

# TECHNICAL RESEARCH REPORT

## Scalable Coding of Video Objects

*by R. Haridasan, J. Baras*

**CSHCN T.R. 98-1  
(ISR T.R. 98-9)**



*The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.*

**Web site <http://www.isr.umd.edu/CSHCN/>**

# Scalable Coding of Video Objects

Radhakrishnan Haridasan & John S. Baras  
Department of Electrical Engineering &  
Institute for Systems Research  
A. V. Williams Building  
University of Maryland  
College Park, MD 20742  
E-mail: krishnan,barasisr.umd.edu

## Abstract

This paper provides a methodology to encode video objects in a scalable manner with regard to both content and quality. Content scalability and quality scalability have been identified as required features in order to support video coding across different environments. Following the object-based approach to coding video, we extend our previous work on motion-based segmentation by using a time recursive approach to segmenting image sequences and decomposing a video "shot" into its constituent objects. Our formulation of the segmentation problem enables us to design a codec in which the information (shape, texture and motion) pertaining to each video object is encoded independent of the other. The multiresolution wavelet decomposition used in encoding texture information is shown to be helpful in providing spatial scalability. Our codec design is also shown to be temporally scalable.

## 1 Introduction

Current research in source coding of video is directed toward efficient coding of image sequences to produce acceptable quality at very low bit rates. The rapid proliferation of multimedia applications owing to the convergence of telecommunications, computers and TV/film industries coupled with the importance of video in multimedia has resulted in new functionalities such as content-based interactivity to be defined. In this context, object-based coding of video performs better than conventional block-based approaches at low-bit rates owing to the fact that it employs more realistic source models for extracting structure and motion. Since video frames are divided into more meaningful entities that represent the actual contents of the scene, this approach can provide access to content. To support this content-based functionality across various environments two kinds of scalabilities - *content scalability* and *quality scalability* have been identified as required features of the emerging MPEG-4 standard [1]. Content scalability refers to the fact that objects are encoded in such a way that selective decoding and manipulation is possible at the receiving end. Quality scalability implies that the same source stream can be received at various levels of quality, in terms of spatial or temporal resolution. This paper provides a methodology to encode video objects in a scalable manner with regard to both content and quality. It is based on our previous work on motion-based segmentation of video.

Object-based coding of video is an approach which uses a source model to decompose video into moving objects, each of which is represented by its shape, motion and texture. Parameters corresponding to each of the three components are encoded and transmitted, and video frames are reconstructed at the decoder by synthesizing each object. It was first introduced by Musmann et. al. [2] as a means of reducing the bit rate required for coding video while avoiding blocking and mosquito effects characteristic of low bit rate codecs based on waveform coding. The three main issues to be addressed in object-based coding are segmentation, partition (or boundary) coding and region coding. Motion is the most commonly used attribute to distinguish objects from one other.

In [3] we have adopted an approach to motion-based segmentation which is based, more appropriately, from the viewpoint of the objects that comprise the scene. The specific formulation can be viewed as being akin to the view-based approach in computer vision. It uses the fact that the same object manifests itself in different frames while (probably) differing in pose owing to changes in camera perspective. Object motion plays a more fundamental role of connecting the different views as opposed to reducing temporal redundancies. The nature of the formulation allows new-objects and uncovered/covered backgrounds to be accounted for as outliers to motion models instead of being discovered after post-processing as in [4] or by special handling as in [5]. This implies that decisions regarding intra-mode or inter-mode coding of objects can be taken after the segmentation procedure itself which makes the object coding more modularized and well-suited for content-based applications. In particular, as shown in this paper, it is possible to code object information on a per-object basis, in a manner that provides both content and quality scalability. The organization of this paper is as follows. We first summarize our previous work on motion-based segmentation in Section 2. Section 3 shows how our approach to segmenting images can be extended to object-based video coding. Section 4 details how the process of obtaining video objects can itself be used to encode them independent of one another. Section 5 discusses the scalability aspects of the codec designed in Section 4. Finally, we provide a summary.

