REPRINTED FROM:

SIGNAL PROCESSING

IMAGE COMMUN

Volume 1, No. 2, October 1989

OBJECT-ORIENTED ANALYSIS–SYNTHESIS CODING O

Hans Georg MUSMANN, Michael HÖTTER and Jörn OSTERMAN

Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Ha
Hannover 1, Fed. Rep. Germany

pp. 117–138

REPRINTED FROM:

SIGNAL PROCESSING

IMAGE COMMUNICATION

Volume 1, No. 2, October 1989

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

ELSEVIER, AMSTERDAM – OXFORD – NEW YORK – TOKYO

Signal Processing: *Image Communication* 1 (1989) 117-138
Elsvier Science Publishers B.V.

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.

Zusammenfassung. Es wird ein objektorientierter Analyse-Synthese Coder vorgestellt, der Objekte anstelle von Blöcken der Größe $N \times N$ Bildpunkte codiert. Die Objekte werden durch drei Parametersätze beschrieben, die die Bewegung, die Berandung und die Farbe eines Objektes definieren. Die Parametersätze werden durch eine Bildanalyse gewonnen, die entweder auf dem Quellenmodell bewegter 2D-Objekte oder bewegter 3D-Objekte basiert. Für die Codierung der Parametersätze werden bekannte Codiertechniken benutzt. Eine objektabhängige Parametercodierung gestattet es, geometrische Verzerrungen anstelle von Quantisierungsfehlern einzuführen. Mit Hilfe der übertragenen Parametersätze kann ein Bild durch modellgestützte Bildsynthese rekonstruiert werden.

Experimentelle Ergebnisse, die mit einer ersten Implementierung des Coders erzielt wurden, werden vorgestellt und diskutiert.

Résumé. Ce texte porte sur un codeur d'analyse-synthèse encodant les objets plutôt que les blocs $N \times N$ constituant l'image. Les objets sont décrits par trois ensembles de paramètres décrivant leur trajectoire, leur forme et leur couleur. Ces ensembles sont obtenus par une analyse de l'image basée sur des modèles de source d'objets 2D ou 3D en mouvement. Des techniques de codage connues sont utilisées pour encoder ces ensembles. Ce codage par objets introduit des distorsions géométriques au lieu d'erreurs de quantification. A l'aide des ensembles de paramètres transmis une image peut être reconstruite par synthèse basée sur modèle.

Des résultats expérimentaux obtenus lors d'une première mise en oeuvre du codeur sont présentés et commentés.

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 and the model of three-dimensional rigid objects with three-dimensional motion. The colour information is considered as being projected on the surface of the objects. Image analysis algorithms have been developed which estimate the shape and motion of the objects in a scene and automatically generate the three sets of parameters to be encoded by predictive or transform coding techniques. To reduce the frame-to-frame redundancy, motion compensated interframe prediction is applied and only the temporal update information is encoded.

In object-oriented analysis–synthesis coding the object-oriented information can be used to control the coding of the parameter sets. Thus by suppression of shape and colour update information of an object, geometrical distortions are introduced instead of quantization error distortions when irreversible coding techniques have to be applied in order to cut down the generated bit rate. The geometrical distortions are less annoying than quantization error distortions in situations where the modelling of an object is sufficiently exact as indicated by relatively small shape and colour update informations.

In Section 2 the new coding concept and the basic block diagram of the coder are explained. The image analysis and synthesis algorithms are described in detail in Sections 3 and 4. The coding of the parameter sets is still in a very early stage as presented in Section 5. First results obtained by a simulation of this coding concept are discussed in Section 6.

2. Concept and structure of the object-oriented analysis–synthesis coder

Object-oriented analysis–synthesis coding subdivides each image of a sequence into moving objects and describes each object i by three sets of parameters defining motion A_i, shape M_i and colour S_i information of the object. The object parameters depend on the kind of source model being applied. To explain the concept and structure of the object-oriented analysis–synthesis coder the block diagram of Fig. 1 is used and the simplified model of planar rigid objects with only translational motion is assumed.

Instead of the frame memory of block-oriented coding techniques, object-oriented coding requires a memory to store the parameter sets $A=\{A_i\}$, $M=\{M_i\}$, $S=\{S_i\}$ of the objects. The object memory of the coder and decoder contains the same parameter informations and allows the coder and decoder to reconstruct a transmitted image by image synthesis. The reconstructed image S'_k is displayed at the decoder and used for image analysis of the next input image S_{k+1} at the coder.

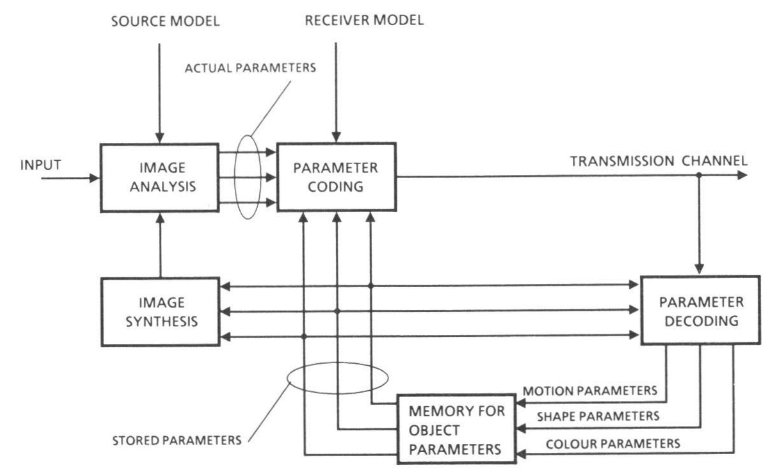

Fig. 1. Block diagram of an object-oriented analysis–synthesis coder.

In the case of the source model considered here the motion parameter set A_i of an object i contains two elements a_1, a_2 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'_k. 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.

