OBJECT-ORIENTED ANALYSIS-SYNTHESIS CODING OF IMAGES

Hans Georg MUSMANN, Michael HÖTTER and Jörn OSTERMAN
Institut für Theoretische Nachrichtentechnik und Informationstechnik, Universität Hannover 1, Fed. Rep. Germany

pp. 117–138
OBJECT-ORIENTED ANALYSIS–SYNTHESIS CODING OF MOVING IMAGES

Hans Georg MUSMANN, Michael HÖTTER and Jörn OSTERMANN

Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover, Appelstr. 9a, D-3000 Hannover 1, Fed. Rep. Germany

pp. 117–138
OBJECT-ORIENTED ANALYSIS–SYNTHESIS CODING OF MOVING IMAGES

Hans Georg MUSMANN, Michael HÖTTER and Jörn OSTERMANN
Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover, Appelstr. 9a, D-3000 Hannover 1, Fed. Rep. Germany

Received 13 February 1989
Revised 13 May 1989

Abstract. An object-oriented analysis-synthesis coder is presented which encodes objects instead of blocks of $N \times N$ picture elements. The objects are described by three parameter sets defining the motion, shape and colour of an object. The parameter sets are obtained by image analysis based on source models of either moving 2D-objects or moving 3D-objects. Known coding techniques are used to encode the parameter sets. An object-depending parameter coding allows to introduce geometrical distortions instead of quantization errors. Using the transmitted parameter sets an image can be reconstructed by model-based image synthesis.

Experimental results achieved with a first implementation of the coder are given and are discussed.

Keywords. Model-based image analysis, parameter coding, model-based analysis–synthesis image coding.

1. Introduction

In order to encode moving video signals at low bit rates each image of a sequence is usually subdivided into blocks of $N \times N$ picture elements (pels) and the luminance and chrominance signals of each block are encoded by motion compensated predictive and transform coding algorithms [16, 17]. Thus an image is described by independently moving square blocks which can lead to visible distortions known as blocking and mosquito effects of low bit rate codecs. To avoid these image distortions more appropriate source models for describing the image have to be introduced. This paper presents a coding method called object-oriented analysis–synthesis coding which subdivides each...
image into moving objects and encodes each object by three sets of parameters defining the motion, shape and colour information of the object.

The objects and their parameters are obtained by an automatic analysis of the actual input image at the coder. Using the encoded and transmitted parameters an image can be reconstructed by image synthesis at the decoder as well as at the coder.

Analysis-synthesis coding techniques have been published in [1, 4, 6, 7, 12, 15, 25]. These techniques have been developed to efficiently encode one known special object, e.g. a human face, and require an algorithm for recognizing the face in a scene. Therefore these techniques are sometimes called knowledge-based analysis-synthesis coding techniques [12]. If the semantical meaning of the face is also considered for coding it might be called semantic coding [4, 7]. In [8] a more general strategy for analysis-synthesis coding is outlined.

The coding concept described here, is not restricted to only one special object and therefore can be applied to a more general class of scenes. It does not require a recognition algorithm. Object-oriented analysis-synthesis coding can be based on different source models for describing the objects and their motion in a scene. Depending on the source model parameter sets with different information content and different bit rates will be generated by the coder. Two source models are considered in this paper: The model of planar rigid objects with three-dimensional motion is assumed.

Instead of the frame memory of block-oriented coding, motion parameter set \( A_i \) of an object \( i \) contains two elements \( d_i, a_i \), corresponding to the frame-to-frame displacement of the object in \( x \)- and \( y \)-direction. The shape parameter set \( M_i \) can be interpreted as picture elements of a bi-level image, where the white picture elements describe the shape of the object being analysed and the black picture elements represent the residual image area. The elements of the colour parameter set \( S_i \) correspond to the luminance and chrominance information of each picture element belonging to the object \( i \). Fig. 2 illustrates the information of these parameter sets for an image with two objects.

In the case of the source model considered here the motion parameter set \( A_i \) of an object \( i \) contains two elements \( d_i, a_i \), corresponding to the frame-to-frame displacement of the object in \( x \)- and \( y \)-direction. The shape parameter set \( M_i \) can be interpreted as picture elements of a bi-level image, where the white picture elements describe the shape of the object being analysed and the black picture elements represent the residual image area. The elements of the colour parameter set \( S_i \) correspond to the luminance and chrominance information of each picture element belonging to the object \( i \). Fig. 2 illustrates the information of these parameter sets for an image with two objects.

The task of the image analysis block in Fig. 1 is to analyse the next input image \( S_{k+1} \) to be coded and to estimate the parameter sets \( A_i, M_i, S_i \) for each object \( i \) by use of the reconstructed image \( S'_i \). A hierarchically structured procedure has been developed which starts with the estimation of \( A_i, M_i, S_i \) of the largest object in the first step of hierarchy and then analyses the smaller objects in the subsequent steps of the hierarchy. At the output of the image analysis block the parameter sets \( A_i, M_i, S_i \) are available in PCM representation as illustrated in Fig. 2.

