REPRINTED FROM:

SIGNAL PROCESSING

# IMAGE COMMUNICATION

Volume 2, No. 4, December 1990

## OBJECT-ORIENTED ANALYSIS–SYNTHESIS CODING BASED ON MOVING TWO-DIMENSIONAL OBJECTS

Michael HÖTTER

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

pp. 409–428

ELSEVIER, AMSTERDAM – OXFORD – NEW YORK – TOKYO

Signal Processing: *Image Communication* 2 (1990) 409–428
Elsevier 

# OBJECT-ORIENTED ANALYSIS–SYNTHESIS CODING BASED ON MOVING TWO-DIMENSIONAL OBJECTS

Michael HÖTTER

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

Received 1 December 1989
Revised 19 April 1990

**Abstract.** An object-oriented analysis-synthesis coder is presented which encodes arbitrarily shaped objects instead of rectangular blocks. The objects are described by three parameter sets defining their motion, shape and colour. Throughout this contribution, the colour parameters denote the luminance and chrominance values of the object surface. The parameter sets of each object are obtained by image analysis based on source models of moving 2D-objects and coded by an object-dependent parameter coding. Using the coded parameter sets an image can be reconstructed by model-based image synthesis. In order to cut down the generated bit-rate of the parameter coding, the colour updating of an object is suppressed if the modelling of the object by the source model is sufficiently exact, i.e., if only a relatively small colour update information would be needed for an errorless image synthesis. Omitting colour update information, small position errors of objects denoted as geometrical distortions are allowed for image synthesis instead of quantization error distortions. Tolerating geometrical distortions, the image area to be updated by colour coding can be decreased to 4% of the image size without introducing annoying distortions. As motion and shape parameters can efficiently be coded, about 1 bit per pel remains for colour updating in a 64 kbit/s coder compared to about 0.1 bit per pel in the standard reference coder (RM8) of the CCITT. Experimental results concerning the efficient coding of motion and shape parameters are given and discussed. The coding of the colour information will be dealt with in further research.

**Keywords.** Object-oriented analysis-synthesis coder, moving video signals.

## 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 [15, 17]. Thus an image is described by independently moving square blocks which can lead to visible distortions known as blocking and mosquito effects in low bit-rate codecs. To avoid these image distortions more appropriate source models for describing the image have to be introduced. In [14], a coding method called object-oriented analysis-synthesis coding has been presented. It subdivides each image into objects and encodes each object by three sets of parameters defining its motion, shape and colour information. In the following, the colour parameters denote the luminance and chrominance values of the object surface. 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. In [14], the source model of two-dimensional rigid objects with three-dimensional motion and the source model of three-dimensional, rigid objects with three-dimensional motion have been considered. Based on these source models image analysis algorithms have been developed [9, 11, 14] which estimate the shape and motion of the objects in a scene and automatically generate the three sets of parameters. First coding approaches of these three sets of parameters, i.e., the parameter coding, have been presented in [14]. A detailed discussion of parameter coding will be addressed by this contribution. Two source models, both based on two-dimensional objects, and their effectiveness for parameter coding are discussed. These are the source model of rigid objects with three-dimensional motion introduced in [14] and the source model of flexible objects with two-dimensional motion introduced below. The introduction of a source model based on flexible objects is motivated by a more efficient parameter coding and a very simple realization of the coder components compared to the source model based on rigid objects. The parameter coding can be controlled by the object-oriented information generated by the image analysis. When irreversible coding techniques have to be applied in order to cut down the generated bit-rate, the coding of the colour parameters of an object is suppressed if the modelling of the object by image analysis is sufficiently exact, i.e., if only a relatively small colour update information would be needed for an errorless image synthesis. Omitting colour update information, small position errors of objects denoted as geometrical distortions are allowed for image synthesis instead of quantization error distortions as geometrical distortions are less annoying to a human observer.

In Section 2 the concept and structure of the coder is explained. Section 3 deals with the concept and techniques of parameter coding and the object-oriented coder control. Further, the influence of the source models on the parameter coding and the properties of the source models for image analysis and image synthesis are discussed. First results obtained with the coding of motion and shape parameters are presented in Section 4.

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

To explain the concept and structure of the object-oriented analysis-synthesis coder the block diagram of Fig. 1 is used [14]. Input of the coder is a sequence of images. The image analysis subdivides each image of this 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 $i$. The object parameters depend on the type of source model being applied. The calculated parameter sets are coded by parameter coding. The parameter coding depends on a receiver model which includes assumptions about the visibility of coding errors. The coded parameter sets are transmitted to the

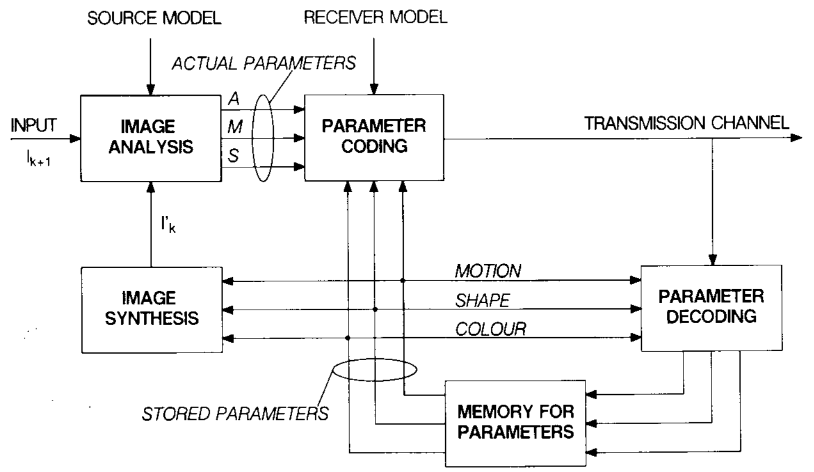



Fig. 1. Block diagram of an object-oriented analysis-synthesis coder.





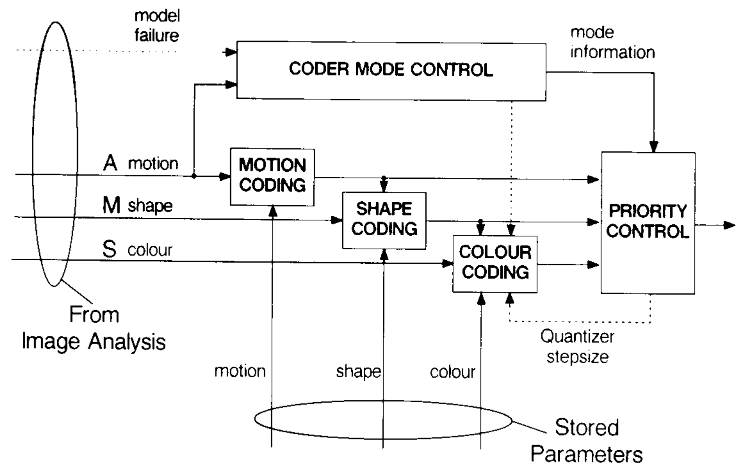



Fig. 2. Parameter coding.

receiver, decoded by parameter decoding and stored in a parameter memory. The parameter memory of the coder and decoder contains the same parameter information and allows both the coder and decoder to identically reconstruct a transmitted image by image synthesis. The reconstructed image $I'_k$ is displayed at the decoder and is also used for image analysis of the next input image $I_{k+1}$ at the coder. Additionally, the stored parameters are used for the coding and decoding of the parameter sets of the next input image $I_{k+1}$.