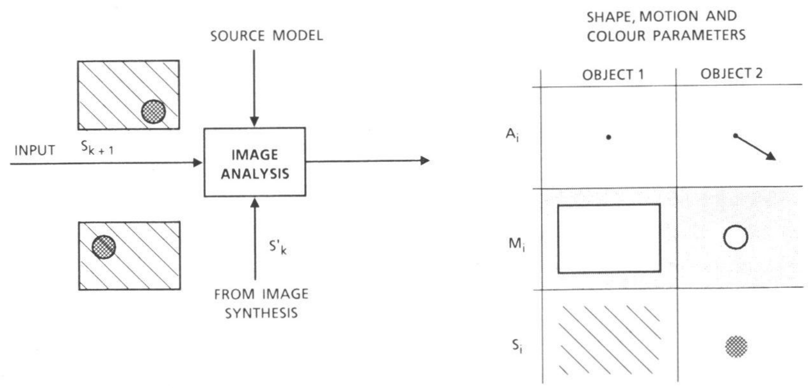

Fig. 2. Illustration of model based image analysis. (Source model of planar rigid objects with translational motion.)

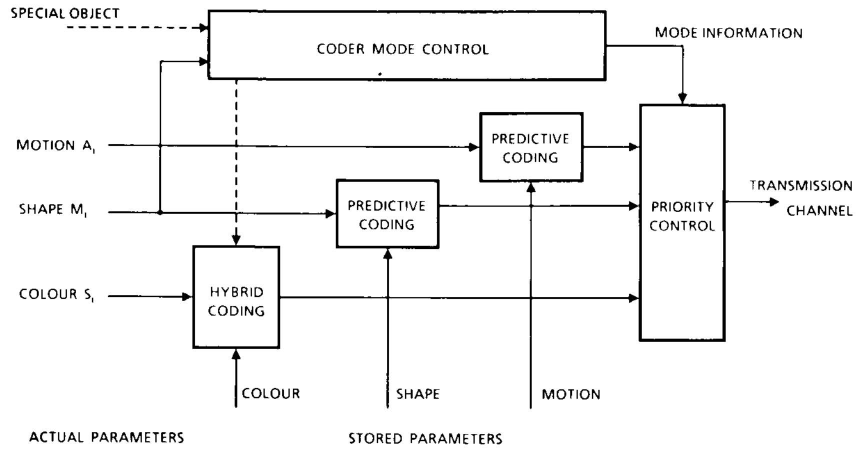

Fig. 3. Parameter coding.

Therefore the success of the image analysis is checked by a verification algorithm. Those areas which cannot be described by the model are marked in a final step of the hierarchical analysis procedure and are denoted as special objects.

The parameter sets A_i, M_i, S_i of each object i have to be encoded efficiently in order to achieve a coding gain when compared to block-oriented coding. As shown in Fig. 3 the motion parameter A_i and the shape parameter M_i are encoded by predictive coding techniques using the motion and shape information of the previous frame k. 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_M required for transmitting the shape information M has to be compensated by a reduction of the bit rates R_A and R_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 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_i, M_i, S_i 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 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.

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

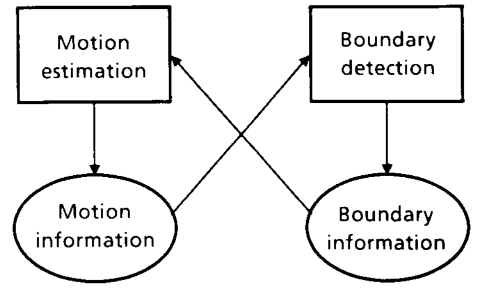

Fig. 4. Illustration of the interdependence of motion estimation and boundary detection of moving objects.

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'_k and S_{k+1} of two successive pictures of an image sequence. The prime denotes that S'_k is the transmitted and synthesized version of S_k. In a first step, the starting hypothesis is postulated that the whole scene represents one object, which does not move. Thus, the synthesized picture $\hat{S}_0$ is identical to the picture S'_k. According to the starting hypothesis S'_k and S_{k+1} should have the same luminance values. The verification test evaluates the frame differences between S'_k and S_{k+1} and detects that area M_1 where the hypothesis is true. The area M_1 represents the object stationary

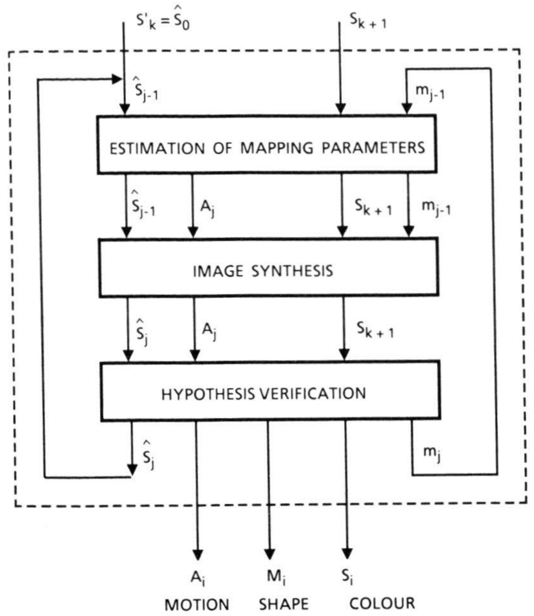

Fig. 5. Image analysis. Model: 3D-motion of planar rigid objects. (S_{k+1}, luminance signal of image $k+1$; $\hat{S}_j$, synthesized image to describe S_{k+1}; m_j, areas to be analysed; A_j, mapping parameters of analysed areas; j, index of hierarchy; i, object i.)

background. Areas with non-zero frame differences indicate changed image regions where the hypothesis is not true. Each disjunct changed image region is then interpreted as the mapping of an additional object which will be analysed in its motion and boundaries in the next step of hierarchy. Therefore, these regions are marked in a binary mask m_1. In the second step of hierarchy, for each marked area of m_1, i.e. for each disjunct, changed image region, one set of eight mapping parameters is calculated by means of the synthesized image $\hat{S}_1$ and the actual image S_{k+1}. Then, for each marked area of m_1, an image synthesis is performed. If there are moving objects in front of moving objects, the assumption, that each changed region represents the mapping of only one moving object is not fulfilled and hence the mapping description is only valid in a part of the region. For that reason, the hypothesis verification yields again a detection of those image parts which have not correctly been synthesized. These image parts represent the mappings of the objects to be analysed in the third step of hierarchy and are marked in the binary mask m_2. Thus, the description of the image S_{k+1} is hierarchically refined until all objects are described in their mappings and their boundaries.

