SOURCE MODELS FOR CONTENT-BASED VIDEO CODING

Jörn Ostermann[1], Markus Kampmann[2]

[1]AT&T Labs-Research, USA, osterman@research.att.com,

[2]Universität Hannover, Germany, kampmann@tnt.uni-hannover.de

ABSTRACT

Different source models for content-based video coding are presented. An object-based analysis-synthesis coder describes video objects by means of motion, shape and texture parameters. In contrast to block-based coding, the representation of arbitrarily shaped objects in a real scene is possible. Using knowledge about the scene content and about the object's behavior, object-based coding is extended to knowledge-based coding and semantic coding, respectively. Compared with other content-based video coding concepts like MBASIC and MPEG-4, the above mentioned coders provide an integrated and more efficient framework for coding video using 3D source models.

1. INTRODUCTION

Block-based hybrid coding according to the standards H.261 and H.263 describes moving objects by motion (2D displacement vectors) and color (DCT-coefficients) parameters of square blocks [4]. This corresponds to a source model "2D square blocks moving parallel to the image plane" (2DSB). Since this source model cannot describe the shape of real objects in a video sequence, it is not suited for representing objects in a real scene. In contrast to block-based video coding, object-based analysis-synthesis coding (OBASC) [3][5] allows to describe arbitrarily shaped video objects by means of motion, shape and color parameters. It requires the additional transmission of shape parameters. The source models moving flexible 2D objects (F2D) [1][2], moving rigid 3D objects (R3D) [5], moving flexible 3D objects (F3D) [6] and moving articulated 3D objects (A3D) [11] have been investigated. Using scene knowledge for improving coding efficiency, we extend OBASC based on a 3D source model to a knowledge-based coder [10]. There, a face in the video sequence is detected and the predefined 3D face model *Candide* is adapted to the detected face in the image sequence. By extending the knowledge-based coder, a semantic coder uses high-level semantic parameter for describing the behavior of objects in the video sequence. In case of a face in the sequence, semantic parameters describe the facial expressions of the person.

In Section 2, we describe the concept of a generic OBASC. In Section 3, we extend this coder to a knowledge-based coder. A semantic coder is presented in Section 4. In Section 5, we provide an overview on how a layered coder [9] can be used to switch between different source models. In Section 6, we compare the layered coder with other content-based video coding concepts like the MBASIC system and the MPEG-4 video coding standard.

2. OBASC

OBASC [3] subdivides each image of a sequence into moving objects and describes each object m by a model object with three sets of parameters $A^{(m)}$, $M^{(m)}$ and $S^{(m)}$, defining its motion, shape, and color, respectively. Motion parameters define position and motion of the object. Color parameters denote the luminance as well as the chrominance reflectance on the surface of the object. In computer graphics, they are sometimes called texture. Fig. 1 is used to explain the concept and structure of OBASC. Instead of a frame memory used in block based hybrid coding, OBASC requires a memory for parameters to store the coded and transmitted object parameters A'(m), M'(m) and S'(m). The parameter memories in coder and decoder contain the same information. Evaluating these parameter sets, image synthesis synthesizes a model image s'_k which is displayed at the decoder. The parameter sets of the memory and the current image s_{k+1} are the input to image analysis.

The task of image analysis is to analyze the current image s_{k+1} to be coded and to estimate the parameter sets $A^{(m)}_{k+1}$, $M^{(m)}_{k+1}$ and $S^{(m)}_{k+1}$ of each object m. First, new motion and shape parameters are estimated for each object in order to reuse most of the already transmitted color parameters $S'^{(m)}_k$. Objects for which the correct motion and shape parameters can be estimated are denoted as *MC-objects* (*model compliance*). Then, image areas which cannot be described by MC-objects using the transmitted color parameters $S'^{(m)}_k$ and the new motion and shape parameters $A^{(m)}_{k+1}$, $M^{(m)}_{k+1}$, respectively, are detected. These areas of *model failures* (MF) are defined by 2D-shape and color parameters only and are referred to as *MF-objects*. The detection of MF-objects exploits that small position and shape errors of the model objects - referred to as *geometrical distortions* - do not disturb subjective image quality. Thus, MF-objects are reduced to those image regions with significant differences between the motion- and shape-compensated prediction image and the current image s_{k+1}. They tend to be small in size. This allows to code color parameters of MF-objects with high quality,

