Adaptive Coding of Moving Objects for Very Low Bit Rates

Soo-Chul Han and John W. Woods, Fellow, IEEE

Abstract—This paper describes an object-based video coding system with new ideas in both the motion analysis and source encoding procedures. The moving objects in a video are extracted by means of a joint motion estimation and segmentation algorithm based on the Markov random field (MRF) model. The two important features of the presented technique are the temporal linking of the objects, and the guidance of the motion segmentation with spatial color information. This facilitates several aspects of an object-based coder. First, a new temporal updating scheme greatly reduces the bit rate to code the object boundaries without resorting to crude lossy approximations. Next, the uncovered regions can be extracted and encoded in an efficient manner by observing their revealed contents. The objects are classified adaptively as P objects or I objects and encoded accordingly. Subband/wavelet coding is applied in encoding the object interiors. Simulations at very low bit rates yielded comparable performance in terms of reconstructed PSNR to the H.263 coder. The object-based coder produced visually more pleasing video with less blurriness and devoid of block artifacts, thus confirming the advantages of object-based coding at very low bit-rates.

Index Terms—Image coding, image motion analysis, video signal processing.

I. INTRODUCTION

VIDEO compression at very low bit rates has attracted considerable attention recently in the image processing community. This is due to the expanding list of very low bit-rate applications, including video conferencing, multimedia, video over existing networks, and wireless communication. This is further evidenced by the ongoing MPEG-4 standardization process [1].

The bulk of the very low bit-rate coding research has centered on so-called second-generation techniques [2] because existing standards such as H.261 and MPEG perform rather poorly at lower rates. The main inherent deficiency with these block-based approaches is that the video scene and motion are modeled as being comprised of square blocks. Thus, at very low bit rates, the coding artifacts appear as the well-known blocking artifacts, which are very disturbing to the human observer. The second-generation techniques aim to alleviate this problem by setting up models more true to the scene contents. Furthermore, such models can enable a better interpretation of the image/video contents, thus leading to other advantages such as improved browsing, scalability, and content-based manipulation.

A natural alternative to the block-based standards is object-based coding, first proposed by Musmann et al. [3]. In the object-based approach, the moving objects in the video scene are extracted, and each object is represented by its shape, motion, and texture. Parameters representing the three components are encoded and transmitted, and the reconstruction is performed by synthesizing each object. Although a plethora of work on the extraction and/or coding of the moving objects has appeared since [3], only a handful of them carry out the entire analysis-coding process from start to finish. Thus, the widespread belief that object-based methods could outperform standard techniques at low bit rates (or any rates) has yet to be firmly established. In this paper, we attempt to take the step in that direction with new ideas both in the motion analysis and the source encoding procedures.

Several roadblocks have prevented an overall object/region-based coding system from outperforming standard block-based techniques. Most of these issues are active areas of research today. For one thing, the extraction of moving objects, such as by means of segmentation, is known to be an ill-posed problem [4]. There are indeed several different criteria that segmentation can be based on, each producing possibly very different results. Next, in object/region-based coding, the contour information describing the shape and locations of the objects must be transmitted. The gain in improving the motion-compensated prediction must outweigh this additional information inherent in an object-based scheme. Applying lossless intraframe techniques to represent the contours at each frame consumes too many bits. An alternative would be to employ lossy compression such as spline approximation [5] or polygonal representation [6]. To sidetrack this problem, Yokoyama et al. propose a region-based scheme in which the contour information need not be encoded at all [7]. However, the analysis must be performed using only the past reconstructed frames, which are expected to be of relatively poor quality at very low bit rates. Methods to incorporate motion compensation into the contour coding have been introduced recently [6], [8], [9], in which the temporal region correspondence (i.e., temporal linking of objects) problem must be addressed. In [8], however, the segmentation process does not account for object motion, resulting in oversegmentation. Furthermore, the motion estimation is done as an afterthought to fit the segmentation that has already been found. This results in a rather large overhead for the contour coding, even

Manuscript received August 28, 1996; revised February 21, 1997. This work was supported in part by the National Science Foundation under Grant MIP-9528312. An early version of this paper was presented at EUSIPCO’96, Trieste, Italy.

The authors are with the Center for Image Processing Research and the Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590 USA.

Publisher Item Identifier S 0733–8716(98)$10.00 © 1998 IEEE
with motion compensation. This is an inherent problem for segmentation techniques which rely solely on the intensity or texture data [10], [11].