First, the blocks of the image analysis scheme are described in detail, then some important features of the whole analysis-synthesis scheme are discussed.

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]

$$(X', Y') = F\{a_1, a_2, \ldots, a_8, X, Y\}$$
$$= \left(\frac{a_1X + a_2Y + a_3}{a_7X + a_8Y + 1}, \frac{a_4X + a_5Y + a_6}{a_7X + a_8Y + 1}\right). \tag{1}$$

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

$$S_{k+1}(X, Y) = S_k(X', Y'). \tag{2}$$

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

$$S_{k+1}(X, Y) - S_k(X, Y)$$
$$= S_k(X', Y') - S_k(X, Y)$$
$$= S_k\left(\frac{a_1X + a_2Y + a_3}{a_7X + a_8Y + 1}, \frac{a_4X + a_5Y + a_6}{a_7X + a_8Y + 1}\right)$$
$$- S_k(X, Y)$$
$$= \mathrm{FD}(X, Y), \tag{3}$$

where $\mathrm{FD}(X, Y)$ 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 $\mathrm{FD}(X, Y)$ can be approximated by

$$\tilde{\mathrm{FD}}(X, Y) = \boldsymbol{H}\boldsymbol{A}, \tag{4}$$

where $\boldsymbol{H}$ is a vector containing weighted local gradients and $\boldsymbol{A} = (a_1, a_1, a_2, \ldots, a_8)^{\mathrm{T}}$ is the mapping parameter vector.

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

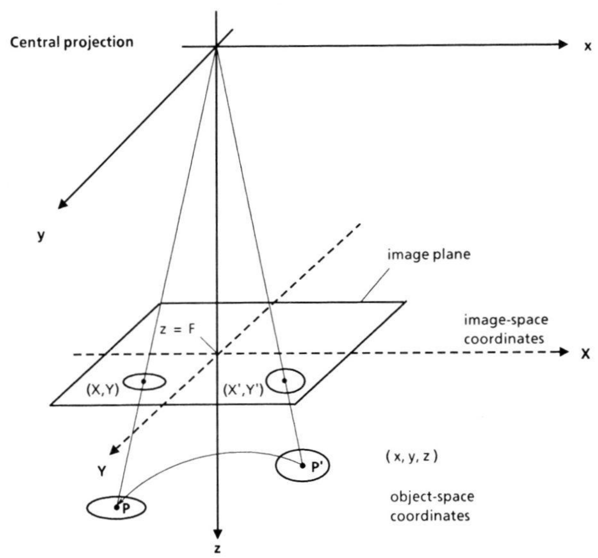

Fig. 6. Imaging of moving planar rigid objects.

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 $\hat{S}_{j-1}$, 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 $\hat{S}_{j-1}$ are determined for each marked image region with the help of its mapping parameters $\boldsymbol{A}_i$. The picture element to be predicted results as [11, 21]

$$\hat{S}_j(X, Y) = \hat{S}_{j-1}(X', Y'), \tag{5}$$

with (X', Y') according to (1) and $(a_1, a_2, \ldots, a_8)^{\mathrm{T}} = \boldsymbol{A}_i$ the estimated mapping parameters of the marked image region considered.

This image synthesis is repeated for each step of hierarchy within the image analysis scheme until all objects to be analysed are motion compensated.

3.3. Verification test

With help of the verification test the success of the analysis scheme is checked for each object i. The verification test controls the coincidence of the estimated and real object motion by means of the mean square displaced frame difference (DFD), because the mapping parameter estimation is based on the minimization of this signal parameter [11, 21]. For each object i of the jth step of hierarchy, the mean squared DFD(obj i) and the mean squared frame difference FD(obj i) are evaluated on all image coordinates (X, Y) belonging to the mapping of the actually considered object i

$$\overline{\mathrm{DFD}^2}\,(\mathrm{obj}\ i)$$
$$= \sum_{(X, Y)\in \mathrm{obj}\ i} \frac{(S_{k+1}(X, Y) - \hat{S}_j(X, Y))^2}{\mathrm{SIZE}\,(\mathrm{obj}\ i)}$$

and

$$\overline{\mathrm{FD}^2}\,(\mathrm{obj}\ i)$$
$$= \sum_{(X, Y)\in \mathrm{obj}\ i} \frac{(S_{k+1}(X, Y) - S'_k(X, Y))^2}{\mathrm{SIZE}\,(\mathrm{obj}\ i)}, \tag{6}$$

where S'_k is the transmitted and synthesized version of S_k, S_{k+1} is the luminance signal of image $k+1$, $\hat{S}_j$ is the prediction image of the jth step of hierarchy to describe S_{k+1} and SIZE (obj i) is the area of the actually considered object i in units of picture elements.

Then, the validity of the mapping description is determined by

$$\overline{\mathrm{DFD}^2}\,(\text{obj } i)/\overline{\mathrm{FD}^2}\,(\text{obj } i) \begin{cases} < T_v\text{: the mapping description of the object is valid} \\ \geqslant T_v\text{: the mapping description is insufficient.} \end{cases} \tag{7}$$

If the value of the quotient in (7) exceeds a given threshold T_v, 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_v, 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 analysed 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 hypothesis, 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 analysed 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 SGXV. 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.