The analysis fails in image areas which cannot be described by the source model being applied.

In the case of the source model of planar rigid objects with translational motion.

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

Fig. 2. Illustration of model based image analysis. (Source model of planar rigid objects with translational motion.)
The transmission starts with the parameter set of the objects in an order of decreasing priority. Thus, the priority control arranges the parameter sets of the objects in an order of decreasing priority. Consequently, the parameter sets which can properly be synthesized without object-oriented coding techniques have to be transmitted to the receiver. Hence, the introduction of object-oriented coding techniques has to be justified with respect to the resulting overall bit rate. Since the parameter information and the resulting bit rate are dependent on the source model of the coding concept, it is important to develop source models and corresponding image analysis techniques that generate temporal smoothly varying parameter sets which can be encoded efficiently. Using the three parameter sets an image can be reconstructed by image synthesis. The image synthesis is determined by the source model of the image analysis part and realized by an application of the estimated parameter sets. As shown in Fig. 3 the motion parameter \( A \) and the shape parameter \( M \), are encoded by predictive coding techniques using the motion and shape information of the previous frame. In the case of planar rigid objects the shape information describes the silhouette of an object. Therefore, contour coding techniques [5] are applied and only the temporal changes of the silhouette are encoded by predictive coding. The colour information is encoded by hybrid coding techniques, which combine motion compensating prediction with intraframe coding.

While block-oriented hybrid coding techniques [17] transmit only two parameter sets (the motion and colour information of each block), object-oriented analysis-synthesis coding transmits three parameter sets (the motion, shape and colour information of each object). Therefore, the additional bit rate \( R_{\text{S}} \) required for transmitting the shape information \( M \) has to be compensated by a reduction of the bit rates \( R_{\text{A}} \) and \( R_{\text{S}} \) required for motion \( A \) and colour information \( S \) in order to obtain a higher coding gain for object-oriented coding. This is achieved in two ways: First, only one motion parameter set is transmitted for a complete object covering the area of several blocks; secondly, the prediction of the colour information is improved by the use of more appropriate source models. The efficiency of object-oriented coding depends on the size of the objects. Therefore, in the case of small objects, the coder mode control switches back to block-oriented coding. By this control the efficiency of object-oriented coding is always superior or equal to that of block-oriented coding.

The priority control in Fig. 3 allows to identify objects which can properly be synthesized without shape and colour update information, i.e. which may be encoded by motion information only in order to save bit rate. Thus geometrical distortions are introduced instead of quantization errors. Further, the priority control arranges the parameter sets of the objects in an order of decreasing priority. The transmission starts with the parameter set of the highest priority and ends with that parameter set where the bit rate available for the image is exhausted.

The parameter sets, which do not move, are transmitted last. Therefore, the encoded motion, shape and colour parameters are transmitted and decoded at the receiver as well as at the transmitter to reconstruct the PCM representations of \( A, M, S \), which are stored in the memory for object parameters. Using the parameters \( A, M, S \) the actual image can be synthesized.

**3. Image analysis and synthesis based on moving 2D-objects**

According to the explained object-oriented coding concept, the objects of a scene are described by three sets of parameters \( A, M, S \) defining the motion, shape and colour information of the objects. The entropy of each parameter set represents a part of the information which has to be transmitted to the receiver. Hence, the introduction of object-oriented coding techniques has to be justified with respect to the resulting overall bit rate. Since the parameter information and the resulting bit rate are dependent on the source model of the coding concept, it is important to develop source models and corresponding image analysis techniques that generate temporally smoothly varying parameter sets which can be encoded efficiently. Using the three parameter sets an image can be reconstructed by image synthesis. The image synthesis is determined by the source model of the image analysis part and realized by an application of the estimated parameter sets.

One of the essential problems of analysis-synthesis coding is the image analysis part, i.e. to subdivide a scene into moving objects and to describe each object by three sets of parameters defining its motion, shape and colour. From the literature [2, 18, 21, 22, 24] several approaches are known to detect moving objects and to measure their velocity. Most of them are based on the evaluation of the optical flow, i.e. of the measured displacement vector field [2, 18, 24]. The disadvantage of these approaches is that the interdependence between motion and object boundary estimation is not taken into account. For an accurate motion description, the object boundaries have to be known, while for a correct object boundary detection an accurate description of motion is necessary. Hence, motion estimation and object boundary detection have to be treated jointly, as they influence each other: The more accurate the motion estimation, the more exact is the description of the object boundaries, and vice versa. This problem is illustrated in Fig. 4.

In this section, the source model of the image analysis is based on the assumption of rigid, planar objects moving arbitrarily in the three dimensional space. It can be shown [21, 23] that in this case the mapping of a moving object onto the camera image is defined by eight parameters—so called mapping parameters. An image analysis algorithm is presented that formulates the analysis task as a hierarchical application of object motion and boundary estimation [11]. Fig. 5 shows the structure of the algorithm which consists of three parts: the estimation of the eight mapping parameters, the internal image synthesis and the verification test.