An object-oriented image analysis algorithm based on the source model of two-dimensional objects has been presented in [9, 11, 14]. This image analysis algorithm is also used here. It subdivides an image into patches (objects) each of which is assumed to be the projection of a moving plane oriented arbitrarily in the three-dimensional scene. In the following section, the parameter coding of the parameter sets generated by image analysis algorithms which are based on the source models of rigid and flexible two-dimensional objects, respectively, will be dealt with in detail.

## 3. 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. 2.

### 3.1. Coding of the motion, shape and colour parameters

The three parameter sets $A_i$, $M_i$, $S_i$ are generated by image analysis in a PCM representation. In order to increase the coding efficiency additional individual parameter coding techniques are applied.

#### 3.1.1. Coding of the shape parameters

While block-oriented hybrid coding techniques [17] transmit only two parameter sets, namely the motion and colour information of each block, object-oriented analysis-synthesis coding additionally has to transmit the shape of each object. In order to obtain a higher coding gain for object-oriented coding, the additional bit rate $R_M$ required for transmitting the shape information $M$ has to be justified by a reduction of the bit rates $R_A$ and $R_S$ required for motion $A$ and colour information $S$. Two properties of object-oriented coding contribute to this requirement. First, the shape coding algorithm used here is very efficient, i.e., the shape coding bitrate $R_M$ is very low. Secondly, the synthesis of the colour information is improved at object boundaries using shape information as, in contrast to block-oriented hybrid coding, different displacement vectors may be assigned to parts of a block.

Among the several approaches for contour coding [2, 3, 16], Fourier descriptors and polygon approximations have been shown to be most and equally effective for an object-oriented coder concept [3]. The main disadvantage of the Fourier descriptors is that of all transform techniques, the difficulty of describing local information, especially shape areas of high curvature as e.g. corners. The main disadvantage of the polygon approximation is the little naturally looking, rough representation of the object shape. In order to handle both problems, a shape coding algorithm is used in this paper which includes an effective shape approximation by a combination of polygon- and spline-representation of the shape. The vertices defining these approximation functions are coded in a temporally predictive way, i.e., only temporal differences of object shapes are coded using stored shape information from the previous image and motion information from the current image.

The object shape of each object is calculated by an image analysis algorithm as described in detail in [11, 14]. According to [8], the object shape is approximated by a polygon-representation. The number of calculated vertices depends on the chosen quality of the shape approximation. The quality is controlled by the absolute distance $d_{max}$ between the approximated and the real object shape. The higher $d_{max}$, the coarser the approximation and the less vertices are calculated for the shape description. In order to combine this approach with the more natural looking spline functions, the vertices of the polygon-representation are used to calculate a spline-representation of the shape as well. Wherever the spline-representation fulfills the quality criterion $d_{max}$, it is used instead of the polygon, so that the final approximation of the shape is composed of spline and polygon parts. By this, a natural looking approximation of the object shape is achieved by

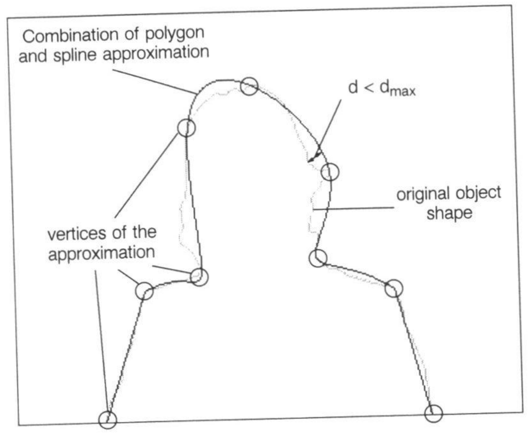



Fig. 3. Example of a coarse contour approximation ($d_{max} = 15$) combining polygon and spline representation.

a low number of vertices, where the quality measure $d_{max}$ is guaranteed everywhere. An example of a coarse shape approximation ($d_{max} =$ 15 pels) combining the polygon- and spline-representation is shown in Fig. 3. As can be seen, the final approximation is composed of spline and polygon parts.

The vertices of the shape description of an object $i$ are predictively coded. First, the vertices of the object $i$ of the previous image are motion compensated by means of the actual motion parameter $A_i$ of object $i$ as described in [14]. The motion compensated vertices represent a first approximation of the actual shape of object $i$. Then, a verification of the motion compensated vertices is achieved. Three classes of vertices are distinguished: the vertices which are maintained as they are appropriate for the approximation of the actual shape, the vertices which are rejected as they cannot be used for the approximation of the actual shape and the vertices which have to be additionally inserted in order to guarantee that the quality measure $d_{max}$ of the approximation is fulfilled everywhere. If the model assumptions of the object-oriented analysis–synthesis coder [14] hold, changes of the object shape are only caused by motion. Under these conditions, the motion compensated vertices of the object $i$ of the previous





image are identical to the vertices of the actually considered object $i$, i.e., no additional update shape information has to be transmitted to the receiver. If however the model assumptions of the coder only partly hold, the shape information to be coded results as follows: the positions of the new inserted vertices, the changes in type of curve of the approximation and the indices of the rejected and inserted vertices within the approximation. As coding techniques, relative addressing is applied for the position of the new inserted vertices and runlength coding for the type of curve as well as for the indices of the rejected and inserted vertices. The more valid the model assumptions, the less shape information has to be coded.

Compared to known techniques [2], this shape coding technique reduces the bit rate by a factor of about 4 to 6 under the condition of a high quality shape approximation ($d_{max} = 2.9$ pels), i.e., an average bit rate of only about 0.3 bit per shape pixel is needed [7].

### *3.1.2. Coding of the motion and colour parameters*

Two different descriptions of object motion are considered in this contribution: first, a displacement vector field in the case of the source model of flexible objects with two-dimensional motion, and second, a set of eight mapping parameters for each moved object in the case of rigid objects with three-dimensional motion. For both source models, the motion parameters are coded by known DPCM techniques.

In the case of the source model of flexible objects with two-dimensional motion, the displacement vector field is first predicted using a spatial three coefficient predictor which minimizes the mean squared prediction error. The coefficients are calculated for the $X$- and $Y$-component of each vector field separately and coded by 6 bits each. The higher the spatial correlation of adjacent displacement vectors, the more accurate the prediction and hence, the lower the data amount for the transmission of the prediction error of the displacement vector field which is coded by runlength coding.