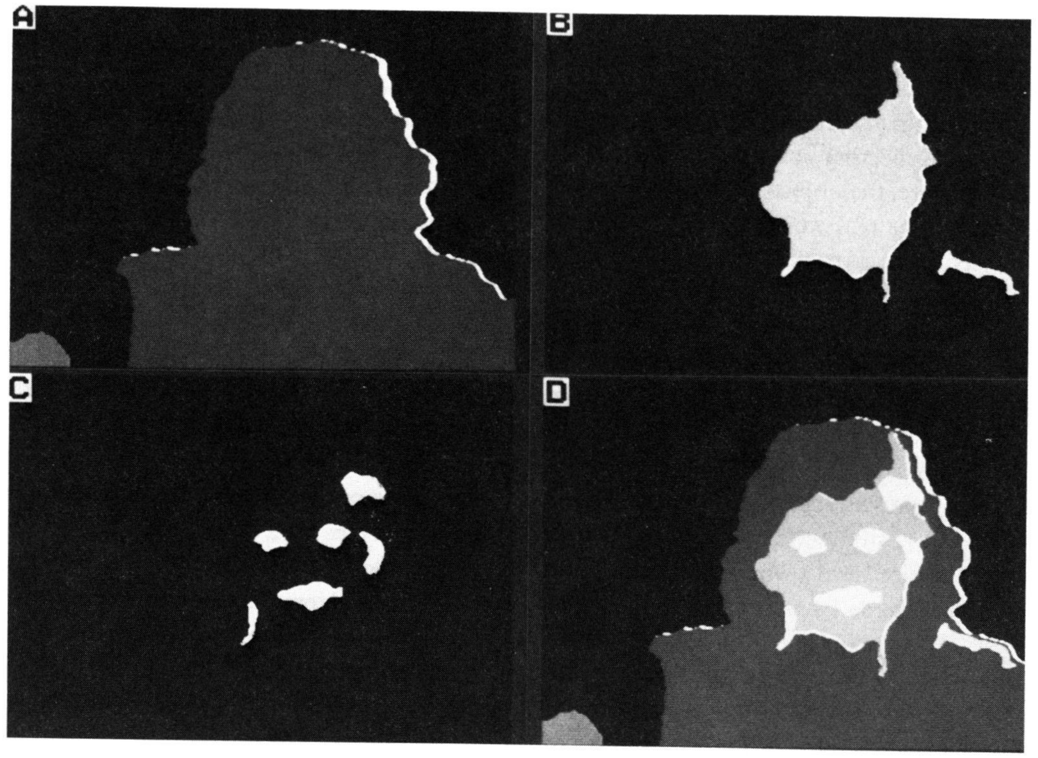

Fig. 7. Illustration of the hierarchical image analysis algorithm. (a) Analysed areas of the first level of hierarchy. (b) Analysed areas of the second level of hierarchy. (c) Analysed areas of the third level of hierarchy. (d) Superimposed, analysed areas of the first, second and third level of hierarchy.

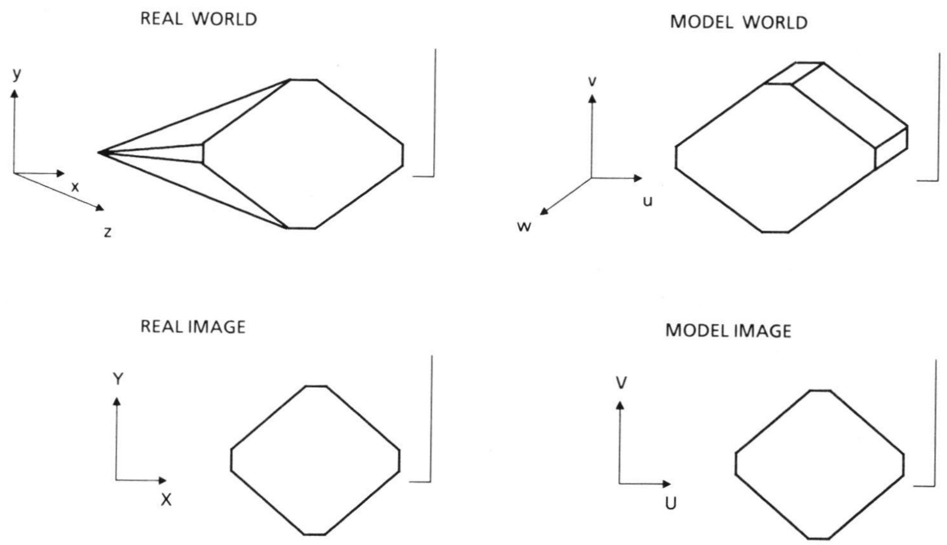

Fig. 8. Example of a real world and a model world giving the same real and model image.

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

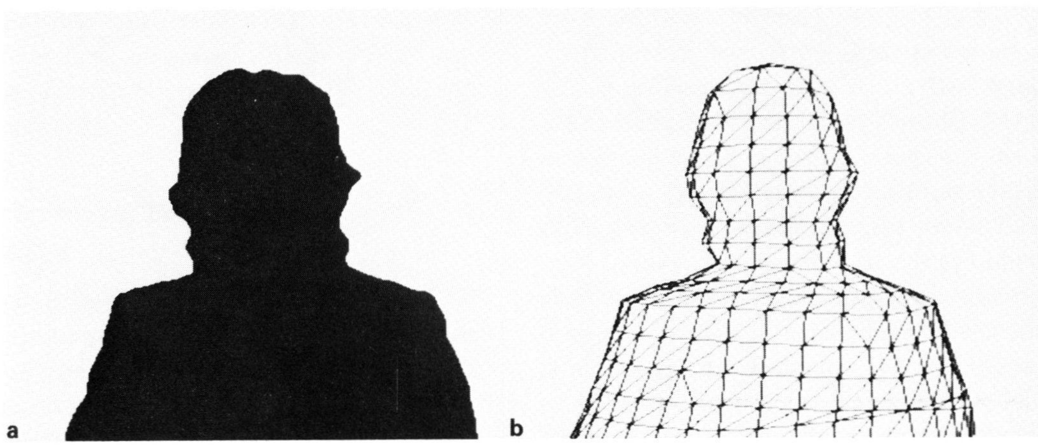

Fig. 9. Modelling 3D-objects. (a) Silhouette of the model object "Claire". (b) Wire-frame of the model object "Claire".

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'_k 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 of P'_k, 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'_k 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 $\hat{P}_k$. By minimizing the MSD between the real image S_{k+1} and the model image of $\hat{P}_k$ 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 $\hat{P}_{k+1}$ whose model image approximates S_{k+1}.