Other issues concerning object-based coding have begun to be properly addressed only recently. This includes the appropriate extraction and processing of uncovered regions due to object movement [6], [12]. Another is the ability to detect and code new objects that appear in a scene [12]–[14]. In addition, some objects undergo complex motion that cannot be properly described by the adopted motion models [3]. Clearly, all of the above-mentioned scenarios call for the coding to be performed in the “intra” mode due to a lack of information in the temporal direction. Few papers have treated this key problem in a systematic manner. Finally, although several approaches have been reported in coding the arbitrarily shaped region intensities [15], [16], few have incorporated their ideas into an overall object-based coding system. Indeed, many of the so-called region/object-based coders proposed so far perform the motion-compensated residual coding on the entire frame [6], [7]. A more effective method would be to code the objects separately as well.

In this paper, an object-based coder addressing all of the above-mentioned issues is presented. Moreover, no a priori assumptions about the contents of the video scene (such as constant background, head-and-shoulders only) need to be made. The extraction of the moving objects is performed by a joint motion estimation and segmentation algorithm based on Markov random field (MRF) models. Thus, our notion of objects is similar to those of [12] and [17]–[20] in that the segmentation is based on coherence in motion. This is in contrast to the approach in [10], [11], [21], where the objects are defined in terms of intensity or texture as pointed out earlier. However, in our approach, the object motion and shape are guided by the spatial color intensity information, thus utilizing the observation that in an image sequence, motion boundaries are a subset of the intensity boundaries [22], [23]. This not only improves the motion estimation/segmentation process itself in extracting meaningful objects true to the scene, but also aids the process of coding the object intensities (especially in the “intra” mode) because a given object has a certain spatial cohesiveness as well.

The MRF formulation also allows us to temporally link the objects, thus creating object volumes in the space–time domain. This helps stabilize the object segmentation process in time, and more importantly, allows the object boundaries to be predicted temporally using the motion information. In fact, the object boundaries are enforced within the MRF model to be “well predictable” by the motion parameters. This leads to an efficient temporal updating scheme to encode the object boundaries, resulting in a significant reduction in bit rate while preserving the accuracy of the boundaries. With the linked objects, uncovered regions can be deduced in a systematic fashion. New objects are detected by utilizing both the motion and intensity information. The interiors of the objects are encoded adaptively, meaning that objects well described by the motion parameters are encoded in the “inter” mode, while those that cannot be predicted in time are encoded in the “intra” mode. This is analogous to \( P \) blocks and \( I \) blocks in the MPEG I/II coding structure, but we now have \( P \) objects and \( I \) objects, which go along with ideas of video objects (VO) and video object planes (VOP) in the MPEG-4 activities [1]. The subband/wavelet approach of [16] is adopted in coding the objects.

The rest of this paper is organized as follows. The MRF-based joint motion estimation and segmentation is presented in Section II. The encoding of the object motion and boundaries is discussed in Section III, as well as the extraction of uncovered regions and detection of new objects. Objects are encoded adaptively based on the motion and texture information, as detailed in Section IV. We present simulation results from the overall coding system in Section V, and make comparisons against existing standard coders. The paper concludes with a summary and closing remarks.

II. JOINT MOTION ESTIMATION AND SEGMENTATION

A. Objectives

In this section, a novel motion estimation and segmentation scheme is presented. An early version was published in [24]. Although the algorithm was specifically designed to meet the coding needs as described in the previous section, the end results could very well be applied to other image sequence processing applications, such as content-based video manipulation, object recognition, and temporal interpolation [25]. The main objective is to segment the video scene into objects that are undergoing distinct motion, along with finding the parameters that describe the motion. In Fig. 1(a), a video scene consists of a ball moving against a stationary background. At each frame, we would like to segment the scene into two objects (the ball and background) and find the motion of each. Furthermore, if the objects are linked in time, we can create three-dimensional (3-D) objects in space–time as shown in Fig. 1(b).

We adopt a Bayesian formulation based on a Markov random field (MRF) model to solve this challenging problem. The MRF approach was initially used in motion segmentation [26] and motion estimation [27] in separate works. Because of the interdependency of the two problems [3], algorithms to perform the motion estimation and segmentation jointly have been proposed [17]–[20]. Our algorithm extends these works by enforcing our moving objects to coincide with spatial intensity boundaries. It is based on a coupled MRF model with constraints such as spatiotemporal smoothness and consistency of the motion vectors and segmentation. The
object segments are encouraged to cluster based on both the motion and intensity information. Furthermore, extraction of the covered/uncovered regions is more robust and meaningful because the segmentation is representative of the real objects that make up the video scene. The MRF formulation also provides a framework for linking the objects in time. A relatively smooth motion field is obtained within each object, making it practical for use in a coding environment. One major drawback to the Bayesian approach is its computational burden. This problem is significantly improved by employing mean-field annealing [28] and performing the computation in a multiresolution pyramid structure.