In the case of the source model of rigid objects with three-dimensional motion, the eight mapping parameters $(a_1, a_2, \ldots, a_8)$ of an object are first normalized by a factor $K$ except of $a_3$ and $a_6$. This factor $K$ is coded by 4 bits and depends on the spatial extension of the object. The normalization yields a motion description of identical accuracy for all objects. The normalized parameters $(a_1, a_2, a_4, a_5, a_7, a_8)$ are coded by 6 bits each. The parameters $a_3$ and $a_6$ describe the horizontal and vertical displacement of the object. They are quantized with quarter pel accuracy and coded by 7 bits each. Thus, the data amount for the motion information depends directly on the number of objects to be evaluated.

The colour coding is still under study. At present, original 8 bit PCM luminance and chrominance values are substituted in areas of model failure and areas of uncovered background. The colour coding technique under study is an intraframe coding approach [5] which is not restricted to a block structure but can be adapted to object boundaries. Furthermore, it is planned for intraframe coding to decompose large areas of model failure into internal quadratic blocks of picture elements and boundary blocks where the boundary blocks are split into segments according to the shape information and coded as described in [5].

### *3.2. Priority control and coder mode control*

The resulting coded information $R_i$ of each object $i$ consists of three components $R_{Ai}$, $R_{Mi}$, $R_{Si}$ representing motion, shape and colour information. A priority control decides which of the coded information of each object $i$ will be transmitted to the receiver (Fig. 2). This decision is controlled by the mode information which is generated by the coder mode control. The coder mode control distinguishes between two classes of objects depending on the validity of the source model (Table 1). The validity of the source model is decided for each object $i$ by image analysis (Fig. 2).

In the case of objects which cannot be described by the source model, the motion information is

Table 1
Coder modes

| Parameter sets to be transmitted | Modes: Synthesized objects allowing geometrical distortions | Model failure |
|---|---|---|
| Motion parameters | × | |
| Shape parameters | × | × |
| Colour parameters | | × |

omitted. Only shape and colour information are transmitted.

If the modelling of the object is sufficiently exact, only motion and shape information are transmitted to the receiver. This includes objects whose shape update information has been set to zero. Thus by suppression of colour update information of an object, small position errors of objects denoted as geometrical distortions are allowed for image synthesis instead of quantization error distortions. Assuming the same mean square distortion error, 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 colour update information.

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

The probability of model failure, i.e., the amount of the image area which cannot be described by the source model, and hence, the amount of data for colour update information depends on the source model itself. This influence of the source model on the parameter coding will be discussed in the following.

### 3.3. *Influence of the source model on the parameter coding*

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, both based on two-dimensional objects, and their effectiveness for parameter coding are discussed. They are first the model of rigid objects with three-dimensional motion and second the model of flexible objects with two-dimensional motion. An image analysis algorithm is used that formulates the analysis task as a hierarchical application of object motion and object boundary estimation [11]. Figure 4 shows the structure of the algorithm which consists of three parts; the estimation of the motion, the internal image synthesis and the verification test.

In the case of rigid objects with three-dimensional motion, eight mapping parameters are calculated for each object to describe object motion [11, 14]. A similar source model describing object motion by four mapping parameters has been used by May in [13]. Inputs to the image analysis scheme are the luminance signals $L'_k$ and $L_{k+1}$ of two successive pictures $I'_k$ and $I_{k+1}$ of an image sequence. The prime denotes that $L'_k$ is the transmitted and synthesized version of $L_k$. In a first step, the starting hypothesis is postulated that

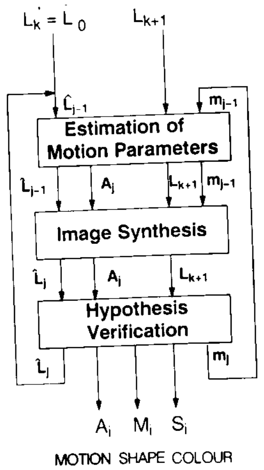



Fig. 4. Image analysis. Model: 2D-objects ($L_{k+1}$, luminance signal of image $k+1$; $\hat{L}_j$, synthesized luminance signal to describe $L_{k+1}$; $m_j$, areas to be analysed; $A_j$, motion parameters of analysed areas; $j$, index of hierarchy; $i$, object $i$).





the whole scene represents one object, which does not move. Thus, the synthesized signal $\hat{L}_0$ is identical to the signal $L'_k$. According to the starting hypothesis, $L'_k$ and $L_{k+1}$ should have the same luminance values. The verification test evaluates the frame differences between $L'_k$ and $L_{k+1}$ and detects that area $M_1$ where the hypothesis is true. The area $M_1$ represents the object stationary 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 signal $\hat{L}_1$ and the actual signal $L_{k+1}$. 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 $L_{k+1}$ is hierarchically refined until all objects are described in their mappings and their boundaries.

For the source model of flexible objects with two-dimensional motion, a displacement vector field is used to describe object motion. A modified hierarchical blockmatching algorithm as presented by Bierling [1] has been applied to calculate the displacement vector field. The essential parameters of the blockmatching algorithm are summarized in Table 2. The number of hierarchy steps for image analysis is now fixed and equals two. In the first step of hierarchy, the object stationary background

Table 2
Parameters of a hierarchical blockmatching algorithm with three levels of hierarchy

| Parameters at level | 1 | 2 | 3 |
|---|---|---|---|
| Max. update displacement | ±7 | ±3 | ±0.5 |
| Measurement window size | 64 | 16 | 16 |
| Spatial resolution of the measured vector field | 32 | 16 | 16 |
| Spatial resolution of the vector field for synthesis | 1 pel by bilinear interpolation of the measured vector field | | |

is detected as described above. In the second step of hierarchy, for each disjunct, changed image region, a displacement vector field instead of eight mapping parameters is calculated by means of the synthesized signal $\hat{L}_1$ and the actual signal $L_{k+1}$. Then, for each disjunct, changed image region an image synthesis is performed by means of the calculated displacement vectors. If there are i.e. covered or uncovered image regions, the assumption that each image area represents the mapping of an object whose motion can locally be described by displacement vectors is not fulfilled and hence the motion description is only valid in a part of the region. For that reason the hypothesis verification now yields a detection of those image parts whose temporal change cannot be described with sufficient accuracy by displacement vectors. These image parts are denoted as model failures. In contrast to the image analysis based on rigid two-dimensional objects, it is not possible to detect moving objects in front of moving objects but the image analysis is restricted to the detection of the following objects: the stationary background consisting of all picture elements which do not move, the objects whose temporal change can be described by displacement vectors and the objects whose temporal change cannot be described with sufficient accuracy by displacement vectors and which are denoted as model failures.