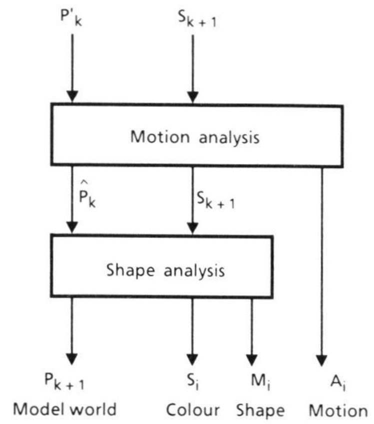

Fig. 10. Image analysis. Model: 3D-motion of 3D quasi-rigid objects. (P'_k, model world approximating real image k; S_{k+1}, colour signal of real image $k+1$; $\hat{P}_k$, model world P'_k with motion compensated objects; A_i, motion parameters of object i; P_{k+1}, model world approximating real image $k+1$; M_i, shape parameters of object i; S_i, colour parameters of object i.)

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'_k. Input to this algorithm are the current model world $\hat{P}_{i-1}$, which equals P'_k for the first iteration, the new real image S_{k+1}, and the object area O_{i-1}, 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 $A_i = (T_u, T_v, T_w, R_u, R_v, R_w)$ describing the translation T and rotation R of the model object i is estimated.

Using A_i and the model world $\hat{P}_{i-1}$ a motion compensated model image $\hat{S}_i$ 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

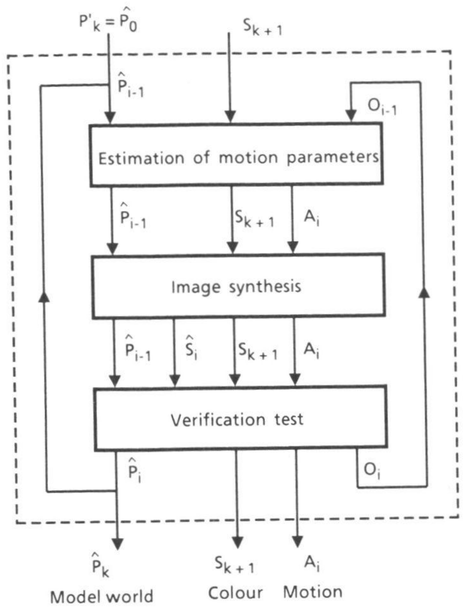

Fig. 11. Motion analysis. Model: 3D-motion of 3D quasi-rigid objects. (P'_k, model world approximating image k; S_{k+1}, colour signal of real image $k+1$; O_i, object area to be analysed; $\hat{S}_i$, model image of $\hat{P}_i$; A_i, motion parameter of object i.)

model image $\hat{S}_i$ computed from the current model world $\hat{P}_{i-1}$ 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 $\hat{P}_{i-1}$. If the frame difference decreases in the area of O_i, the model world $\hat{P}_{i-1}$ is updated to $\hat{P}_i$, 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).

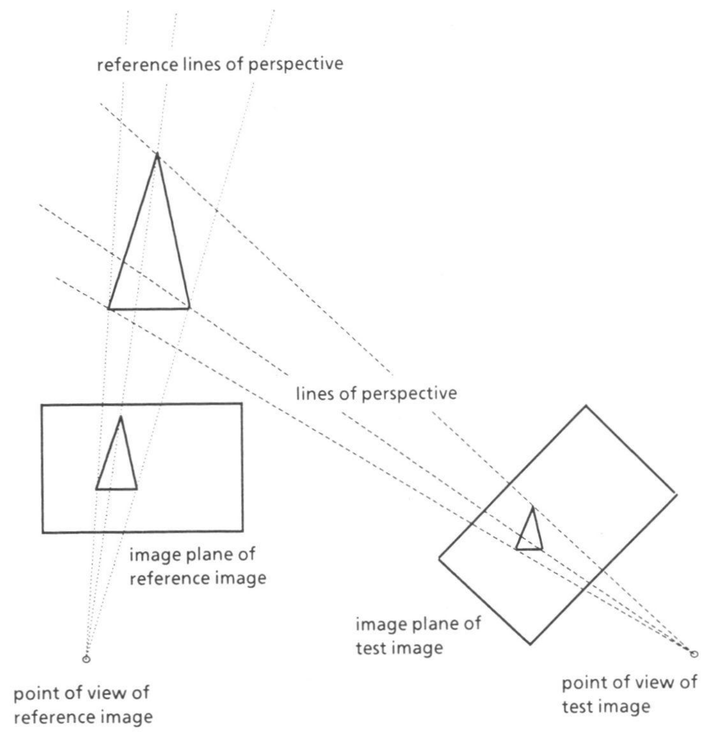

Fig. 12. Shape adaption using lines of perspective.

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 $\hat{P}_{i-1}$, 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 $\hat{P}_{i-1}$ and image S_{k+1} is evaluated and for the control points of object i the parameter set $\boldsymbol{M}_i = (\Delta_1, \ldots, \Delta_n, \ldots, \Delta_N)$ containing one shape parameter Δ_n for each control point is estimated. A shape parameter Δ_n determines the shift of control point $\boldsymbol{P}_n(u', v', w')$ on its reference line of perspective. $\boldsymbol{L}_n$ is a vector in the direction of the reference line of perspective of $\boldsymbol{P}_n$ (u', v', w'). Equation (8) shows how a control point $\boldsymbol{P}_n(u', v', w')$ moves as a function of its reference line of perspective vector $\boldsymbol{L}_n$ and its shape parameter Δ_n to $\boldsymbol{P}_n(u, v, w)$

$$\boldsymbol{P}_n(u, v, w) = \boldsymbol{P}_n(u', v', w') + \boldsymbol{L}_n \Delta_n. \tag{8}$$