Inputs to the image analysis scheme are the luminance signals \( S_i \) and \( S_{i+1} \) of two successive pictures of an image sequence. The prime denotes that \( S_i \) is the transmitted and synthesized version of \( S_i \) in a first step, the starting hypothesis is postulated that the whole scene represents one object, which does not move. Thus, the synthesized picture \( \hat{S}_i \) is identical to the picture \( S_i \). According to the starting hypothesis \( S_i \) and \( S_{i+1} \), should have the same luminance values. The verification test evaluates the frame differences between \( S_i \) and \( S_{i+1} \), and detects that area \( M \), where the hypothesis is true. The area \( M \) represents the object stationary
3.1. Estimation of eight mapping parameters

The considered source model is based on rigid, planar objects moving arbitrarily in the three dimensional space. According to the physics of the camera, each object is mapped onto the camera target by central projection (see Fig. 6). Caused by the three dimensional motion, the mapping of a point \( P' \) on a planar, rigid object is moved in the image plane from \((X',Y')\) to \((X,Y)\). The functional relationship between this pair of image plane coordinates can be described by a formula which depends on the location \((X,Y)\) within the image plane and eight parameters—the so called mapping parameters [23]

\[
(X', Y') = F(a_1, a_2, \ldots, a_5, X, Y) = \left( \frac{a_1 X + a_2 Y + a_3}{a_4 X + a_5 Y + 1}, \frac{a_6 X + a_7 Y + a_8}{a_4 X + a_5 Y + 1} \right).
\]

(1)

These mapping parameters include both, the motion description and the position description of a planar, rigid object and can be estimated from two successive frames [11, 21]. Equation (1) is valid for the set of all mapping points \((X', Y')\) of one moving object.

Accounting all luminance changes due to object motion, the signal description of the mapping results as

\[
S_{j+1}(X, Y) = S_j(X, Y).
\]

(2)

Thus, the relationship between the signal and the mapping parameters can be formulated as follows

\[
S_{j+1}(X, Y) - S_j(X, Y) = S_j(a_1 X + a_2 Y + a_3, a_4 X + a_5 Y + 1, a_6 X + a_7 Y + 1, a_8 X + a_9 Y + 1) - S_j(X, Y) = \text{FD}(X, Y),
\]

(3)

where \( \text{FD}(X, Y) \) is the frame difference of two successive frames to be evaluated.

Using a Taylor series expansion to describe the image luminance in the neighborhood of \((X, Y)\) as a function of the mapping parameters \( a_i \), the frame difference \( \text{FD}(X, Y) \) can be approximated by

\[
\text{FD}(X, Y) = HA,
\]

(4)

where \( H \) is a vector containing weighted local gradients and \( A = (a_1, a_2, \ldots, a_9)^T \) is the mapping parameter vector.

The evaluation of (4) on the set of all mapping points \((X, Y)\) of one moving object, planar object yields the parameter vector \( A \) by linear regression [11, 21].

\[
\text{FD}(\text{obj } i) = \sum_{(X, Y) \in \text{obj } i} \frac{(S_{j+1}(X, Y) - S_j(X, Y))^2}{\text{SIZE}(\text{obj } i)},
\]

(5)

and

\[
\text{FD}(\text{obj } i) = \sum_{(X, Y) \in \text{obj } i} \frac{(S_{j+1}(X, Y) - S_j(X, Y))^2}{\text{SIZE}(\text{obj } i)}.
\]

(6)

3.2. Image synthesis

In each step of the hierarchy of the image analysis scheme, an internal image synthesis has to be performed. For image \( S_{j+1} \), this synthesis is achieved by a motion compensating prediction of all image regions marked in the binary mask \( m_{j+1} \). The addresses of the corresponding picture elements of a position \((X, Y)\) in the prediction image \( \hat{S}_j \) and \((X', Y')\) in the image \( S_j \) are determined for each marked image region with the help of its mapping parameters \( \hat{A} \). The picture element to be predicted results as [11, 21]

\[
\hat{S}_j(X, Y) = \hat{S}_j(X', Y'),
\]

(7)

The verification test controls the coincidence of the estimated and real object motion by means of the mean square displaced frame difference (DDF), because the mapping parameter estimation is based on the minimization of this signal parameter [11, 21]. For each object \( i \) of the \( j \)th step of hierarchy, the mean squared DDF(\text{obj } i) and the mean squared frame difference DDF(\text{obj } i) are evaluated on all image coordinates \((X, Y)\) belonging to the mapping of the actually considered object \( i \).
where $S_k$ is the transmitted and synthesized version of $S_k$, $S_{k+1}$ is the luminance signal of image $k+1$, $S_k$ is the prediction image of the $j$th step of hierarchy to describe $S_{k+1}$, and SIZE $(obj)$ is the area of the actually considered object $i$ in units of picture elements.