## 2 Motion-based Segmentation

This paper is based on our previous work on motion-based segmentation, which can be assumed to form the front end for the current system. In [3] we have formulated the motion-based image segmentation problem as a Maximum-Likelihood (ML) estimation problem using multiple parametric motion models and support maps. The motion-based segmentation algorithm effectively uses two successive frames  $I(\mathbf{x}, t - 1)$  and  $I(\mathbf{x}, t)$  to partition the current frame,  $I(\mathbf{x}, t)$ , into two kinds of objects, namely *model compliant* (MC) objects and *model failure* (MF) objects. MC objects are those that have manifested themselves in both frames, and can be predicted using parametric motion models on the basis of the motion that took place in  $[t - 1, t]$ . MF objects are those that do not conform to motion models. They include newly appearing objects, uncovered regions and

regions that do not conform to a parametric motion model. The region of support for the  $k$ th object (MC or MF) in the current frame is indicated using the binary variable  $s^k(\mathbf{x}, t)$ . MC objects can be predicted based on their appearance in the previous frame,  $I(\mathbf{x}, t-1)$ , using a displacement vector  $\mathbf{u}_{\mathbf{a}^k(t)}(\mathbf{x})$ . They are, therefore, associated with a parametric motion vector  $\mathbf{a}^k(t)$  in addition to the support map  $s^k(\mathbf{x}, t)$ . The algorithm in [3] simultaneously estimates the support map and the parametric motion vector for all objects occurring in  $I(\mathbf{x}, t)$ . Fig. 1(a) shows two successive frames and Fig. 1(b) shows the support map for the model compliant and model failure objects.