B. Problem Formulation

Let $I^t$ represent the frame at time $t$ of the discretized image sequence. The motion field $\mathbf{d}^t$ represents the displacement between $I^t$ and $I^{t-1}$ for each pixel. The segmentation field $z^t$ consists of numerical labels at every pixel with each label representing one moving object, i.e., $z^t(x) = n$ ($n = 1, 2, \ldots, N$), for each pixel location $x$ on the lattice $\Lambda$. Here, $N$ refers to the total number of moving objects. Using this notation, the goal of motion estimation/segmentation is to find $\mathbf{d}^t$ and $z^t$ given $I^t$ and $I^{t-1}$. We further assume that $\mathbf{d}^{t-1}$ and $z^{t-1}$ are available, making it possible to impose temporal constraints and to link the object labels.

We adopt the maximum a posteriori (MAP) formulation

$$\{\mathbf{d}^t, z^t\} = \arg \max_{\{\mathbf{d}^t, z^t\}} p(\mathbf{d}^t, z^t | I^t, I^{t-1})$$

which can be rewritten via Bayes rule as

$$\{\mathbf{d}^t, z^t\} = \arg \max_{\{\mathbf{d}^t, z^t\}} p(I^{t-1} | \mathbf{d}^t, z^t, I^t)$$

$$\times p(\mathbf{d}^t | z^t, I^t) p(z^t | I^t).$$

Given the formulation of (2), the rest of the work amounts to specifying the probability densities (or the corresponding energy functions) involved and solving for the solution. Of course, we incorporate various assumptions and models about our motion and segmentation field when formulating the probability densities.

C. Probability Models

The first term on the right-hand side of (2) is the likelihood functional that describes how well the observed images match the motion field data. It reflects the relationship between the gray level changes between frame $t-1$ and $t$ that are corrupted by additive noise. Thus, the actual observed image $\bar{I}^t$ is regarded as a noisy version of the original image $G^t$ or

$$\bar{I}^t(x) = G^t(x) + \eta(x).$$

Ignoring such factors as illumination changes, the change of gray level between the two frames is assumed to be only due to object motion, and we have

$$G^t(x) = G^{t-1}(x - \mathbf{d}^t(x)).$$

Based on the models (3) and (4), and if the noise is assumed to be white, Gaussian with zero mean and variance $\sigma^2$, $p(I^{t-1} | \mathbf{d}^t, z^t, I^t)$ is also Gaussian with

$$p(I^{t-1} | \mathbf{d}^t, z^t, I^t) = Q_t^{-1} \exp \{-U_t(I^{t-1} | \mathbf{d}^t, I^t)\}$$

where

$$U_t(I^{t-1} | \mathbf{d}^t, I^t) = \sum_{x \in \Lambda} \left( I^t(x) - I^{t-1}(x - \mathbf{d}^t(x)) \right)^2 / 2\sigma^2$$

and $Q_t$ is a normalization constant. This is the same likelihood model used by previous authors [27], [28].

The a priori density of the motion $p(\mathbf{d}^t | z^t, I^t)$ enforces prior constraints on the motion field. We adopt a coupled MRF model to govern the interaction between the motion field and the segmentation field both spatially and temporally. The probability density and corresponding energy function are given as

$$p(\mathbf{d}^t | z^t, I^t) = Q_d^{-1} \exp \{-U_d(\mathbf{d}^t | z^t)\}$$

and

$$U_d(\mathbf{d}^t | z^t) = \lambda_1 \sum_x \sum_{y \in N_x} ||\mathbf{d}^t(x) - \mathbf{d}^t(y)||^2 \delta(z^t(x) - z^t(y))$$

$$+ \lambda_2 \sum_x ||\mathbf{d}^t(x) - \mathbf{d}^{t-1}(x - \mathbf{d}^t(x))||^2$$