Then, the validity of the mapping description is determined by

$$DFD_i^j(\text{obj}) / F D_j^i(\text{obj})$$

\[ \begin{cases} < T_j & : \text{the mapping description of the object is valid} \\ \geq T_j & : \text{the mapping description is insufficient} \end{cases} \quad (7) \]

If the value of the quotient in (7) exceeds a given threshold $T_j$, the minimization achieved by the algorithm is insufficient, i.e., a mapping description of the moving object $i$ according to the model assumptions of the algorithm is not possible. Moving objects whose mapping description is insufficient are marked as special objects.

If the value of the quotient in (7) falls below the threshold $T_j$, the mapping description of the object $i$ is accepted. By means of the accepted mapping description, uncovered image regions are separated from the mapping of the moving object $i$ [11]. If there are moving objects in front of the actually considered object, the mapping description is only valid in a part of the object. For that reason, the verification test yields additionally a detection of those objects which are not correctly described in their mappings. These objects represent the objects to be analyzed in the next step of hierarchy [11].

3.4. Improvement of image analysis by object tracking

In an analysis-synthesis coder concept, the estimated object shapes are part of the information to be transmitted. To guarantee an efficient coding of the object shapes, the shapes may vary only smoothly from image $k$ to image $k+1$ in order to be predictable by the source model. In the case of planar, rigid objects the silhouette information of an object has to be considered. The object silhouettes of the presented image analysis algorithm can vary temporally due to the fact, that the identification of motion in uniform areas is unreliable.

The image analysis algorithm is based on the assumption that motion of objects generates luminance changes in the image. In uniform areas, it is difficult to distinguish between luminance changes which are caused by noise or by object motion. In order to achieve a more reliable result, the objects are tracked, i.e., the image analysis results of the actual image are used as initial guesses for the image analysis of the next picture. In this way, ambiguity is exploited to give consistent object shapes which are only changing if this is necessary because of apparent model violations.

To illustrate the function of the hierarchical image analysis algorithm, Fig. 7 shows the analyzed areas of the first, second and third level of hierarchy and their superposition which have been calculated from two pictures of the test sequence "Miss America" provided by the Specialist Group for Visual Telephony in CCITT SG XV. In Fig. 7 uncovered image regions and the segmented objects are displayed with white and gray luminance values.

4. Image analysis and synthesis based on moving 3D-objects

Whereas in Section 3 considerations are restricted to moving 2-D objects, in this chapter the source model is extended to consider 3-D objects in the real world in order to improve the coding efficiency. Fig. 8 shows the relation between the three-dimensional real world—a view of this world called real image—and a three-dimensional artificial model world with its projection onto the image plane called model image. The synthesized model image approximates the real image. Goal of the modelling is to generate a model world which has a model image identical to the real image.
A world is described by a scene, its illumination and its camera. A scene consists of objects and relationships between them. The applied modelling algorithm [14] requires diffuse illumination and diffuse reflecting surfaces for the model world. Furthermore, the objects have to be opaque and quasi-rigid. The real world in Fig. 8 contains one real object, i.e. a pyramid. The real image of the camera shows only the front face of the pyramid, i.e. an octagon. From this reference image the initial model world is generated without any a priori knowledge. The image analysis algorithm described in Section 3 computes silhouettes of moving objects. From a silhouette a model object is generated. It is represented by a mesh of triangles, i.e. a wire-frame. A silhouette computed from the test sequence "Claire" and a wire-frame describing the model object shape by triangles are shown in Fig. 9.

Since one image does not give any depth information, the initial three-dimensional shape of the object in the model world is estimated. Similar to the source model with 2D-objects the shape is assumed to be a planar patch. The example in Fig. 8 shows that the projections from real and model world can be identical, although their three-dimensional contents differs significantly. If the pyramid is moved, other faces than the front face will be seen in the real image. Hence, the initial guess of the model world to represent the pyramid by a planar patch will generate a difference between the new real image and the model image showing the motion compensated model object. This difference can be analysed and used for adapting the three-dimensional shape of the model object to the real world pyramid, i.e. for adapting the depth information.

In order to define the surface colours of the objects, the reference image is projected onto the initial model world. Hence the surface colours of each visible triangle of an object are defined.

If the same model world is generated at the coder and decoder, only changes of illumination and object shape, object motion, or new objects entering the scene have to be transmitted. In addition to the source model of 2D-objects, this source model based on 3D-objects allows the effective modelling of rotation of 3D-objects. Whereas the motion description of 2D-objects is implicit, i.e. the meaning of each motion parameter is not related to our imagination of real movement, the use of 3D-objects allows an explicit description of motion and shape. In coding applications this eases coder control, because every motion and shape parameter can easily be interpreted. The gain of this effective modelling of rotation and of the explicit parameter description has to be paid for. As Fig. 8 indicates, using a source model of 3D-objects for image analysis causes the additional problem of determining the 3D-shape of the objects. This requires two steps, in a first step the motion and in a second step the three-dimensional shape of an object has to be estimated.

Image synthesis is based on the same source model and is solved by means of central projection and the Z-buffer algorithm [19].