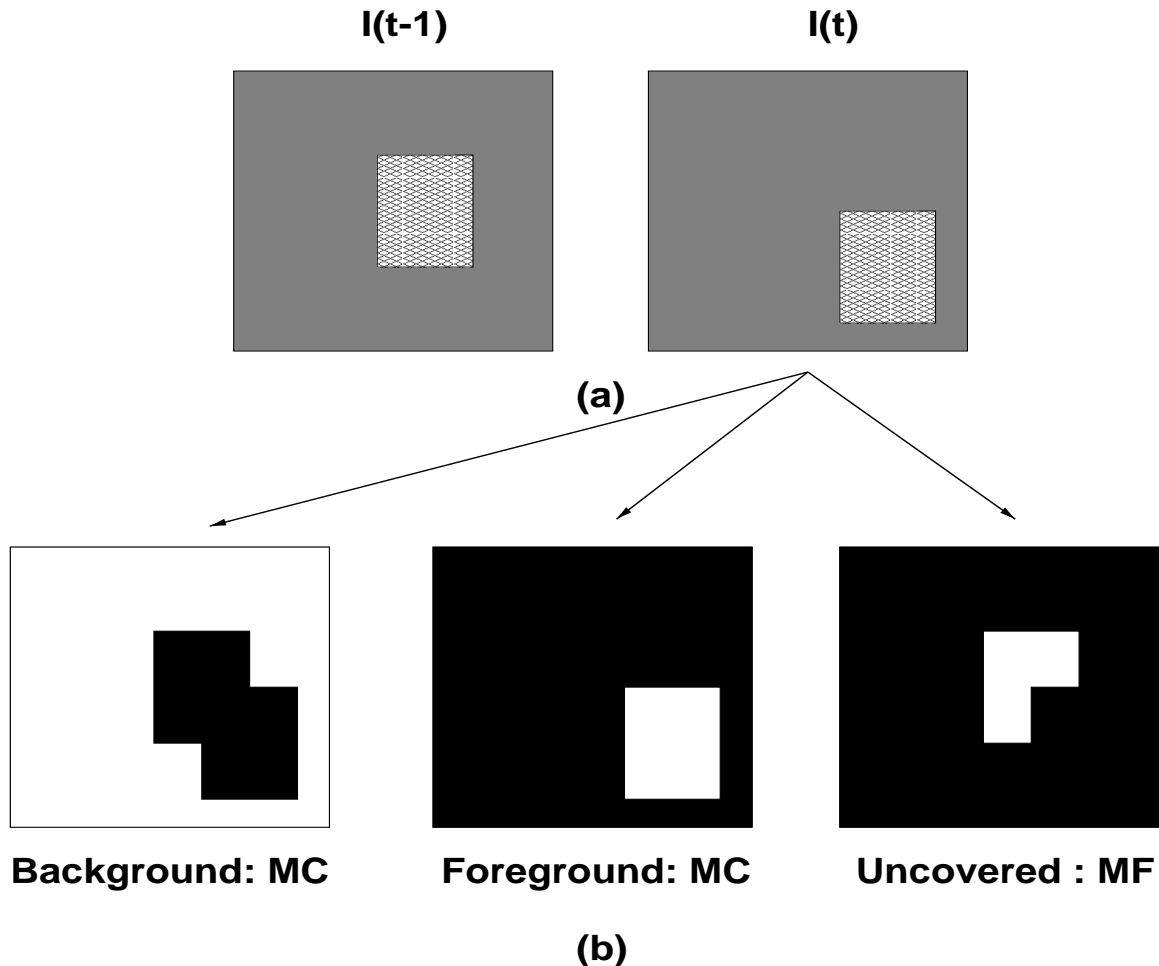


Figure 1: (a) Successive frames (b) Support regions of objects

### 3 Extension to Object-based Video Coding

A video sequence can be organized in terms of *shots*, *scenes* and *segments* [6]. Each *shot* is an unbroken sequence of frames from one camera, e.g. a zoom of a person talking. If we confine

ourselves to coding a shot, we can identify physical objects that manifest themselves in successive frames while (probably) differing in pose due to changes in camera perspective relative to the object. Following the time-recursive approach to segmenting image sequences, where successive frames are dealt with in overlapping pairs, and using the motion-based segmentation approach of the previous section we can decompose a shot into video objects, provided that we can ensure the object labeling (the  $\#k$ ) corresponds to the same physical object in time. For the purpose of this paper we assume that such a time-coherent segmentation has been obtained and focus on coding the video objects.

Since the motion-based segmentation algorithm decomposes a frame into its constituent objects it can provide access to content. In terms of MPEG-4 terminology [1], it can be seen that the support map  $s^k(\mathbf{x}, t)$  for the  $k$ th object corresponds to the binary alpha map of its Video Object Plane (VOP $_k$ ). The mechanism by which these VOPs are obtained can be used to our advantage in coding video objects (VO). Section 4 details a methodology of coding the information (shape, texture and motion) pertaining to each VO independent of others, thus providing content-based scalability. The specific encoding algorithm employed in coding object texture is seen to provide quality scalability<sup>1</sup> as well.

## 4 Coding of Video Objects

We detail our approach by confining ourselves to the  $k$ th video object. The first appearance of the  $k$ th object is on account of a model failure. Hence, the shape and texture information needs to be coded in an intra-mode (I-mode). All subsequent appearances of the  $k$ th object can be inter-frame coded in a predictive manner (P-mode) based on estimated motion. The following notation is used in the rest of the paper. Predicted values are denoted using  $\hat{\phantom{x}}$  while decoded values are denoted using  $\tilde{\phantom{x}}$ .  $\tilde{\phantom{x}} - \hat{\phantom{x}}$  is used to denote prediction errors.

### 4.1 I-mode Encoding

An object to be encoded in I-mode is characterized by its shape (or boundary) and texture. Shape information is sent first, followed by texture information. The region of the current frame,  $I(\mathbf{x}, t)$ , to be encoded is specified by the binary  $s^k(\mathbf{x}, t)$ . The texture information, denoted by  $I_I^k(\mathbf{x}, t)$ , is obtained by the pixel-wise multiplication of the support region with the current frame, i.e. ,

$$I_I^k(\mathbf{x}, t) = s^k(\mathbf{x}, t) \odot I(\mathbf{x}, t) \tag{1}$$

Shape information can be encoded losslessly, either by computing the chain code of the boundary of the object and applying entropy coding on the symbols or by using a Modified Modified MMR

---

<sup>1</sup>Ability of a coder to selectively enhance the quality of a video object by spending more bits on the object.