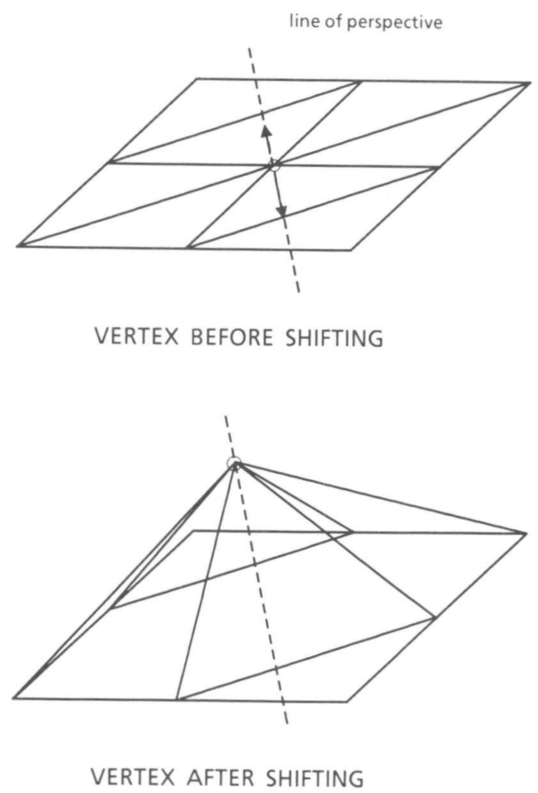

Fig. 13. Shape adaptation with triangular nets.

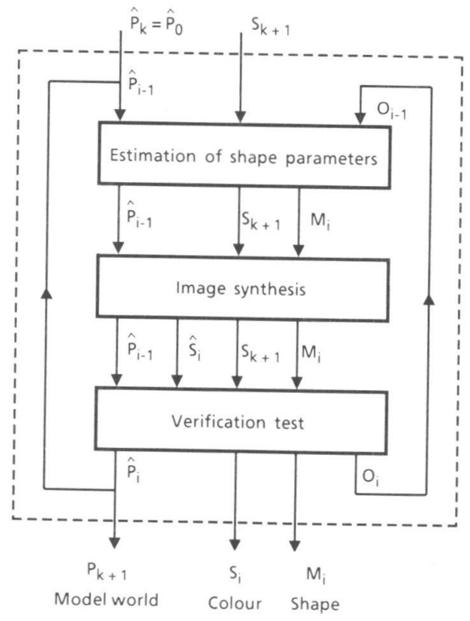

Fig. 14. Shape analysis. Model: 3D-motion of 3D quasi-rigid objects. ($\hat{P}_k$, model world with motion compensated objects; S_{k+1}, colour signal of real image $k+1$; $\hat{P}_i$, model world with partly updated shapes; M_i, shape parameters of object i; $\hat{S}_i$, model image of $\hat{P}_i$; S_i, colour parameters of object i.)

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 $\hat{S}_i$ of $\hat{P}_{i-1}$ and M_i. In areas where the MSD between $\hat{S}_i$ and S_{k+1} decreased, the shape parameters Δ_n 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 synthezised 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

Fig. 15. Example of an image analysis based on the source model of moving 3D-objects. (a) Reference image from the test sequence "Claire". (b) Successive actual image of the sequence. (c) Wire-frame of two connected sub-objects. (d) Frame difference of model image and actual image after motion compensation. (e) Frame difference of model image and actual image after motion compensation and shape adaptation. (f) Frame difference with source model of moving 2D-objects for comparison.

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_{A_i}, R_{M_i}, R_{S_i} 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_i and M_i of an object have to be transmitted and whether or not 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

$$\text{SIZE (obj } i) \begin{cases} \in [129, 256]: & \text{Block-oriented intraframe coding (small objects } B), \\ \in [1, 128]: & \text{Object-oriented intraframe coding with transmission of shape information (small objects } A), \end{cases} \qquad (10)$$

where the size of object i is measured by the number of pels 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 information has to be transmitted for each object.

Table 1
Coder modes

Parameter sets to be transmitted	Modes: Normal objects	Special objects	Small objects A	Small objects B
Motion parameters	×			
Shape parameters	×	×	×	
Colour parameters	Hybrid coding	Intraframe coding	Intraframe coding	Intraframe block-coding

5.3. Priority control

The priority control arranges the parameter sets A_i, M_i, S_i to be transmitted in an order of decreasing priority for each image. To the parameter sets A_i and M_i of each object the highest priority is assigned because these parameter sets 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 $\mathrm{PR}(S_i)$ of a parameter set S_i of an object i is determined by object features as the object size O_{si} and the quality of object synthesis O_{qi}.

By the normalization of the object features O_{si}, O_{qi}

$$O_{si} = \text{SIZE min}/\text{SIZE(obj } i) \qquad (11)$$

and

$$O_{qi} = \text{MADFD(obj } i)/\text{MADFD max}, \qquad (12)$$

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

Thus, the priority $\mathrm{PR}(\cdot)$ of the parameter sets A_i, M_i, S_i results as

$$\mathrm{PR}(A_i) = \max_i\{\mathrm{PR}(S_i)\},$$

$$\mathrm{PR}(M_i) = \max_i\{\mathrm{PR}(S_i)\},$$

$$\mathrm{PR}(S_i) = f(O_{si}, O_{qi}), \qquad (13)$$

where $f(\cdot)$ denotes the functional relationship of the priority $\mathrm{PR}(S_i)$ and its arguments O_{si} and O_{qi}.

A simple example of the priority control is shown in Table 2. There are two objects with $O_{s1} = 0.1$, $O_{q1} = 0.1$ and $O_{s2} = 1$, $O_{q2} = 1$, i.e. object 1 is ten times the size of object 2 and has a MADFD which is one tenth of the MADFD of object 2. Object 2 is a special object. Hence, no motion information is transmitted for object 2. Each parameter set A_1, M_1, S_1, M_2, S_2 has its own priority $\mathrm{PR}(\cdot)$. For the special priority function $\mathrm{PR}(S_i) = O_{si}\, O_{qi}$ the priority queue is shown in Table 2(b). 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 after 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

Table 2
Example of the priority control. (a) Priority calculation. (b) Priority queue.