### 4.1. Image analysis

Image analysis is based on the new real image \( S_{k+1} \) of the sequence and on the current model world \( P \), which is the a priori knowledge acquired from previous images. By evaluating the frame difference between the image \( S_{k+1} \) and the model image \( P \), an update of the model world is computed. The goal of the updating is to minimize the mean square difference (MSD) between the model and the real image. According to the source model, frame differences are due to motion of objects or insufficient information about object shape, i.e. there are two causes for frame differences. Since a change of motion as well as a change of shape parameters in the model world influence the frame difference between model and real image, the cause of frame differences can not always be determined uniquely.

In order to get a stationary model world, which is very important with respect to coding, it is assumed that the frame differences are mainly caused by motion of the objects. Only frame differences which can not be compensated by motion are assumed to be due to insufficient information about the object shape.

Fig. 10 shows the structure of the image analysis algorithm for extracting the motion, shape and colour parameters of an object and for updating the model world. In contrast to Fig. 5 input of the algorithm are the model world \( P \) and the new real image \( S_{k+1} \). Output of the motion analysis are the motion parameter sets \( A_i \) of all objects \( i \) of the model world and the updated model world \( P \). By minimizing the MSD between the real image \( S_{k+1} \) and the model image \( P \), the shape analysis computes an update of the object shapes. Results are the shape parameter sets \( M_i \), the colour parameter sets \( S_i \), and the model world \( P_i \), whose model approximates \( S_{k+1} \).

#### 4.1.1. Motion analysis

Motion analysis will be described here using Fig. 11.

The differential estimation algorithm [13] computes the motion parameters necessary for motion compensating the objects of a model world \( P \). Input to this algorithm are the current model world \( P \), which equals \( P \) for the first iteration, the new real image \( S_{k+1} \), and the object area \( O_i \), which defines the area of the image plane, in which motion parameters have to be estimated. In contrast to Fig. 5 only one object is considered for each iteration.

For each object \( i \) a motion parameter set \( A_i = (T_i, T_i', T_i, R_i, R_i, R_i) \) describing the translation \( T \) and rotation \( R \) of the model object is estimated.

Using \( A_i \) and the model world \( P_i \), a motion compensated model image \( \tilde{S}_i \) is computed by means of image synthesis.

The verification test looks at the actual frame difference between the current image \( S_{k+1} \) and the
model image $\hat{S}_k$ computed from the current model world $\hat{P}_k$, and the motion parameters $A_i$ and compares it to the frame differences between $S_{k+1}$ and the model image of the model world $\hat{P}_k$. If the frame difference decreases in the area of $O_i$, the model world $\hat{P}_k$ is updated to $\hat{P}$, otherwise the motion parameters are rejected and if necessary object area $O_i$ is subdivided into two areas [3].

The loop consisting of estimation of motion parameters, image synthesis and verification test inside the motion analysis block is executed until no substantial decrease of the MSD between the model and real image is achieved.

4.1.2. Shape analysis

The described motion analysis deals with objects moving as a whole. Shape analysis, on the other hand, individually moves vertices of the mesh of triangles (control points) of an object, in order to further decrease the MSD between the real and model image. To simplify the optimization problem, lines of perspective [13] are used (Fig. 12).

To each control point of a model object one line of perspective is assigned. A line of perspective is a ray starting at its control point in the direction of the point of focus of the camera. The line of perspective assigned to a control point when the initial model world is built up is the reference line of perspective of the reference image. The motion parameters determined for an object are applied to each control point and to its reference line of perspective. Hence, the motion of a triangle can also be interpreted as a view from another position.

Fig. 13 shows part of an object wire-frame and how it changes when one control point is moved on its reference line of perspective. If the control point is moved on its reference line of perspective (see Fig. 12) there will be no changes in the view of the model world from the reference position but views from other positions will change. The dominant influence of the first image is justified because all luminances of the model world are extracted from the first image.

Shape analysis (Fig. 14) starts with the estimation of shape parameters. Input to this algorithm are the actual model world $\hat{P}_k$, which in the first iteration equals the output $\hat{P}_k$ of the motion analysis part, and the current image $S_{k+1}$. The difference signal between the image of the model world $\hat{P}_k$, and image $S_{k+1}$ is evaluated and for the control points of object $i$ the parameter set $M_i = (A_1, \ldots, A_N)$ containing one shape parameter $A_N$ for each control point is estimated. A shape parameter $A_N$ determines the shift of control point $P_i(U', V', W')$ on its reference line of perspective. Hence, the line of perspective $P_i(U', V', W')$ moves as a function of its reference line of perspective vector $L_i$ and its shape parameter $A_N$ to $P_i(U, V, W)$:

$$P_i(U, V, W) = P_i(U', V', W') + L_i A_N$$  \[8\]
The shape parameters are computed using a simple search method [13].

The verification test simply evaluates the MSD between $S_{k+1}$ and the synthesized model image $\tilde{S}$ of $\tilde{\mathbf{P}}_{k+1}$ and $\mathbf{M}_{k+1}$. In areas where the MSD between $\tilde{S}$ and $S_{k+1}$ decreased, the shape parameters $\Delta_k$ are accepted. Then the algorithm decides, whether to iterate the same object or to optimize another object. The process stops as soon as there is no substantial improvement in image quality.