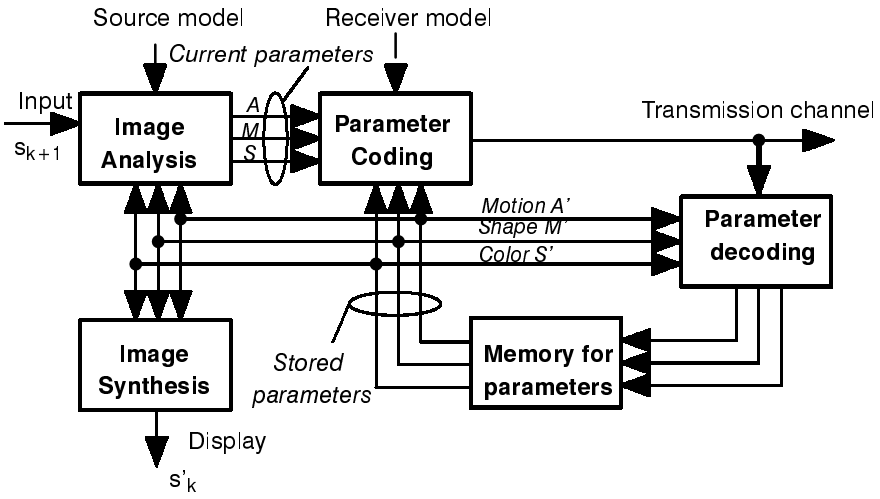
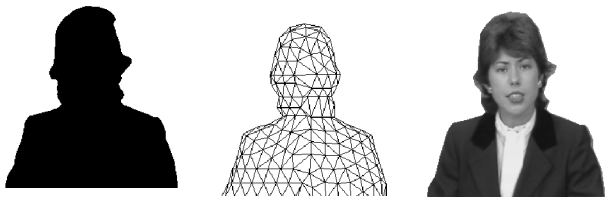


Fig. 1 Block diagram of an object-based analysis-synthesis coder.

thus avoiding subjectively annoying quantization errors. Since the transmission of color parameters is expensive in terms of data rate, the total area of MF-objects should not be greater than 4% of the image area assuming 64 kbit/s, CIF and 10Hz. Depending on the *object class* MC/MF, the parameter sets of each object are coded using predictive coding techniques. Motion and shape parameters are coded, transmitted and decoded for MC-objects, as well as shape and color parameters for MF-objects. In OBASC, the suitability of source models can be judged by comparing the overall bit rates required for coding the same image sequence at the same image quality. Image quality is influenced mainly by the algorithm for detecting model failures and by the bit rate available for coding the color parameters of model failures.

OBASC has been implemented for different source models. The source model flexible 2D objects (F2D) describes each moving object by its 2D shape and a dense displacement vector field inside this shape. The 2D shape is coded using a polygon/spline approximation and motion compensated prediction [1]. Texture is coded using DCT. The source model rigid 3D objects (R3D) describes each object with an opaque 3D wireframe. Fig. 2 shows how a model object is created from an object silhouette. The 3D shape of the object is computed from the silhouette using a distance function.



(a) (b) (c)

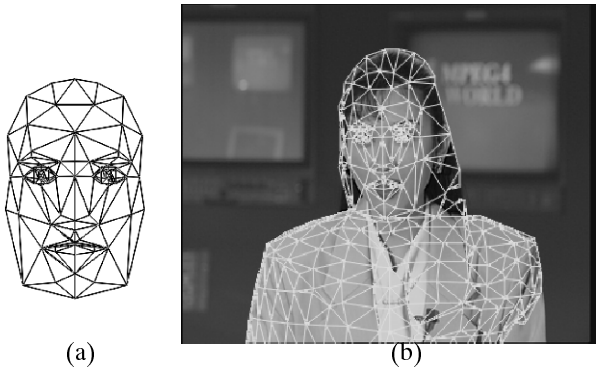
Fig. 2 Processing steps from object silhouette to model object: (a) object silhouette; (b) 3D wireframe; (c) model object with color parameters projected onto.

Therefore, we only code the 2D object silhouette similar to the source model F2D. The decoder uses the distance function to create the actual 3D shape. The 3D motion of a model object is described by six parameters defining translation and rotation. In addition to the properties of the source model R3D, the source model F3D allows for local flexible shifts on the surface of the model-object shape modelled by a flexible skin [6]. This flexible skin can be moved tangentially to the surface of the object without changing the shape of the object. It allows to model local 2D motion in contrast to the global motion parameters $\boldsymbol{R}^{(m)}$ and $\boldsymbol{T}^{(m)}$. Based on R3D, the source model of moving 3D articulated objects (A3D) is introduced in [11]. There, objects may be articulated i.e. they consist of several rigid object components linked to each other by joints. A person with head, arms and legs or an industrial robot are examples of an articulated object.

Tab. 1 shows the total area of MF-Objects for the different 3D source models when coding head and shoulder scenes. Compared with R3D, the source models F3D and A3D reduces the area of MF-objects from 4% to 3% of the image area. Since the transmission of color parameters for the MF-objects is expensive in terms of data rate, a reduction of the overall data rate by more than 16% is achieved [6][11]. We expect that adding flexible surface shifts to the A3D will further reduce the size of MF-objects and overall bit rate.