Object-oriented analysis-synthesis coders based on these two source models have been implemented and experimentally investigated by computer simulations based on the CCITT test sequence 'Miss America'. As the colour coding has not yet been realized for the coders, original 8 bit PCM colour values are substituted in areas of model failures. Hence, the influence of quantization error distortions of the colour update is not considered in these experiments.

The first experiment compares the two source models under the constraint that the total amount for shape and motion information should in each case be in the same range as the corresponding side information of a CCITT coder [19], i.e., it should be about 1500 bits per picture in order to leave a sufficient bitrate for the colour update information. The smaller the image area to be updated by colour coding, i.e., the smaller the area of model failures, and the smaller the variance of the synthesis error, i.e., the higher the image quality, the more efficient is the parameter coding in this experiment. The variance of the synthesis error is measured excluding the stationary background and the model failures. As can be seen from Table 3, the motion information of the source model of rigid objects with three-dimensional motion is very low. This is mainly due to the fact that a big image area is described in its motion by only a few parameters, i.e., the motion description of the scene is very efficient. This advantage has to be paid for by quite a large amount of shape information, a large image area of model failures and an insufficient quality of the synthesized image. These disadvantages are caused by the fact that the model assumption of rigid objects usually does not hold for natural scenes. In contrast, the source model of flexible objects with two-dimensional motion suffers from a large amount of motion information, which is due to the high resolution of the motion description of the objects. On the other hand, the data amount for the shape information is lower than for the source model of rigid objects, the image area of model failures is reduced and the variance of the luminance error of the synthesis is decreased as well. These advantages are mainly due to the fact that for videophone scenes most of the image changes, which are caused by object motion and object deformations, can be modelled by locally translatory motion.

The second experiment comparing the above-mentioned source models was performed under the constraint that the variance of the luminance error of the synthesis, i.e., the quality of the image reconstruction, should be the same for both coders. Again, the variance of the synthesis error is measured excluding the stationary background and the model failures. In this situation, the more efficient the parameter coding becomes, the smaller the image area to be updated by colour coding, i.e., the smaller the area of model failures, and the lower the data amount for motion and shape information. The results of this experiment are summarized in Table 4. As can be seen, the amount of shape and motion information as well as the image area of model failures is drastically increased for the source model of rigid objects compared to the first experiment (Table 3). This is due to the fact that more rigid objects have to be generated to describe the flexible object parts more accurately for image synthesis. Moreover, a lot of these flexible parts cannot be modelled and hence, they are denoted as model failures.

Table 3

Influence of the source model on the parameter coding. Comparison of the two source models under a side information constraint

| Model | Motion information (bits/picture) | Shape information (bits/picture) | Area of model failures (%) | Variance of the luminance error of the synthesis ($1/255^2$) |
|---|---|---|---|---|
| Rigid 2D-objects with 3D-motion | 450 | 950 | 9 | 23 |
| Flexible 2D-objects with 2D-motion | 1000 | 500 | 4 | 6 |





Table 4

Influence of the source model on the parameter coding. Comparison of the two source models under an equal synthesis error constraint

| Model | Motion information (bits/picture) | Shape information (bits/picture) | Area of model failures (%) | Variance of the luminance error of the synthesis ($1/255^2$) |
|---|---|---|---|---|
| Rigid 2D-objects with 3D-motion | 600 | 1300 | 15 | 6 |
| Flexible 2D-objects with 2D-motion | 1000 | 500 | 4 | 6 |

Demanding a low variance of the synthesis error, i.e. a high image quality of the synthesis, drastically increases all parameter data streams for the source model of rigid objects with three-dimensional motion, i.e., parameter coding becomes inefficient in a coder with this source model.

Based on these comparisons, the source model of flexible objects allows a more efficient parameter coding than does the model of rigid objects. The essential difference between the two source models is their definition of objects and hence their description of object motion. This aspect will be outlined in the following.

The introduction of a more complex motion description allowing arbitrary three-dimensional motion aims at two things. First, the modelling of the objects and hence their synthesis should become more precisely when compared to the motion description of the source model of flexible objects with two-dimensional motion. Secondly, the source model based on rigid objects with three-dimensional motion should decrease the redundancy of the motion information, as it efficiently describes motion by only a few parameters. These arguments will be discussed in the following.

As can be seen from Tables 3 and 4, the variance of the synthesis error for the source model of rigid objects is higher than for the source model of flexible objects. One reason is that motion such as rotation and linear deformation can be described precisely by a two-dimensional motion description if the resolution of the motion description is fine enough. Introducing three-dimensional motion descriptions does not necessarily improve the modelling. Additionally, the assumption of rigid objects does not usually hold for natural scenes, i.e., object motion in natural scenes always contains local flexible deformations which are superimposed onto the motion of a rigid object. These deformations are not modelled by the source model of rigid objects but they are well approximated by the source model of flexible objects.

As can be seen from Tables 3 and 4, the area of model failures is very small and the quality of the synthesized image is quite high in the case of the source model of flexible objects. This indicates that almost all temporal changes of successive images due to object motion and object deformations are nearly perfectly modelled by this source model. The remaining model failures are not due to motion but to other effects like shadows, reflections and new image content entering the scene. For that reason, it is assumed in the following that the source model of flexible objects approximates the 'true' motion more accurately. The question arising is how this motion description differs from the motion description generated by the source model of rigid objects.

Table 5 shows the $x$-component of the vector field of an object in front of stationary background calculated from two frames of the videophone sequence 'Miss America' based on the source model of flexible objects. The principal idea of the source model of rigid objects is to eliminate the redundancy of the motion information by describing the vector field of a whole object by only a few parameters. For comparison, the vector field from Table 5 has been approximated by a six parameter

Table 5

Pel-wise $X$-component of the vector field for one moving object in front of the stationary background calculated from the source model of flexible objects in ($\frac{1}{2}$ pel)

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 5 | 7 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 9 | 11 | 9 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 11 | 11 | 9 | 8 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 9 | 11 | 9 | 8 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 8 | 9 | 9 | 7 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 9 | 7 | 5 | 7 | 5 | 5 | −1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 7 | 6 | 6 | −3 | 3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 3 | 5 | 5 | 3 | 1 | −1 | −1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | −1 | −2 | −5 | −5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −3 | 1 | 1 | 1 | −9 | −5 | −5 | −5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | −7 | −4 | 0 | 0 | −3 | −1 | −1 | −3 | −5 | −5 | −5 | −4 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | −4 | −3 | −5 | −5 | −6 | −5 | −1 | −3 | −7 | −4 | −5 | −5 | −4 | −4 | −5 | −4 | 0 | 0 |
| 0 | 0 | 0 | 0 | −4 | −5 | −4 | −5 | −8 | −5 | −3 | −3 | −5 | −4 | −5 | −4 | −4 | −4 | −2 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | −4 | −5 | −6 | −4 | −5 | −5 | −5 | −7 | −4 | −4 | −3 | −4 | −4 | −4 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | −4 | −5 | −6 | −4 | −5 | −5 | −5 | −5 | −4 | −4 | −4 | −4 | −3 | −3 | 0 | 0 | 0 |
| −3 | −4 | 0 | 0 | −4 | −4 | −4 | −6 | −4 | −4 | −5 | −5 | −4 | −4 | −4 | −4 | −4 | −2 | −3 | 0 | 0 | 0 |

model calculated from a linear vector field regression used for the source model of rigid objects and known from [4, 12]. The $x$-component of the vector field generated by the six parameters corresponding to the vector field in Table 5 is shown in Table 6. The differences of the two vector fields are shown in Table 7. As can be seen from Table 6, the principal motion direction is well approximated, but as shown in Table 7, the local deviations of the six parameter motion description compared to the 'true' vector field are quite high.