(M<sup>4</sup>R) coding. The texture of the object can be encoded at a specified bit-rate using the region-based wavelet coding technique proposed in [7] in which a wavelet decomposition of the frame is followed by embedded zero tree quantization and arithmetic coding [8]. Better performance can be obtained using the more recent implementation based on SPIHT [9]. Only wavelet coefficients pertaining to the region of support need to be sent on account of the spatial localization property of the wavelet coefficients. The regions of support at the higher levels of the pyramid decomposition are obtained by down-sampling the full resolution chain coded region of support. The decoded version of the object texture stored in the object memory is denoted by  $\check{I}(\mathbf{x}, t)$  and is assigned a zero motion vector, i.e.,  $\check{\mathbf{a}}^k(t) = \mathbf{0}$ . These values are used for predicting the object at the next time instance.

## 4.2 P-mode Encoding

An object to be encoded in P-mode is characterized by a parametric motion vector  $\mathbf{a}^k(t)$ , a region of support  $s^k(\mathbf{x}, t)$  and texture,  $I_P^k(\mathbf{x}, t)$ . The fact that the object has appeared at the previous time instance, along with the mechanism of obtaining video objects can be used to encode the information pertaining to the object as an incremental update. We can do so because the prediction information is already available to the decoder based on the decoded values at the previous time instance,  $t - 1$ . The decoder needs to be present in the encoder feedback loop to ensure that the prediction is consistent with that at the decoding end. For each object, motion vector information is encoded first, followed by contour information and texture information.

### 4.2.1 Motion Vector Coding

The parametric motion vector is represented using a fixed number of bits. More bits are assigned to represent the translation component as compared to the remaining components of the motion vector. The motion vector can then be encoded using DPCM encoding since a prediction vector,  $\hat{\mathbf{a}}^k(t)$ , can be formed from the decoded motion vector at the previous time instance (See Fig. 2). A simple model for prediction assumes that the motion vector does not change drastically and uses  $\hat{\mathbf{a}}^k(t) = \check{\mathbf{a}}^k(t - 1)$ . More complex models, based on Kalman filter can also be incorporated depending on whether such models are employed by the motion-based segmentation algorithm.

### 4.2.2 Support Region Coding

The region of support,  $s^k(\mathbf{x}, t)$ , to be encoded in P-mode is indicative of those pixels in the current frame that can be predicted from the previous frame,  $I(\mathbf{x}, t - 1)$  using the parametric motion vector.

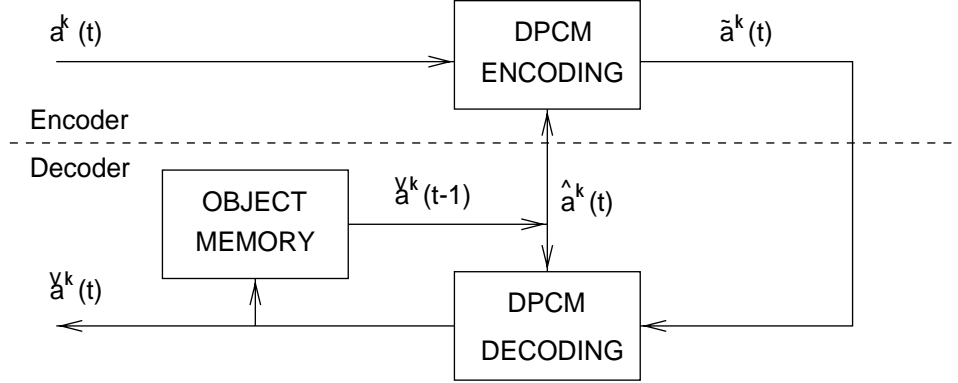


Figure 2: Motion Vector Encoder

It can be decomposed into two parts as

$$s^k(\mathbf{x}, t) = \hat{s}^k(\mathbf{x}, t) + \tilde{s}^k(\mathbf{x}, t) \quad (2)$$