4.2. Properties of image analysis based on moving 3D-objects

The chosen approach of first estimating motion and then estimating shape is essential for coding applications. First motion analysis estimates motion parameters for each object area and, if necessary, decomposes an object area into flexibly connected sub-objects. Afterwards shape analysis adapts model object shape to the shape of the real objects visible in the sequence. Hence, this image analysis yields a subdivision of a scene into objects defined by their motion, shape and colour parameters.

As an illustration of the image analysis based on arbitrarily moving three-dimensional objects Fig. 15 shows two frames of the test sequence "Claire" and results of the image analysis. The two frames (Figs. 15(a) and (b)) are used to generate a model world and to estimate the parameter sets for motion, shape, and colour. Fig. 15(c) shows the wire-frame of the model object. The object was automatically subdivided into two sub-objects indicated by different gray levels in the wire-frame. Figs. 15(d) and (f) show frame differences between the model image and the actual real image of Fig. 15(b). Fig. 15(d) shows the frame difference after motion compensation and Fig. 15(e) after motion compensation and shape adaptation. In order to demonstrate the advantage of three-dimensional object models over two dimensional object models, the frame difference using a two dimensional model object is displayed in Fig. 15(f).

5. Parameter coding

Parameter coding in an analysis-synthesis coder includes the coding of the motion, shape and colour parameters, the control of the coder modes and the priority control of the parameter transmission as shown in Fig. 3. The following explanations refer to an analysis-synthesis coder which is based on a source model with 2D-objects.

5.1. Coding of the motion, shape and colour parameters

The three parameter sets $A_i$, $M_i$, $S_i$ defining motion, shape and colour information for each object $i$ are generated by image analysis in PCM representation. In order to increase the coding efficiency additional individual parameter coding techniques are applied. Motion parameters are coded by known DPCM techniques using stored motion parameters from the previous image. Shape parameters of an object are coded by a contour coding algorithm [10] in combination with predictive coding. Only the temporal differences of the object boundaries are coded using stored shape parameter information of the corresponding object from the previous image. The transmission of the shape update information is suppressed by setting the temporal differences to zero if the shape of an object has not changed significantly. In this case the object shape remains unchanged. Colour information is coded with a hybrid scheme combining motion compensating prediction and intraframe coding techniques. For motion compensating prediction of the colour information a prediction image is synthesized using the previous transmitted image and the actual motion and shape information of the objects. Therefore, motion and shape information of an image have to be transmitted first. A new intraframe transform coding technique [9] is applied which is not restricted to a block structure but can be adapted to object boundaries. At present, for intraframe coding of the prediction error large objects are decomposed into internal square blocks of
picture elements and boundary blocks whereby the boundary blocks are split into segments considering the shape information. The resulting coded information \( R_i \) of each object \( i \) consists of three components \( R_{si}, R_{m}, R_q \), representing motion, shape and colour information.

### 5.2. Coder mode control

The object-oriented coding concept offers the possibility of an object dependent control of the parameter coding. For this reason, the image analysis indicates to the coder mode control whether the object considered is a special object or a normal object and the coder mode control analyses the size of each object. Based on these informations, the coder mode control decides which parameter sets \( A_j, M_i, S_i \) of an object have to be transmitted and whether or not the motion compensating prediction is applied, i.e. whether intraframe coding is used instead of hybrid coding. The coder mode control distinguishes between four classes of objects and adapts the parameter sets to these classes (see Table 1).

In the case of normal objects, motion and shape information is transmitted by encoded prediction errors. The motion information is omitted for special objects which can not be described by the source model. In addition, the coder control informs the hybrid coder to switch off the motion compensating prediction such that only intraframe coding is applied. Also, in the case of small objects, the motion information is omitted since transmission of motion information is inefficient. Again, the coder control informs the hybrid coder to switch off the motion compensating prediction such that only intraframe coding is applied. Additionally, the coder control decides for small objects whether or not it is worthwhile transmitting the shape information. If not, the coder control generates a square shape \( M \) and thus reduces the object-oriented coding to block-oriented intraframe coding without transmitting shape information of the object. This decision is based on the following rule

\[
\text{SIZE}(obj_i) = \begin{cases} 
[129, 256]: & \text{Block-oriented intraframe coding (small objects)}, \\
[1, 128]: & \text{Object-oriented intraframe coding with transmission of shape information (small objects)}. 
\end{cases} 
\tag{10}
\]

where the size of object \( i \) is measured by the number of pixels covered by its silhouette.

Thus, the coder mode control can avoid a possible decrease of the coding gain below that of block-oriented intraframe coding.

The coder mode control has to be transmitted for each object.