Another method to reduce the redundancy of the motion information is a spatial prediction taking into account the statistical properties of the vector field, i.e., its local correlations. The coefficients of a spatial predictor are given in Table 8. The resulting coding gains of the six parameter approach compared to the statistical approach are shown in Table 9. As can be seen, the coding gain of both methods is nearly the same, i.e., the six parameter approach nearly eliminates the redundancy of motion information which is caused by the local correlation of adjacent vectors of the vector field. The six parameter motion description represents an approximation of all local motion within an object. Hence, the motion description of the source model of rigid objects works quite well for image analysis [11, 14] where it is important to find moving objects, i.e., where it is important to gather difference image areas and to attach them to the mapping of one moving object. These techniques work well for image analysis but they fail for image synthesis, as for image synthesis the true object motion is necessary in order to achieve an image synthesis without distortions. The higher the quality requirements for image synthesis and the higher the deviations between true motion and approximated object motion, the worse the source model of rigid objects works for image synthesis.

Table 6

Pel-wise $X$-component of the vector field corresponding to a six parameter motion description calculated from Table 5 by linear vector field regression in ($\frac{1}{2}$ pel)

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 9 | 9 | 9 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 8 | 8 | 8 | 8 | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | −1 | −1 | −1 | 0 | −1 | −1 | −1 | −1 | −1 | −1 | −2 | −2 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | −2 | −2 | −2 | −2 | −2 | −2 | −2 | −2 | −2 | −2 | −3 | −3 | −3 | −3 | −3 | −3 | 0 | 0 |
| 0 | 0 | 0 | 0 | −3 | −3 | −3 | −3 | −3 | −3 | −3 | −3 | −3 | −4 | −4 | −4 | −4 | −4 | −4 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | −4 | −4 | −4 | −4 | −4 | −4 | −4 | −5 | −5 | −5 | −5 | −5 | −5 | −5 | 0 | 0 | 0 |
| −5 | 0 | 0 | 0 | 0 | −5 | −5 | −5 | −5 | −5 | −5 | −6 | −6 | −6 | −6 | −6 | −6 | −6 | −6 | 0 | 0 | 0 |
| −6 | −6 | 0 | 0 | −6 | −6 | −6 | −6 | −6 | −6 | −6 | −7 | −7 | −7 | −7 | −7 | −7 | −7 | −7 | 0 | 0 | 0 |





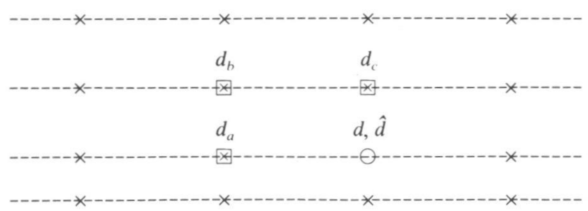
Table 7

Difference of the vector fields of Tables 5 and 6

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −2 | −4 | −2 | −4 | −9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −8 | −1 | 1 | 3 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −7 | 0 | 4 | 4 | 2 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −6 | 3 | 3 | 5 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −5 | 2 | 3 | 4 | 4 | 2 | −4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 3 | 1 | 3 | 1 | 1 | −5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −2 | 4 | 3 | 3 | −6 | 0 | −2 | −2 | −3 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −3 | 1 | 3 | 3 | 1 | −1 | −3 | −3 | −2 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 2 | −2 | −3 | −6 | −6 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −3 | 1 | 1 | 1 | −9 | −5 | −5 | −5 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | −6 | −3 | 1 | 0 | −2 | 0 | 0 | −2 | −4 | −4 | −3 | −2 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | −2 | −1 | −3 | −3 | −4 | −3 | 1 | −1 | −5 | −2 | −2 | −2 | −1 | −1 | −2 | −1 | 0 | 0 |
| 0 | 0 | 0 | 0 | −1 | −2 | −1 | −2 | −5 | −2 | 0 | 0 | −2 | 0 | −1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | −1 | −2 | 0 | −1 | −1 | −1 | −2 | 1 | 1 | 2 | 1 | 1 | 1 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 1 | 0 | −1 | 1 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 0 | 0 | 0 |
| 3 | 2 | 0 | 0 | 2 | 2 | 2 | 0 | 2 | 2 | 1 | 2 | 3 | 3 | 3 | 3 | 3 | 5 | 4 | 0 | 0 | 0 |

Table 8

Parameters of a spatial predictor used for predictive coding of motion vector components




$\hat{d} = h_a d_a + h_b d_b + h_c d_c$

with $h_a = 3/8$, $h_b = 1/8$, $h_c = 1/2$

Table 9

Comparison of the coding gain of the six parameter and the statistical motion description

Entropy of the vector field: 7.23 bits/vector

| Approach | Coding gain (bits/vector) |
|---|---|
| Six parameter | 0.81 |
| Statistical | 0.77 |

### *3.4. Detection of allowable geometrical distortions*

An important goal of parameter coding is to code only those parameter sets of an object which are important for a synthesis without visible distortions, and to suppress the coding of all these parameter sets which are irrelevant for a human observer.

One difficulty of parameter coding is the colour coding of synthesis errors which are caused by small position errors of the objects. These mismatches between a luminance signal $L_{k+1}$ and its corresponding synthesized luminance signal $\hat{L}_{k+1}$ generate synthesis errors which typically exhibit line structures as illustrated in Fig. 5. On one hand, these line structures are difficult to encode, on the other hand small position errors of objects in the synthesized image are not annoying, as a human observer pays attention to a natural motion of the object but not to its exact position. Therefore a large amount of data is used for the colour updating of image areas which however are not important for a human observer.