where,  $\hat{s}^k(\mathbf{x}, t)$  is the predicted support region and  $\tilde{s}^k(\mathbf{x}, t)$  is the contour prediction error. The predicted region need not be sent as it can be computed at both the encoder and the decoder as

$$\hat{s}^k(\mathbf{x}, t) = \text{warp}(\check{s}^k(\mathbf{x}, t-1), \check{\mathbf{a}}^k(t)) \quad (3)$$

$$= \check{s}^k(\mathbf{x} - \mathbf{u}_{\check{\mathbf{a}}^k(t)}(\mathbf{x}), t-1) \quad (4)$$

where  $\check{s}^k(\mathbf{x}, t-1)$  is the decoded support region at the previous time instance and  $\check{\mathbf{a}}^k(t)$  is the decoded motion vector at the current time instance. Since motion vector information is made available by suitable encoding for use by both the shape and texture decoding process, only the contour prediction error needs to be sent.  $\tilde{s}^k(\mathbf{x}, t)$  is encoded just as in the I-mode. Fig. 3 shows one possible way of losslessly encoding this information.

### 4.2.3 Texture Coding

The texture of the object encoded in P-mode is denoted by  $I_P^k(\mathbf{x}, t)$  and is given by

$$I_P^k(\mathbf{x}, t) = s^k(\mathbf{x}, t) \odot I(\mathbf{x}, t) \quad (5)$$

$$= s^k(\mathbf{x}, t) \odot I(\mathbf{x} - \mathbf{u}_{\mathbf{a}^k(t)}(\mathbf{x}), t-1) \quad (6)$$

Since the entire frame  $I(\mathbf{x}, t-1)$  may not be available at the decoding end<sup>2</sup>, we can decompose the texture of the object into two parts. The first part is a prediction,  $\hat{I}_P^k(\mathbf{x}, t)$  and is based on the

<sup>2</sup>This is the case when only a part of the previous frame has been decoded owing to selective decoding of objects.

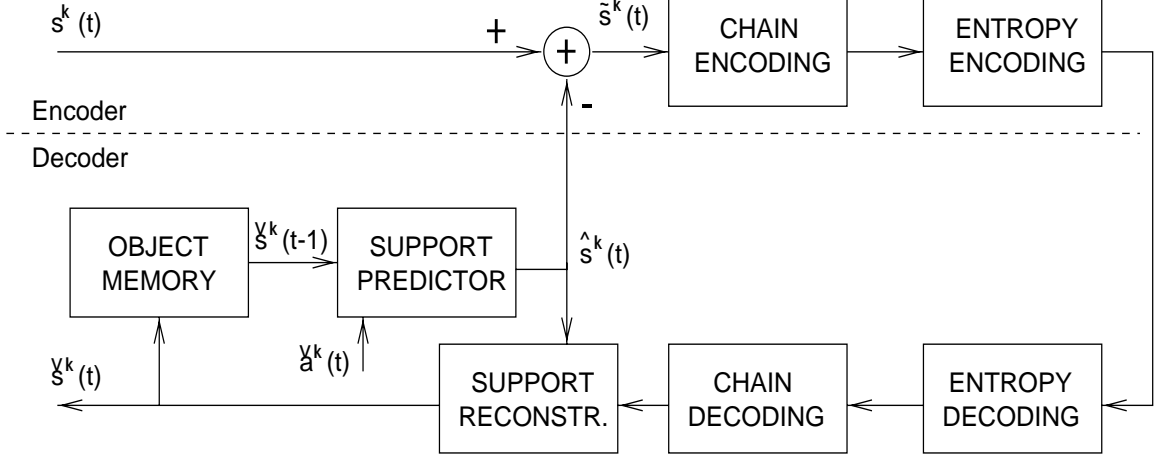


Figure 3: Support Region Encoder

decoded motion vector  $\hat{\mathbf{a}}^k(t)$  and the previous instantiation of the object at the decoder  $\check{I}_X^k(\mathbf{x}, t-1)$  and need not be sent. It is computed at both the encoder and decoder as

$$\hat{I}_P^k(\mathbf{x}, t) = \check{s}^k(\mathbf{x}, t) \odot \check{I}_X^k(\mathbf{x} - \mathbf{u}_{\hat{\mathbf{a}}^k(t)}(\mathbf{x}), t-1) \quad (7)$$

where  $\check{I}_X^k(\cdot, \cdot)$  stands for  $\check{I}_I^k(\cdot, \cdot)$  or  $\check{I}_P^k(\cdot, \cdot)$ , as the case may be.