3. KNOWLEDGE-BASED CODING

As soon as we recognize what type of object is in the scene a knowledge-based coder can be used for video coding. Here we use a face detection algorithm for searching faces in the video sequence [10]. As soon as a face is detected, we adapt the predefined 3D face model *Candide* automatically to the generic wireframe of the person (Fig. 3). Using this scene knowledge gives us two advantages: First, motion estimation and compensation is improved due to the



(a) (b)

Fig. 3 Adaptation of the face model *Candide*: (a) 3D face model; (b) 3D wireframe of *Akiyo* with integrated face model projected onto original frame.

better knowledge of the object shape. Secondly, the detection of model failures is improved. Within the face area, the detection is more sensitive to errors while larger errors in the remaining parts of the object are allowed without decreasing subjective image quality. Because of these two advantages, the MF-objects are reduced to 2.5% of the image area (Tab. 1), thus improving coding efficiency compared with OBASC.

4. SEMANTIC CODING

A semantic coder can model the behavior of some objects in a video scene. Especially, the facial expressions of a human person are under investigation. In [8], a more complex 3D face model consisting of more triangles than *Candide* is adapted automatically to the face in the sequence. Then, facial expressions can be created by deforming the face model's wireframe (Fig. 4). This allows the transmission of semantic parameters like smile instead of MF objects. We expect that this technology will further reduce the rate for coding of head and shoulder scenes. Using semantic coding to synthesize offline video sequences of a human based on the spoken text has been shown in [7]. However, reliable real-time video analysis remains a challenge.

Source models	Area of MF-objects [% of image area]
R3D	4.0
F3D	3.0
A3D	3.0
Knowledge-based	2.5

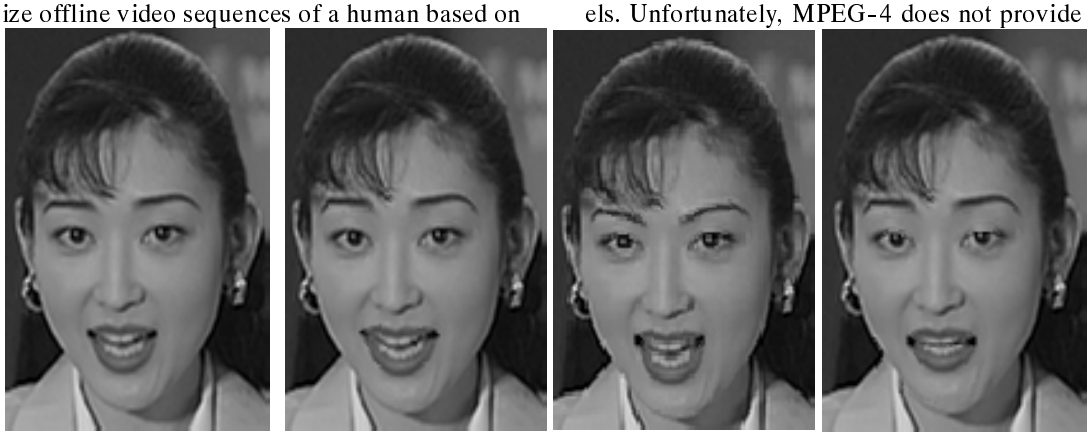
Tab. 1 Area of MF-objects for different source models.

5. LAYERED CODING

In Fig. 5 we show how the different coders presented in this paper may work together. The Layer I coder encodes independently each image using color parameters only. The hybrid coder uses the source model 2DSB transmitting motion and color parameters. Layers 3 to 5 work as described above. The purpose of the Layer Selector is to select the most efficient coder for the given video sequence. We can view this as an extension of the coding mode selection known from standardized video coders like H.261/H.263.

6. MPEG-4 AND OTHER CONTENT-BASED CODERS

The recent video coding standard MPEG-4 provides content-based video coding capabilities by coding the shape of an object as a binary mask. The texture of the objects is coded using the DCT. The decoder displays only pictures of the texture that are inside the transmitted object shape. Inherently, MPEG-4 is still a block-based video coder with the well known blocking and mosquito artifacts. MPEG-4 provides also for coding of face and body animation parameters (FAP, BAP). These can be used to animate human wireframe models. Unfortunately, MPEG-4 does not provide the tools to



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

Fig. 4 Synthesis of facial expressions (*Akiyo*): (a) original image; (b)(c)(d) images with various synthesized facial expressions.