In order to suppress the colour updating of such areas, a simple technique has been developed to detect mismatches originated from small position errors of objects. Figure 6 shows the block diagram. Input to this detector are the actual luminance signal $L_{k+1}$ and the luminance signal $\hat{L}_{k+1}$ from image synthesis. These signals are subtracted and binarized using a threshold $T$. The binarized synthesis error image, denoted in the following as synthesis error mask, contains the synthesis errors which are generated by model failures as well as by small position errors of objects. By median filtering the synthesis error mask, line structures which are generated by small position errors of objects are eliminated while compact areas which are due to model failures remain unchanged. The areas of synthesis errors which are generated by small position errors of objects can thus simply be detected by subtracting the median filtered synthesis error mask from the synthesis error mask. In the following, the synthesis errors which are generated by small position errors of objects are attributed to geometrical distortions. In object-oriented analysis-synthesis coding, geometrical distortions are tolerated, i.e., they are not updated by colour information, in order to leave the remaining bit rate for the updating of image areas which contain model failures and hence are important for a human observer. For the case of a videophone scene, this corresponds to the updating of the face area of a speaking person.

The outlined concept of preferring position errors over quantization errors causes a certain percentage of the overall image area to be updated. Table 10 compares this with other update procedures.

As a first example the CCITT-Coder [19] without forward control is considered. The selection unit of this coder concept is a block, the selection criteria for updating colour information are the prediction error and the quantization

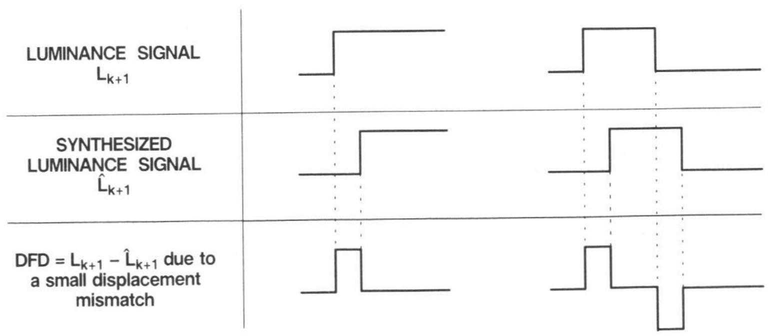



Fig. 5. Typical structure of the DFD (Displaced Frame Difference) due to a small displacement mismatch.





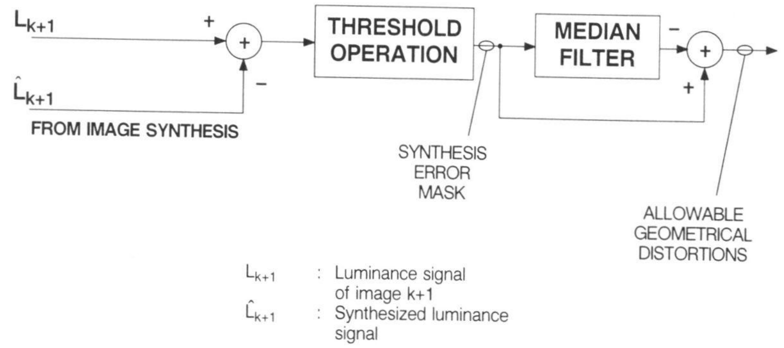



Fig. 6. Block diagram of a detector of allowable geometrical distortions.

characteristic. The quantization is variable and about 40% of the whole image area is updated. The main disadvantage of this coding concept is the coarse quantization caused by the large image area to be updated and the low data rate available, yielding a low resolution of the coded image area.

The next example combines the CCITT-Coder with a forward control [6]. The selection unit is again a block. The selection criteria are the prediction error and the available data-rate. The quantization is fixed and only 10% of the whole image is updated. The main disadvantage of this coding concept is that the coding of colour information is terminated when the available data rate is exhausted without taking into account whether all relevant image areas of synthesis errors have been updated.

The object-oriented analysis–synthesis coder presented in this paper is included as the third example in Table 10. The selection unit is now not a block but an object. The selection criteria are model failures and the data rate available. As mentioned in Section 3.1.2, the colour coding has not been realized in our coder concept until now. Principally, object-oriented analysis–synthesis coding allows to adopt the quantization to the validity of the source model, i.e., to the area of model failures. The more valid the source model, the smaller the area of model failure and hence the finer the quantizer stepsize yielding a high resolution of the coded image area. By variation of the quantizer stepsize, it can be guaranteed that all relevant image areas, i.e. all model failures, are updated. Hence the transmission of colour information is not determined by a fixed quantizer stepsize but

Table 10

Update procedures of different coder concepts

| Technique | Selection unit | Selection criterion | Quantization | Percentage of updated area |
|---|---|---|---|---|
| CCITT-Coder without forward control | Block | Prediction error + quantization characteristic | Variable | 40% |
| CCITT-coder with forward control | Block | Prediction error + data rate | Fixed | 10% |
| Object-oriented analysis–synthesis coder | Object | Model failure + data rate | Variable | 4% |

by the image area of model failures to be updated. Another advantage of this coding concept is that, allowing geometrical distortions and introducing the transmission of shape information, the image area to be updated can be decreased to 4% of the whole image area.

## 4. Experimental results

The previously described analysis-synthesis coder based on the source model of flexible two-dimensional objects with two-dimensional motion has been experimentally investigated by means of computer simulations. The CCITT test sequence 'Miss America' has been used with a reduced field frequency of 10 Hz and a quantization of 8 bit per sample. Each of the frames 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. Figures 7 and 8 show two successive images of the sequence where closing eyes and an opening mouth are superimposed onto head motion.

Figure 9 shows the result of image analysis obtained from these two images. As can be seen from Fig. 9, model failures and hence the image area to be updated by colour information are concentrated on eyes and mouth, i.e., on image areas which are important for a human observer.

Figure 10 shows the bit distribution plots of motion and shape information generated by the coder for the whole sequence. The motion vectors have been predictively coded as described in Section 3.1.2. For shape coding, the quality measure $d_{max}$ has been chosen to 2.9 pels corresponding to a maximum absolute distance of 2 pels in horizontal and vertical direction between the original and the approximated shape. This value seems to be sufficient for a high quality shape approximation [7]. As can be seen from Fig. 10, the total amount of shape and motion data is about 1500 bits. Thus, 4000-5000 bits remain for the colour updating of about 4% of the whole image area corresponding to about 1 bit available per pixel to be updated by colour information.

Fig. 7. Reference image from the test sequence 'Miss America'.





Fig. 8. Successive actual image from the test sequence.