#### Table 1

<table>
<thead>
<tr>
<th>Coder modes</th>
<th>Normal objects</th>
<th>Special objects</th>
<th>Small objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter sets to be transmitted</td>
<td>( A_j )</td>
<td>( M_i )</td>
<td>( S_i )</td>
</tr>
<tr>
<td>Motion parameters</td>
<td>( x )</td>
<td>( x )</td>
<td>( x )</td>
</tr>
<tr>
<td>Shape parameters</td>
<td>( x )</td>
<td>( x )</td>
<td>( x )</td>
</tr>
<tr>
<td>Colour parameters</td>
<td>Hybrid coding</td>
<td>Intraframe coding</td>
<td>Intraframe block-coding</td>
</tr>
</tbody>
</table>

#### 5.3. Priority control

The priority control arranges the parameter sets \( A_j, M_i, S_i \) to be transmitted in an order of decreasing priority for each image. To the parameter sets \( A_j, M_i, S_i \) of each object the highest priority is assigned because these parameters are required for image synthesis in both cases that colour information \( S_i \) is transmitted or is omitted. This includes parameter sets \( M_i \) where the shape update information has been set to zero.

The priority \( PR(S_i) \) of a parameter set \( S_i \) of an object \( i \) is determined by object features as the object size \( O_i \) and the quality of object synthesis \( O_{q_i} \).

By the normalization of the object features \( O_i \), \( O_{q_i} \),

\[
O_i = \text{SIZE}_\text{min}/\text{SIZE}(obj_i) \quad \tag{11}
\]

and

\[
O_{q_i} = \text{MADFD}(obj_i)/\text{MADFD}_\text{max} \quad \tag{12}
\]

the object features are in the range \([0, 1]\). High values of \( O_i, O_{q_i} \) indicate high priority. \( \text{SIZE}_\text{min} \) denotes the smallest size of the mapped objects of the scene and \( \text{SIZE}(obj_i) \) the size of the actually considered, mapped object \( i \). The parameter \( O_{q_i} \) evaluates the object oriented success of the prediction of an object. \( \text{MADFD}_\text{max} \) describes the maximum mean absolute displaced frame difference of all objects of an image and \( \text{MADFD}(obj_i) \) describes the mean absolute displaced frame difference of the actually considered object \( i \).

Thus, the priority \( PR() \) of the parameter sets \( A_j, M_i, S_i \) results as

\[
\begin{align*}
PR(A) & = \max(PR(S)), \\
PR(M) & = \max(PR(S)), \\
PR(S) & = f(O_i, O_{q_i}) \quad \tag{13}
\end{align*}
\]

where \( f() \) denotes the functional relationship of the priority \( PR(S) \) and its arguments \( O_i \) and \( O_{q_i} \).

#### Example of the priority control

- **Object 1**
  - \( O_i = 0.1 \)
  - \( O_{q_i} = 0.1 \)
  - \( O_{q_i} = 1 \)

- **Object 2**
  - \( O_i = 1 \)
  - \( O_{q_i} = 0.1 \)

First, the parameter sets of highest priority, in this example the motion and shape information of object 1 and the shape information of object 2, are transmitted. Then, the following parameter sets are transmitted, i.e. the colour parameter set of object 2 and the colour parameter set of object 1. The transmission of the parameter sets ends if no more transmission rate is available. If, for example, the transmission rate is exhausted before the transmission of the colour parameter of object 2, geometrical distortions are introduced for object 1 instead of quantization errors.

In the case of large objects which are decomposed into internal blocks and boundary blocks the colour information is encoded separately for...
each block. Hence, an additional parameter priority \( PR_p \) is introduced which is determined by local features of a block as the mean absolute prediction error \( B_p \) and blocks with local object boundaries \( B_s \).

The priority \( PR_p \) results as

\[
PR_p = g(B_p, R_s),
\]

where \( g(\cdot) \) denotes the functional relationship of the priority \( PR_p \) and its arguments \( B_p \) and \( R_s \).

By the normalization of the block features \( B_s, B_p \),

\[
B_s = \begin{cases} 1, & \text{boundary block}, \\ 0, & \text{else}, \end{cases}
\]

\[
B_p = \frac{\text{MADFD}(\text{block})}{\text{MADFD max}},
\]

all block features are within the range \([0, 1]\), where high values of \( B_s, B_p \) indicate high priority. The control parameter \( B_s \) denotes boundary blocks which have to be split into segments. These blocks have high priority, the segments are separately encoded. The control parameter \( B_p \) evaluates the local success of the prediction in a block. MADFD max describes the maximum mean absolute displaced frame difference of all blocks of the image and MADFD(block) describes the mean absolute displaced frame difference of the actually considered block.

Thus, the priority of a block \( PR_{block} \) based on these normalized object and block features results in

\[
PR_{block} = h(\text{PR}(S), \text{PR}_p),
\]

where \( h(\cdot) \) denotes the functional relationship of the priority \( PR_{block} \) and its arguments \( \text{PR}(S) \) and \( PR_p \).