(a)

Object i	O_{si}	O_{qi}	Priority of the parameter sets
1	0.1	0.1	$\mathrm{PR}(A_1) = 1$ (motion) $\mathrm{PR}(M_1) = 1$ (shape) $\mathrm{PR}(S_1) = 0.01$ (colour)
2	1	1	$\mathrm{PR}(M_2) = 1$ $\mathrm{PR}(S_2) = 1$

(b)

Priority function	Priority $\mathrm{PR}(\cdot)$
$\mathrm{PR}(A_1)$	1
$\mathrm{PR}(M_1)$	1
$\mathrm{PR}(M_2)$	1
$\mathrm{PR}(S_2)$	1
$\mathrm{PR}(S_1)$	0.01

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_b.

The priority PR_p results as

$$PR_p = g(B_p, B_b), \tag{14}$$

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

By the normalization of the block features B_b, B_p

$$B_b = \begin{cases} 1, & \text{boundary block,} \\ 0, & \text{else,} \end{cases} \tag{15}$$

$$B_p = \text{MADFD(block)}/\text{MADFD max}, \tag{16}$$

all block features are within the range [0, 1], where high values of B_b, B_p indicate high priority. The control parameter B_b 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

$$PR_{block} = h(PR(S_i), PR_p), \tag{17}$$

where $h(\cdot)$ denotes the functional relationship of the priority PR_{block} and its arguments $PR(S_i)$ 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

Fig. 16. Reference image from the test sequence "Miss America".

a reduced field frequency of 10 Hz and a quantization of 8 bit per sample. Each of the non-interlaced fields consists of 288 lines and 352 picture elements per line for the luminance and 144 lines and 176 picture elements for the chrominance, respectively. The test sequence "Miss America" represents a typical videophone scene showing head and shoulders in motion. Fig. 16 and Fig. 17 show two successive images of the sequence where closing eyes and an opening mouth are superimposed onto head motion.

In a first experimental implementation of the coder, the source model of 2D-objects as described in Section 3 has been applied. The shapes of the

Fig. 17. Successive actual image of the test sequence "Miss America".

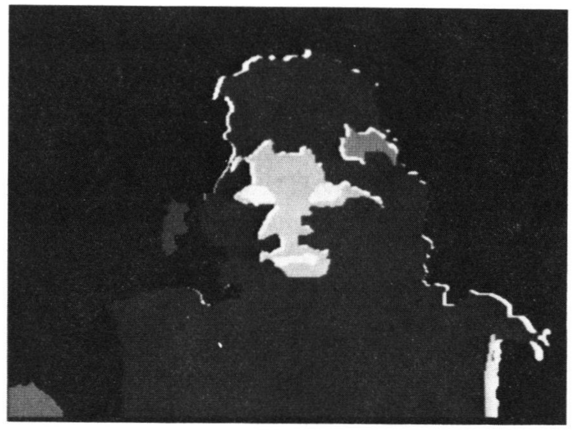

Fig. 18. Shapes of the objects found by image analysis of Figs. 16 and 17.

detected objects obtained by the image analysis are shown in Fig. 18 where uncovered image regions and the moving objects are displayed with white and different gray luminance values.

According to this source model, the motion information A_i is represented by eight mapping parameters $a_1, \ldots, a_8$ per object. With the exception of a_3 and a_6 the parameters are normalized by a factor K. This factor K is coded by 4 bits and depends on the maximum coordinates contained in the mapped object. The normalization yields an equivalent motion compensation accuracy for all objects. Each normalized parameter is coded by 6 bits. The parameters a_3 and a_6 describe the horizontal and vertical displacement of the object. They are represented with quarter pel accuracy and coded by 7 bits per parameter. Thus, $R_{A_i} =$ 54 bits are needed to code the mapping parameter set of each object. No predictive coding is used in this experiment. Assuming 3 to 5 moving objects in a scene, e.g. head, shoulders and arms which are typical for videophone scenes, the total amount for the motion information results in 150 to 250 bit per image.

In order to encode the shape information, a predictive contour coding algorithm [10] has been developed which encodes the temporal differences of the object boundaries. For encoding the shape information of a large object i a bit rate R_{M_i} of 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 800 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 16×16 pels and two chrominance blocks of 8×8 pels.

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_i and shape M_i 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.

Fig. 19. Prediction image of an object oriented analysis-synthesis coder.

The advantage of object-oriented coding becomes obvious when prediction images are compared. Fig. 19 shows an image as generated by object-oriented coding. Fig. 20 shows the corresponding image generated by block-oriented coding. Both images can be interpreted as coder output

Fig. 20. Prediction image of a block-oriented hybrid coder.

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 $\mathrm{PR}(S_i)$ of a parameter set S_i according to

$$\begin{aligned}\mathrm{PR}(S_i) &= f(O_{si}, O_{qi})\\ &= (1 + w_s O_{si})(1 + w_q O_{qi}) \qquad (18)\end{aligned}$$

and a priority PR_p according to

$$\begin{aligned}\mathrm{PR}_p &= g(B_p, B_b)\\ &= (1 + w_p B_p)(1 + w_b B_b), \qquad (19)\end{aligned}$$

are used.

Thus, the priority of a macro block $\mathrm{PR}_{\mathrm{block}}$ based on these normalized object and block features results in

$$\begin{aligned}\mathrm{PR}_{\mathrm{block}} &= h(\mathrm{PR}(S_i), \mathrm{PR}_p)\\ &= \mathrm{PR}(S_i)\,\mathrm{PR}_p\\ &= (1 + w_s O_s)(1 + w_q O_q)\\ &\quad \times (1 + w_p B_p)(1 + w_b B_b). \qquad (20)\end{aligned}$$

If all weightening 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 synthezised 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.

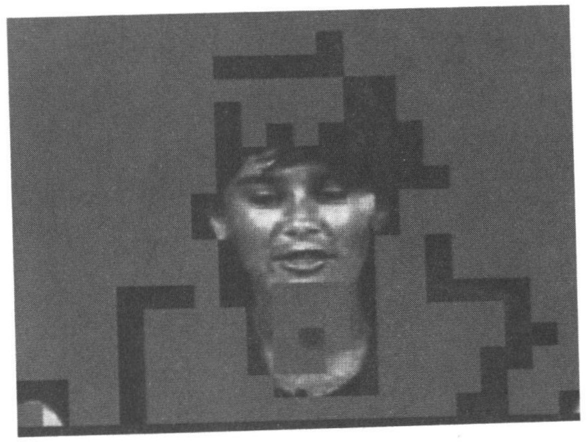

Fig. 21. Colour update information for a priority control accentuating large prediction errors and boundary blocks.