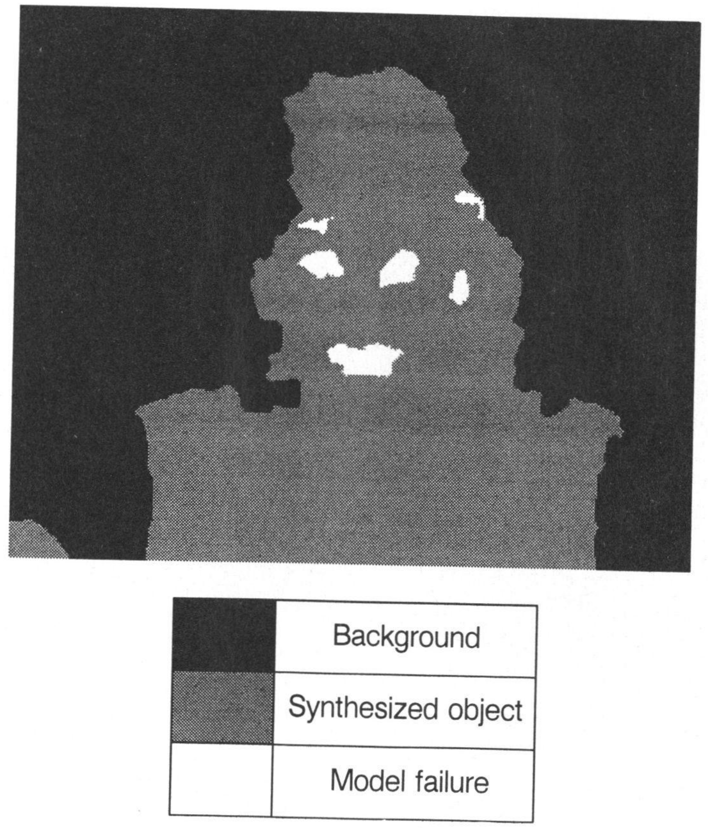



Fig. 9. Shapes of the objects found by image analysis of Figs. 7 and 8.

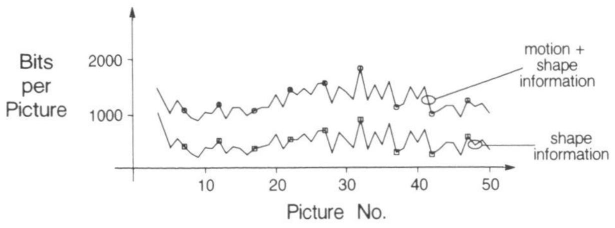



Fig. 10. Bit distribution plots of motion and shape information for the test sequence 'Miss America'.

Figures 11–13 show the 10th, 20th and 30th synthesized image of the squence. As the colour coding has not yet been realized for the coder concept, original 8 bit PCM colour values are substituted in areas of model failure. As can be seen, a high image quality is preserved in image areas of the face which are important for a human observer. Further, disturbances known from the CCITT-Coder as mosquito effects are eliminated by the introduction of shape information transmission. Remaining errors are caused by the repeated image synthesis without colour update and aliasing disturbances at object boundaries.

The problem of repeated image synthesis without colour update information can be outlined as follows: let us assume that luminance changes of successive images are only due to motion, i.e., that theoretically image synthesis can perfectly be performed by motion and shape information only. The approach of image synthesis used in this paper is image synthesis by filter concatenation as outlined in Fig. 14. If luminance changes of successive images are only due to motion, an image $I_1$ to be synthesized is identical to the image $I_0$ at a displaced position. As this displaced position does not necessarily coincide with the sampling grid of $I_0$, the image $I_0$ has to be filtered in order to interpolate the missing samples. The image $I_n$ to be synthesized is then calculated by means of the previous synthesized image $I_{n-1}$ and the actual transmitted motion and shape parameters. The main disadvantage of image synthesis by filter concatenation is that the synthesis errors due to the interpolation filter accumulate. In order to avoid this accumulation of synthesis errors, an image synthesis by parameter concatenation is actually under study as outlined in Fig. 15. For this approach, an image $I_n$ is synthesized by means of the stored colour parameters $S_0$ and an appropriate addition of all motion and shape parameters which

Fig. 11. 10th image synthesized.





Fig. 12. 20th image synthesized,

have been transmitted. The main advantage of this technique compared to image synthesis by filter concatenation is that for each image to be synthesized the interpolation filter is only once applied.

The aliasing disturbances can be eliminated, if the contour information is taken into account with subpel accuracy and contour filtering as known from computer graphics is applied.

Fig. 13. 30th image synthesized.

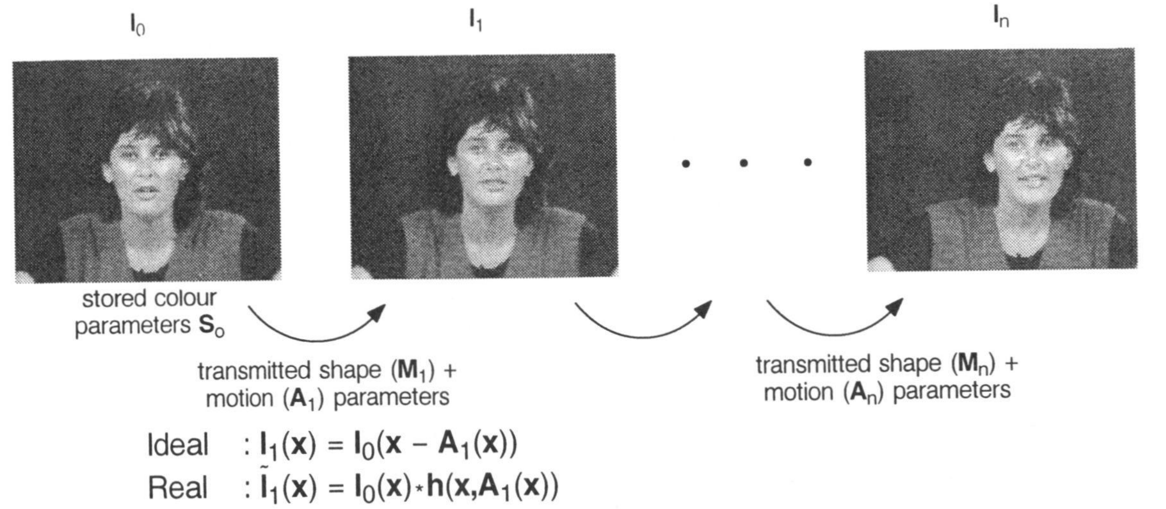



Fig. 14. Image synthesis by means of transmitted motion and shape parameters and stored colour parameters. Assumption: Luminance changes of successive images are only due to motion. Approach: Filter concatenation.