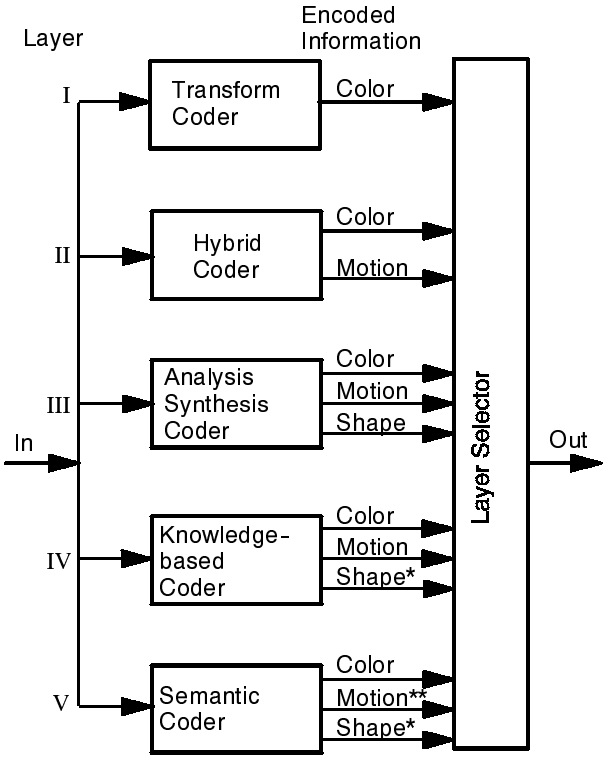


Fig. 5 Block diagram of a layered coder (from [9]).

efficiently update only parts of the texture maps of these models. Therefore, it is not possible to build a MPEG-4 compliant semantic coder that displays photo-realistic animations.

The MBASIC system combines a hybrid coder with the source model 2DSB and a knowledge-based coder using a predefined 3D face model. Since these source models cannot be integrated seamlessly, coded images show different distortions inside and outside of the face area. In [12], a block-based selector is used to switch a coder between Layer II and Layer V. This method improves coding efficiency but does not enable content-based functionalities like object-based editing.

7. CONCLUSIONS

We present a layered coder that switches adaptively its source models as further information about a video sequence becomes available. One Layer is represented by an OBASC based on 3D source models that are used to code arbitrary moving objects in a video scene. The shape of these objects is modelled using a 3D wireframe. This flexible shape representation allows us to update the object shape with a predefined and adapted wireframe if the Layer with a knowledge-based coder recognizes a known object or object part like a face in a body. We animate this predefined wireframe in case that semantic knowledge becomes available and we can estimate its parameters. This consistent shape representation allows us to seamlessly integrate video coding and animation that modern multimedia communciation standards like MPEG-4 are still lacking.

8. LITERATURE

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

[2] M. Hötter, "Optimization and efficiency of an object-oriented analysis-synthesis coder", IEEE Transactions on Circuits and Systems for video technology, Vol. 4, No. 2, pp. 181-194, April 1994.

[3] 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.

[4] ITU-T Recommendation H.263 - Video coding for low bit rate communication, February 1998.

[5] 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.

[6] J. Ostermann, "Object-oriented analysis-synthesis Coding (OOASC) based on the source model of moving flexible 3D objects", IEEE Trans. on Image Processing, Vol. 3, No. 5, Sep. 1994.

[7] E. Cosatto, G. Potamianos and H. P. Graf, "Autio-Visual unit selection for the synthesis of photo-realistic talking heads," IEEE Int. Conf. Multimedia and Expo, New York, July 2000.

[8] M. Kampmann, R. Farhoud, "Precise face model adaptation for semantic coding of videophone sequences", *Picture Coding Symposium* (PCS '97), Berlin, Germany, pp. 675-680, Sep. 1997.

[9] H. Musmann, "A layered coding system for very low bit rate video coding", *Signal Processing: Image Communication*, Vol. 7, Nos. 4-6, Nov. 1995, pp. 267-278.

[10] M. Kampmann, J. Ostermann, "Automatic Adaptation of a Face Model in a Layered Coder with an Object-based Analysis-Synthesis Layer and a Knowledge-based Layer", *Signal Processing: Image Communications*, Vol. 9, No. 3, March 1997, pp. 201-220.

[11] G. Martinez, "Analysis-Synthesis Coding Based on the Source Model of Articulated Three-Dimensional Objects", *Picture Coding Symposium* (PCS '99), Portland, USA, pp. 213-216, April 1999.

[12] P. Eisert, T. Wiegand, and B. Girod, "Model-Aided Coding: A New Approach to Incorporate Facial Animation into Motion-Compensated Video Coding," IEEE Trans. on Circuits and Systems for Video Technology, vol. 10, no. 3, pp. 344-358, April 2000