To demonstrate the influence of the priority control $\mathrm{PR}_{\mathrm{block}}$, the coding of large prediction errors and boundary blocks have jointly been accentuated by choosing $w_s = 0$, $w_q = 0$, $w_b = 20$ and $w_p = 1$ (Fig. 21). By this weightening, 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 the known visible 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 colour parameter transmission 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 coding gain of the intraframe 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 Fernmeldetechnischen Zentralamt. We would like to express our thanks to Professor Gallenkamp for encouraging this work and to Dr. Stenger, Dr. Geuen, Dr. Billotet-Hoffmann and Mr. Heipel for many helpful discussions.

References

[1] K. Aizawa, H. Harashima and T. Siato, "Model-based synthesis image coding system", *Picture Coding Symp.* (PCS '87), Pres. No. 3.11, Stockholm, Sweden, June 9-11, 1987.

[2] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects", *IEEE Pattern Anal. Mach. Intell.*, Vol. PAMI-7, No. 4, July 1985, pp. 384-401.

[3] H. Busch, "Subdividing non rigid 3D objects into quasi rigid parts", *IEE 3rd Int. Conf. Image Process. Applic.*, Warwick, U.K., July 1989.

[4] P. Eckman and V.W. Friesen, *Facial Coding System*, Consulting Psychologists Press Inc, 577 College Arc, Palo Alto, 1977.

[5] M. Eden and M. Kocher, "On the performance of a contour coding algorithm in the context of image coding—Part I: Contour segment coding", *Signal Process.*, Vol. 8, No. 4, July 1985, pp. 381-386.

[6] R. Forchheimer and O. Fahlander, "Low bit rate coding through animation", *Picture Coding Symp.* (PCS '83), Pres. No. 13.5, Davis, CA, March 1983.

[7] R. Forchheimer, O. Fahlander and T. Kronander, "A sematic approach to the transmission of face images", *Picture Coding Symp.* (PCS '84), Pres. No. 10.5, Cesson-Sevigne, July 3-5, 1984.

[8] W. Geuen and F. Kappei, "Principle strategy of model based source coding", *Picture Coding Symp.* (PCS '87), Pres. No. 12.2, Stockholm, Sweden, June 9-11, 1987.

[9] M. Gilge, "Coding of arbitrarily shaped image segments using moment theory", in: J.L. Lacoume *et al.*, eds., *Signal Processing IV, Theories and Applications* (*Proc. 4th Eur. Signal Processing Conf. EUSIPCO* '88), Grenoble, France, September 1988, Elsevier, Amsterdam 1988.

[10] K.-W. Hahn, "Untersuchungen zur strukturierten Beschreibung und Codierung von Bildsegmenten für ein Konturcodiersystem", Diplomarbeit University of Hannover, Hannover, Fed. Rep. Germany, July 1989.

[11] M. Hötter and R. Thoma, "Image segmentation based on object oriented mapping parameter estimation", *Signal Process.*, Vol. 15, No. 3, October 1988, pp. 315-334.

[12] M. Kanako, A. Koike and Y. Hatori, "Codings with knowledge-based analysis of motion pictures", *Picture Coding Symp.* (PCS '87), Pres. No. 12.3, Stockholm, Sweden, June 9-11, 1987.

[13] F. Kappei, "Modellierung und Rekonstruktion bewegter dreidimensionaler Objekte aus einer Fernsehbildfolge", Ph.D. Thesis, University of Hannover, Hannover, Fed. Rep. Germany, 1988.

[14] F. Kappei and C.-E. Liedtke, "Modelling of a natural 3D scene consisting of moving objects from a sequence of monocular TV images", *Proc. SPIE*, Vol. 860, p. 126.

[15] S. Murahami, H. Ichihara and T. Miyata, "Model-based coding and facial image transformation for intelligent video communications", *Picture Coding Symp.* (PCS '88), Pres. No. 4.7, Torino, Italy, September 12-14, 1988.

[16] A.N. Netravali and B.G. Haskell, *Digital Pictures*, Plenum Press, New York, 1988.

[17] R. Plompen, Y. Hatori, W. Geuen, J. Guichard, M. Guglielmo and H. Brusewitz, "Motion video coding in CCITT SG XV—The video source coding", *Proc. IEEE GLOBECOM*, Vol. II, December 1988, pp. 31.2.1-31.2.8.

[18] J.L. Potter, "Velocity as a cue to segmentation", *IEEE Trans. Syst. Man. Cybern.*, May 1975, pp. 390-394.

[19] D.F. Rogers, *Procedural Elements for Computer Graphics*, McGraw-Hill, New York, 1985.

[20] H. Schiller and B.B. Chaudhuri, "Efficient coding of side information in a low bit rate hybrid image coder", *Signal Processing*, accepted for publication.

[21] P. Spoer, "Schätzung der 3-dimensionalen Bewegungsvorgänge starrer, ebener Objekte in digitalen Fernsehbildfolgen mit Hilfe von Bewegungsparametern", Ph.D. Thesis, University of Hannover, Hannover, Fed. Rep. Germany, 1987.

[22] R. Thoma, "A segmentation algorithm for motion compensating field interpolation", *Picture Coding Symp.* (PCS '87), Pres. No. 6.1, Stockholm, Sweden, June 9-11, 1987.

[23] R.Y. Tsai and T.S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch", *IEEE Trans. Acoust. Speech, Signal Process.*, Vol. ASSP-29, No. 6, December 1981, pp. 1147-1152.

[24] S. Ullmann, *The Interpretation of Visual Motion*, M.I.T. Press, Cambridge, MA, 1979.

[25] W.J. Welsh, "Model-based coding of moving images at very low bit rates", *Picture Coding Symp.* (PCS '87), Pres. No. 3.9, Stockholm, Sweden, June 9-11, 1987.