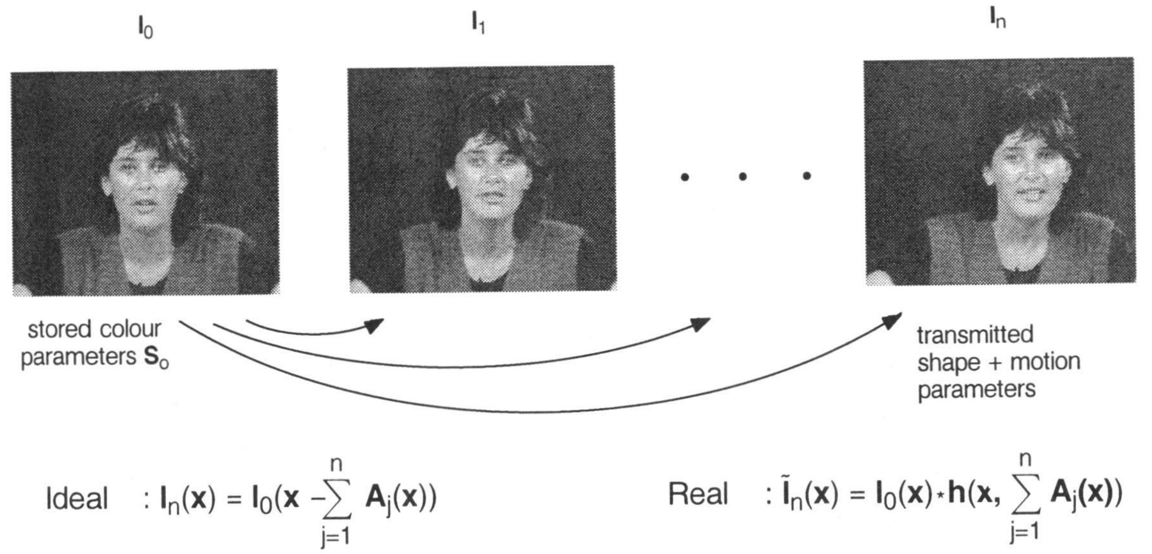



Fig. 15. Image synthesis by means of transmitted motion and shape parameters and stored colour parameters. Assumption: Luminance changes of successive images are only due to motion. Approach: Parameter concatenation.

Both problems will be dealt with in further researches.

## 5. Conclusion

The structure of an object-oriented analysis–synthesis coder has been described which segments an image into objects and encodes each object by three parameter sets. Two different source models both based on the model of moving two-dimensional objects, have been compared with respect to their effectiveness for parameter coding, first a model of rigid objects with three-dimensional motion and second a model of flexible objects with two-dimensional motion. It could be seen that the data rate for motion information is very low in the case of rigid objects. This advantage has to be paid for by a large image area of model failures and a low quality of synthesized objects. In contrast to this, the source model of flexible objects generates quite a high data-rate for the motion information which, however, is more than balanced by an image synthesis of high quality and a small image area of model failures. As a conclusion of the comparison, the source model of flexible objects seems to be more appropriate for parameter coding. The main reason for the different properties of the two source models are the different optimizing criteria necessary for image analysis and image synthesis. The source model of rigid objects works well for image analysis, since an important goal of image analysis is to find image areas which can approximately be attached to the mapping of one moving object. Image synthesis does not need an average object motion, but it needs a locally exact motion description in order to guarantee an image synthesis without distortions.

Introducing object shape transmission and allowing geometrical distortions due to small errors of an object position yield a coder concept whose essential advantages compared to the CCITT-Coder are based on the following properties:

—By the transmission of shape information, the quality of the image synthesis is drastically increased in image background areas, which border on moving objects. Annoying coding errors known from the CCITT-Coder as mosquito effects are eliminated by the new coder concept.

—In the CCITT-Coder concept about 40% of the image area is updated by colour coding. Allowing small position errors of objects and transmitting the shape information, this amount can be reduced to 4% of the image area without introducing annoying distortions.

—Owing to the efficient coding of the shape and motion information about 80% of the data-rate is available for the coding of the colour updating. This corresponds to a data rate of about 1 bit per pixel compared to 0.1 bit per pixel for the CCITT-Coder, i.e., the resolution of the image area updated by colour coding is higher than for the CCITT-Coder.

The colour coding of model failures and the treatment of distortions as caused by a repeated image synthesis without colour update and aliasing effects at object boundaries will be dealt with in the future.






## Acknowledgments

The author wishes to thank Professor Musmann for encouraging this work and for many helpful discussions about the field of parameter coding. Further, the author wishes to thank Mr. Bierling for his software support and his suggestions concerning the subject of true object motion and hierarchical blockmatching. This work has been supported by the Forschungsinstitut der Deutschen Bundespost beim Fernmeldetechnischen Zentralamt, Darmstadt.


## References


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

[2] 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 Processing*, Vol. 8, No. 4, July 1985, pp. 381–386.

[3] U. Franke, "Regionenorientierte Bildbeschreibung—Algorithmen und Möglichkeiten", *Ph.D. Dissertation*, RWTH Aachen 1988. Fortschritt-Berichte VDI, Reihe 10, Nr. 101, Aachen, West Germany, 1989.

[4] A. Gerhard, "Bewegungsanalyse bei der Codierung von Bildsequenzen", *Ph.D. Dissertation*, Techn. Univ. München, München, West Germany, 1988.

[5] M. Gilge, T. Engelhardt and R. Mehlan, "Coding of arbitrarily shaped image segments based on a generalized orthogonal transform", *Signal Processing: Image Communication*, Vol. 1, No. 2, October 1989, pp. 153–180.

[6] K. Grotz, J. U. Mayer and G.K. Suessmeier, "A 64 kbit/s videophone codec with forward analysis and control", *Signal Processing: Image Communication*, Vol. 1, No. 2, October 1989, pp. 103–115.

[7] K.-W. Hahn, "Untersuchungen zur strukturierten Beschreibung und Codierung von Bildsegmenten für ein Konturcodiersystem", *Diplomarbeit Univ. of Hannover*, Hannover, West Germany, July 1989.

[8] M. Hötter, "Untersuchungen zur Analyse und strukturierten Beschreibung von Segmenten für ein Kontur/Textur Bildcodiersystem", *RWTH Aachen*, Aachen, West Germany, May 1986.

[9] M. Hötter, "Image segmentation into areas of uniform mapping parameters", *5th Workshop on Multidimensional Signal Processing*, Noordwijkerhout, The Netherlands, September 1987.

[10] M. Hötter and J. Ostermann, "Analysis-synthesis coding based on the model of planar, rigid, moving objects", *1st International Workshop on "64 kbit/s Coding of Moving Video"*, Hannover, West Germany, June 1988.

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

[12] G. Kummerfeldt, F. May and W. Wolf, "Coding television signals at 320 and 64 kbit/s", *2nd Internat. Tech. Symposium on Optical and Electro-Optical Applied Science and Engineering*, Cannes, France, December 1985, pp. 119-128.

[13] F. May, "Codierung von Bildfolgen mit objektbezogener Bewegungskompensation und Signifikanzklassifikation", *Ph.D. Dissertation*, Univ. of Karlsruhe, Karlsruhe, West Germany, 1984.

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

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

[16] E. Persoon and K.S. Fu, "Shape discrimination using Fourier descriptors", *IEEE Trans. Systems Man Cybernet.*, Vol. SMC-7, 1977, pp. 170-179.

[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] P. Spoer, "Schätzung der 3-dimensionalen Bewegungsvorgänge starrer, ebener Objekte in digitalen Fernsehbildfolgen mit Hilfe von Bewegungsparametern", *Ph.D. Dissertation*, Univ. of Hannover, Hannover, West Germany, 1987.

[19] CCITT SG XV, Doc. #525, "Description of Ref. Model 8 (RM8)".