6. Experimental results

The described analysis-synthesis coder with a simplified parameter coder has been experimentally investigated by means of computer simulations at 64 kbit/s transmission rate. The CCITT test sequence “Miss America” has been used with about 300 bit is required. Again assuming 3 to 5 moving objects of different sizes the total amount for the shape information results in about 900 to 900 bit per image.

At present, the colour information is coded with a hybrid coding technique using object-oriented image synthesis for prediction and macro blocks for intraframe transform coding of the prediction error signal [20]. Each macro block consists of one luminance block of \( 16 \times 16 \) pixels and two chrominance blocks of \( 8 \times 8 \) pixels.

The coder mode control is reduced to use only mode 1 and mode 2 according to Table 1.

The transmission is organized via a priority control where the motion \( A \) and shape \( M \) information are transmitted with highest priority requiring about 1000 bit per image. Thus motion and shape information require a transmission bit rate which can be compared with that of the side information of a block-oriented hybrid coder. After coding of shape and motion parameters, the remaining bit rate of about 5000 bit per image is available for updating the colour information.
when colour transmission is omitted. This example demonstrates that, with the same amount of side information, object-oriented coding generates prediction images that do not show the artefacts of block-oriented hybrid coding although there is no great difference in the mean square prediction error of both coding techniques. To take advantage of the more natural prediction image of the object-oriented coder the priority control for transmitting the colour information has to consider sophisticated criteria which are more complex than the mean square error.

In this implementation, the transmission of the colour information is organized in macro blocks by the priority control. A priority \( \text{PR}(S) \) of a parameter set \( S \) according to

\[
\text{PR}(S) = f(O_u, O_v) = (1 + w_u O_u) (1 + w_v O_v)
\]

and a priority \( \text{PR}_p \) according to

\[
\text{PR}_p = g(B_u, B_v) = (1 + w_u B_u) (1 + w_v B_v)
\]

are used. Thus, the priority of a macro block \( \text{PR}_{\text{block}} \) based on these normalized object and block features results in

\[
\text{PR}_{\text{block}} = h(\text{PR}(S), \text{PR}_p) = \text{PR}(S) \times \text{PR}_p = (1 + w_u O_u) (1 + w_v O_v) \times (1 + w_u B_u) (1 + w_v B_v).
\]

If all weighting factors are equal to 1, a macro block achieves a high priority, if it belongs to an object of small size or to an object which can not be synthesized properly. Further, a macro block gets high priority, if the macro block contains object boundaries or if the local prediction in a macro block is inefficient.

To demonstrate the influence of the priority control \( \text{PR}_{\text{block}} \), the coding of large prediction errors and boundary blocks have jointly been accentuated by choosing \( w_u = 0, w_v = 0, w_u = 20 \) and \( w_v = 1 \) (Fig. 21). By this weighting, the so called "mosquito effects" which are typical for low bit rate hybrid coders in contour areas can be avoided since even small prediction errors of high visibility are coded. Furthermore, all blocks of high prediction error, e.g. the blocks which contain the mapping of the face are coded.

7. Conclusion

The structure of an object-oriented analysis-synthesis coder is described which segments an image into objects and encodes each object by three parameter sets. Algorithms are presented which allow an automatic estimation of the parameter sets based on a source model of either 2D- or 3D-objects. For the case of 2D-objects the parameter coding is described. First results obtained with a simplified version of the described parameter coding are discussed.

The object-oriented analysis-synthesis coder is able to synthesize images for the prediction of colour information which look more natural than the images predicted by a block-oriented hybrid coder although the mean square prediction error may be the same or even higher. In the case of low bit rate coding where the available bit rate does not allow a perfect coding of all prediction errors, a block-oriented hybrid coder generates known quantization errors while an object-oriented coder can suppress the updating of the colour information of an object to generate geometrical distortions instead of quantization errors. By transmitting the motion and shape information the object is reproduced in the right position but due to the changed projection of the object onto the camera target the non-updated colour information may lead to a loss of local resolution and locally incorrect perspective view at object boundaries. These distortions are less annoying than quantization errors. Further, the object-oriented priority control of the parameter coder allows to improve the coding of important object features. It is possible e.g. to encode small objects like eyes more accurately than the other areas. In addition, the knowledge about the object boundaries can be used to improve the colour gain of the intriframe transform coder.

At present the parameter coding is still in a very early stage. Object-oriented analysis-synthesis coding provides many new possibilities for encoding the colour update information which have to be investigated in the future. There is a need for more appropriate criteria to control the colour parameter transmission since the mean square error criterion is inadequate for a quality assessment of geometrical distortions.

Furthermore, the parameter coding for the case of 3D-objects has to be developed to recognize the advantage of a source model with 3D-objects compared to a source model with 2D-objects.

Acknowledgement

This work has been supported by the Forschungsinstitut der Deutschen Bundespost beim Fernmelde¨technischen Zentralamt. We would like to express our thanks to Professor Gallenkamp for encouraging this work and to Dr. Stenger, Dr. Geuen, Dr. Billiotet-Hoffmann and Mr. Heipel for many helpful discussions.

References


Signal Processing: Image Communication