$$- \lambda_3 \sum_x \delta(z^t(x) - z^{t-1}(x - \mathbf{d}^t(x)))$$

where $\delta(\cdot)$ refers to the usual Kronecker delta function, $|| \cdot ||$ is the Euclidean norm in $\mathbb{R}^2$, and $N_x$ indicates a spatial neighborhood system with respect to $x$. The first two terms of (8) are similar to those in [18], while the third term is added to encourage consistency of the object labels along the motion trajectories. The first term enforces the constraint that the motion vectors be locally smooth on the spatial lattice, but only within the same object label. This allows motion discontinuities at object boundaries without introducing any extra variables such as line fields [29]. The second term accounts for causal temporal continuity, under the model that the motion vector changes slowly frame-to-frame along the motion trajectory. This assumption would be valid only as long as the frame rate of the image sequence is high enough. In very low bit-rate situations, where the frame rate is forced to be relatively low, this assumption was no longer valid, and this constraint had to be minimized by lowering the value of $\lambda_2$. We will elaborate further on this in Section II-F. Finally, the third term encourages the object labels to be consistent along the motion trajectories. This assumption allows for causal temporal continuity, under the model that the object labels be linked in time, and is a generalization of the function used by Stiller [18], where the consistency is enforced only on the object boundaries.

Lastly, the third term on the right-hand side of (2), $p(\mathbf{d}^t | z^t, I^t)$, models our a priori expectations about the nature of the
object label field. In the temporal direction, we have already modeled the object labels to be consistent along the motion trajectories. Our model incorporates the spatial intensity information \( I' \) based on the reasonable assumption that (moving) object discontinuities are a subset of the spatial intensity boundaries. We should point out that intensity boundaries could exist within a uniformly moving or stationary object. The segmentation field is a discrete-valued MRF

\[
P(z' | I') = Q_{z'}^{-1} \exp \{-U_z(z' | I')\} \tag{9}
\]

with the energy function given as

\[
U_z(z' | I') = \sum_{\mathbf{x}} \sum_{\mathbf{y} \in N_{\mathbf{x}}} V_c(z(x), z(y) | I') \tag{10}
\]

where we design the clique potential [29] to be

\[
V_c(z(x), z(y) | I') = \left\{ \begin{array}{ll}
-\gamma, & \text{if } z(x) = z(y), s(x) = s(y) \\
0, & \text{if } z(x) = z(y), s(x) \neq s(y) \\
+\gamma, & \text{if } z(x) \neq z(y), s(x) = s(y) \\
0, & \text{if } z(x) \neq z(y), s(x) \neq s(y).
\end{array} \right.
\tag{11}
\]

Here, \( s \) refers to the spatial segmentation field that is predetermined from \( I \). It is important to note that \( s \) is a deterministic field that can be determined uniquely from \( I \) alone. Thus, our model has the added complexity that the intensity segmentation of \( I' \) must be precalculated. A simple region-growing method (see Appendix) is used. According to (11), if the spatial neighbors \( x \) and \( y \) belong to the same intensity-based object \( s(x) = s(y) \), then the two pixels are encouraged to belong to the same motion-based object. This is achieved by the \( \pm \gamma \) terms. On the other hand, if \( x \) and \( y \) belong to different intensity-based objects \( s(x) \neq s(y) \), we do not encourage \( z \) to be either way, and hence, the 0 terms in (11). This slightly more complex model ensures that the moving object segments we extract have some sort of spatial cohesiveness as well. This is a very important property for our adaptive coding strategy to be presented in Section IV. Furthermore, the above clique function allows the object segmentations to remain stable over time and adhere to what the human observer defines as “objects.”

D. Hierarchical Implementation

Meanwhile, motion estimation, as well as other image processing applications, have been successfully implemented using a hierarchy of image resolutions. One of the major reasons to carry out motion estimation over a hierarchy is to reduce the computation. Another reason is that a multiresolution approach actually helps improve the motion estimation. This makes the computation of relatively large motion feasible. For example, iterative relaxation schemes are very slow to propagate velocity information within image areas with almost homogeneous gray-level distribution [30]. Hierarchical methods alleviate this problem because the size of such areas will be reduced on the top levels of the pyramid. An added feature is that motion estimates and segmentation fields are obtained at different spatial resolutions, leading naturally to the idea of scalable coding.