The second part is an update or a residual defined as

$$\tilde{I}_P^k(\mathbf{x}, t) = I_P^k(\mathbf{x}, t) - \hat{I}_P^k(\mathbf{x}, t) \quad (8)$$

which is coded just as in the I-mode, by using wavelet decomposition followed by quantization and arithmetic coding (See Fig. 4). To achieve higher efficiency in coding residual information we can adopt a cleaning procedure consisting of thresholding (to eliminate residues insignificant with respect to the current quantization step), morphological opening, low-pass filtering (or smoothing) and gray-level restoration (converting from binary to the original gray-scale). This procedure ensures that bits are spent only on visually significant regions.

## 5 Scalability of the object-based Codec

The previous section has shown that it is possible to design an encoder-decoder combination which can code information on a per-object basis, independent of one another. Hence, our object-based codec is content scalable. This feature supports *object scalability*, i.e., the ability to selectively encode/decode specific objects and achieve a lower bit rate when a lesser number of objects are decoded. In this section we show that our codec is also quality scalable, in terms of supporting various spatial and temporal resolutions. Quality scalability is made possible by layering the information to be transmitted in terms of a base layer and enhancement layer(s).



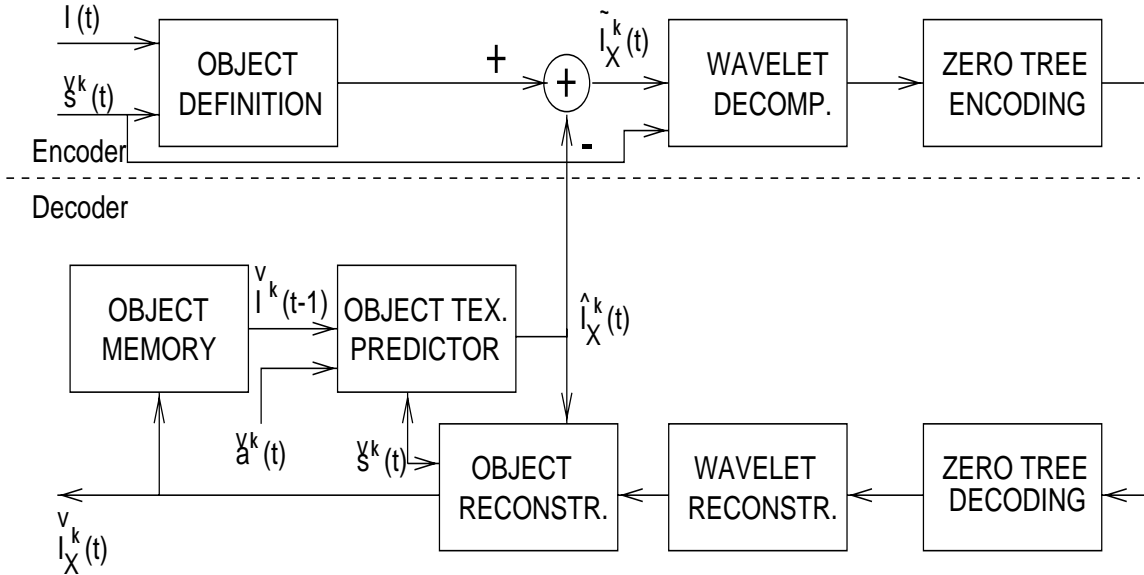


Figure 4: Texture Encoder

## 5.1 Spatial scalability

Spatial scalability refers to the ability to decode a portion of the bit stream to generate object/video at lower spatial resolutions. The use of a multiresolution wavelet decomposition (EZW or SPIHT) in encoding texture information allows our codec to be spatially scalable. To obtain a spatially scalable bitstream, the conventional embedded zero-tree coders need to be modified slightly. Instead of encoding the wavelet coefficients across multiple resolutions at a desired bit rate, the zero-tree encoder is changed so as to yield a bit stream that is layered in spatial resolution. Specifically, this means that at a given bit rate although the wavelet coefficients across different resolutions are quantized, in sending the information the wavelet coefficients corresponding to the lowest resolution is sent first (yielding a base layer) followed by the next higher resolution and so on[10]. This layering makes it possible for the decoder to receive the texture information at a level commensurate with its requirement or capability. The motion vector and support map information are always, however, sent at the maximum resolution and are scaled or down-sampled to obtain the information at lower levels of spatial resolution.

## 5.2 Temporal Scalability

Temporal scalability is the ability to decode objects/video at different frame rates. Since each VO is coded independent of the other it is possible to receive each VO at a different temporal resolution. This is made possible by encoding successive frames corresponding to a particular VO in a layered manner. Our method of encoding VOs based on motion estimation allows us to support a base

