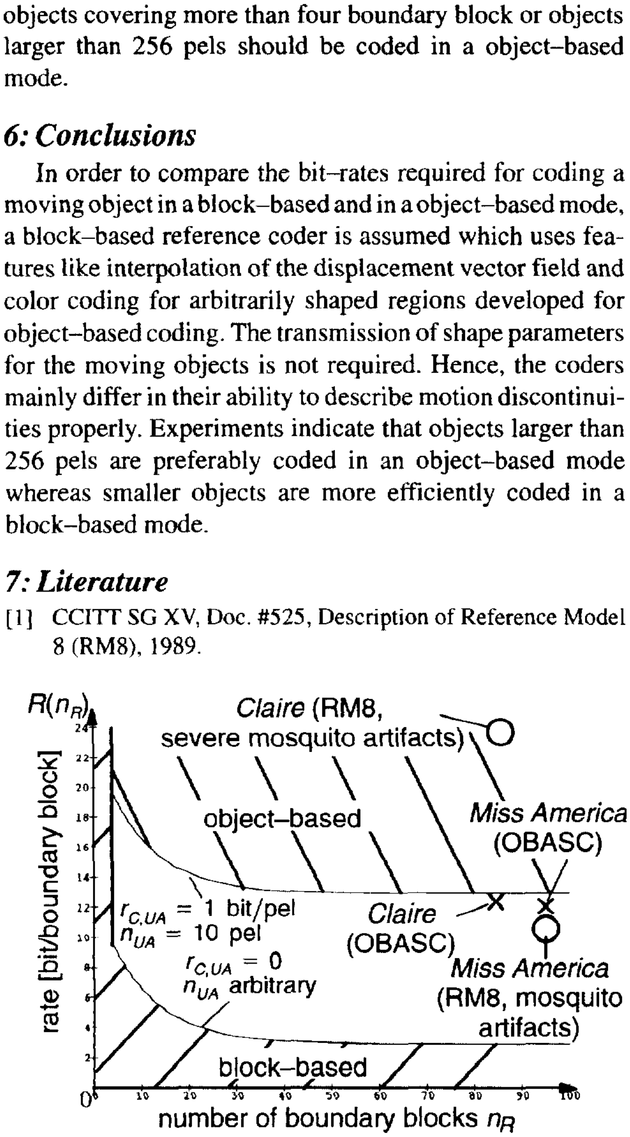

Fig. 5.3 Data rate in bit/boundary block for predictive Coding of MC–objects (M=0.5) and different rates $r_{C,UA}$ for coding color parameters in bit/pel. Crosses and circles mark the data rates which object–based and block–based coders spent for coding boundary blocks.

[2] M.F. Chowdhury, A.F. Clark, A.C. Downton, E. Morimatsu, D.E. Pearson, "Architecture and performance of a switched model/block–based image coder", COST 211ter European workshop on new techniques for coding of video signals at very low bitrates, Hannover, Germany, December 1993.

[3] P. Gerken, "Object–based analysis–synthesis–coding of image sequences at very low bitrates", COST 211ter European workshop on new techniques for coding of video signals at very low bitrates, Hannover, Germany, December 1993.

[4] M. Hötter, "Object–oriented analysis–synthesis coding based on moving two–dimensional objects", Signal Processing: Image Communication, Vol. 2, No. 4, December 1990, pp. 409–428.

[5] M. Hötter, "Predictive contour coding for an object–oriented analysis–synthesis coder", IEEE International Symposium on Information Theory, San Diego, California, U.S.A., S. 75, January 1990.

[6] M. Hötter, "Objektorientierte Analyse–Synthese–Codierung basierend auf dem Modell bewegter, zweidimensionaler Objekte", PhD thesis, Universität Hannover, Fortschritt–Berichte VDI, Reihe 10, Nr. 217, Hannover, 1992.

[7] H.G. Musmann, M. Hötter, J. Ostermann, "Object–oriented analysis–synthesis coding of moving images", Signal Processing: *Image Communication*, Vol. 1, No. 2, pp. 117–138, Nov. 1989.

[8] ITU–T Recommendation H.261: *Video codec for audiovisual services at* p *x 64 kbit/s*. March 1993.

[9] ITU–T Study Group 15, Working Party 15/1, TMN–2, Doc. LBC–94–124.

[10] E. Jensen, K. Rijkse, I. Lagendijk, P. van Beek, "Coding of arbitrarily shaped image segments", Workshop on image analysis and synthesis in image coding, Berlin, Germany, October 1994, E2.

[11] J. Ostermann, H. Li, "Detection and Coding of Model Failures in an Analysis–Synthesis Coder Based on Moving 3D–Objects", 3rd International Workshop on 64 kbit/s Coding of Moving Video, Rotterdam, The Netherlands, No. 4.4, September 1990.

[12] J. Ostermann, "Object–based analysis–synthesis Coding based on the source model of moving rigid 3D objects", Signal Processing: Image Communication, No. 6. pp. 143–161, 1994.

[13] J. Ostermann, P. Gerken, "Object–oriented analysis–synthesis coding based on the source models of moving 2D– and 3D–objects", ISO/IEC JTC1/SC29/WG11, MPEG–4 Seminar, Doc. (93)710, New York, July 1993.

[14] D. Qian, "Block–based motion compensated subband coding at very low bit rates using a psychovisual model", International Workshop on Coding Techniques for very low bit–rate video (VLBV 94), Colchester, Großbritannien, No. 6.8, April 1994.

[15] T. Sikora, B. Makai, "Low Complex shape–adaptive DCT for generic and functional coding of segmented video", Workshop on image analysis and synthesis in image coding, Berlin, Germany, October 1994, E1.

[16] L. Ward, S. Marlow, "H.261 Colour Parameter coding for SIMOC", Workshop on image analysis and synthesis in image coding, Berlin, Germany, October 1994, E3.