layer temporal resolution and a continuum of enhancement layers (up to the incoming frame rate of the video). Given successive VOP frames and a base frame rate of  $1/N$ , we can perform motion estimation every  $N$  VOP frames to obtain a base layer coding of the object. Higher frame rates can then be supported by sending the object information (shape, texture and motion) in an incremental manner with the corresponding base layer information used for prediction.

## 6 Summary

The motion-based segmentation algorithm developed previously was used to provide content access to video by decomposing shots of a video into its constituent objects. The mechanism of extracting video objects was found to be useful in coding objects. In particular it was shown that it is possible to encode objects independent of one another. The use of a multiresolution wavelet decomposition in encoding texture enabled the codec to be spatially scalable. It was also shown that temporal scalability can be obtained. Thus, we have provided a methodology to encode video objects in a content and quality scalable manner. Such a representation is expected to play an important role in low bit rate coding of video.

## References

- [1] T. Sikora, "The MPEG-4 Video Standard Verification Model", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 1, pp. 19-31, February 1997.
- [2] H.G. Musmann, M. Hotter and J. Osterman, "Object-oriented Analysis-synthesis Coding of Moving Images", *Signal Processing: Image Communication*, Vol 1, pp. 117-138, October 1989.
- [3] R. Haridasan and J. S. Baras, "Accurate Segmentation and Estimation of Parametric Motion Fields for Object-based Video Coding Using Mean Field Theory", *SPIE Conference on Visual Communications and Image Processing*, San Jose, California, January 1998.
- [4] S.C. Han, *Object-based Representation and Compression of Image Sequences*, Ph.D. thesis, Department of Electrical Engineering, Rensselaer Polytechnic Institute, 1997 .
- [5] P. Bouthemy and E. Francois, "Motion Segmentation and Qualitative Dynamic Scene Analysis from an Image Sequence", *International Journal of Computer Vision*, Vol. 10, No. 2, pp. 157-182, 1993.
- [6] R. Picard, "Light-years from Lena: Video and Image Libraries of the Future", *Proc. of International Conference on Image Processing, ICIP95*, Washington, DC, October 1995.

- [7] K. Oehler, "Region-based Wavelet Compression for Very Low Bit-rate Video Coding", *Proc. of International Conference on Image Processing, ICIP96*, Lausanne, Switzerland, pp. 573-576, September 1996.
- [8] J. Shapiro, "Embedded Image Coding Using Zero-trees of Wavelet Coefficients, *IEEE Transactions on Image Processing*, Vol. 4, pp. 3445-3462, December 1993.
- [9] A. Said and W. A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 6, pp. 243-250, June 1996.
- [10] Z. Xiong, B-J. Kim and W. A. Pearlman, "Multiresolutional Coding/ Decoding in Embedded Image and Video Coders", *IEEE Signal Processing Letters* (to appear).