Hierarchical implementation of the joint motion estimation/segmentation is rather straightforward. Returning to our MAP formulation of (2), the motion estimation and segmentation can be performed on each level of the image pyramid, with estimates at each level being used as the initial estimates for the next level. Specifically, given \( I \), the image pyramid is formed by a series of spatial low-pass filtering and subsampling by a factor of 2 in both the \( x \) and \( y \) directions. This results in the pyramid \( I^k, k = 0, 1, \ldots, K = 1 \), where \( I^0 \) represents the full-resolution image, i.e., the bottom level. We also need a spatial segmentation \( s^k \) of \( I^k \) at every level, and this is obtained by sequentially subsampling the segmentation of the full-resolution image. Although the spatial segmentation could be performed at each resolution, we chose this simpler method to cut down on computation. In Fig. 2, an example of a pyramid decomposition of both the image and its resulting segmentation field at different spatial resolutions is demonstrated. Once the motion estimates are found for level \( k \), they serve as the initial estimates for level \( k - 1 \). Of course, the values have to be doubled because the spatial resolution is doubled when going down one level of a pyramid. The motion segmentation field is interpolated by pixel replication, and these values are used as initial estimates for the next level as well. The hierarchical implementation greatly reduced the computation time of the estimation process. Again, this is more crucial in the very-low-bit-rate situation, where the necessarily low frame rate results in a rather large motion estimate.

E. Solution

Due to the equivalence of MRF’s and Gibbs distributions [29], the MAP solution amounts to a minimization of the sum of potentials given by (6), (8), and (10). In [17], the unknowns \( d \) and \( z \) are grouped as one vector and found jointly. This makes the dimensions of the solution overwhelming, forcing the use of a suboptimal iterated conditional modes (ICM) algorithm [31].

Instead, we adopt a two-step iterative procedure [19] in which the motion and segmentation fields are found in an alternating fashion assuming the other is given. This not only reduces the computational burden, but also allows different relaxation algorithms to be implemented for finding \( d \) and \( z \). We must keep in mind that even though both are modeled as Markov random fields, the motion field and the segmentation field are fundamentally different entities. The key difference lies in the fact that the label values of the segmentation field are merely tokens to separate the different moving objects, and do not possess any physical meaning. We employ mean field annealing (MFA), but only for the motion field for the reason given above. We resort to ICM for finding the object label field \( z \). MFA has been shown to perform nearly as well as the optimal simulated annealing, but much more quickly. Furthermore, it is less likely to be trapped in local minima than ICM, while converging just as quickly [28]. Our experiments also confirmed that MFA outperformed ICM in finding the motion field, especially in cases where good initial estimates are not available. This includes the motion between the first two frames, and frames in which there is a sudden change.
of direction or speed in object motion. The reader is referred to [28] for details on the implementation of MFA. ICM was performed via the Gibbs sampler [29] with the temperature held at $T(0) = 1$ for all iterations.

F. Experimental Results

In this subsection, we present and discuss experimental results from our joint motion estimation and segmentation algorithm. Simulations for the motion analysis and subsequent video coding were done on two test sequences, Miss America and Carphone, both suitable for low bit-rate applications. The analysis was performed on every fourth frame, corresponding to a frame rate of 7.5 frames/s. Before the actual results are analyzed, some additional implementation issues must be addressed.

The energy functions as defined in (6), (8), and (11) contain several control parameters, namely, $\sigma$, $\lambda_1$, $\lambda_2$, $\lambda_3$, and $\gamma$. These parameters can be regarded as weights that can be designed to emphasize or deemphasize the various constraints. For example, increasing $\gamma$ would make the object segments adhere more to the intensity-based segmentation. On the other hand, if our primary goal is to minimize the displaced frame difference, $\sigma$ would be assigned a relatively low value. As of now, we set the parameters in an ad hoc manner by experiment. The values $\sigma = 5.0$, $\lambda_1 = 0.5$, $\lambda_2 = 0.1$, $\lambda_3 = 1.0$, and $\gamma = 3.0$ were used for obtaining the results presented here and used in the subsequent coding. Changing any of the values slightly had a minimal effect on the results. As pointed out earlier in this section, $\lambda_2$ had to be assigned a relatively small value due to the low frame rate. Alternatively, we also tried the analysis at the full rate of 30 frames/s. In this implementation, $\lambda_2 = 1.0$ gave good results because the temporal smoothness of the motion is justified. Of course, the rate needed to transmit the motion information becomes threefold with such a scheme, and we are presently investigating methods to alleviate this. Such an example where the segmentation is done at the full frame rate while the coding is performed at a lower rate can be found in [8].

The initial frame of a sequence is segmented ($\mathbf{z}^0$) with only the intensity information since no motion information is available. The same region-growing algorithm mentioned earlier (see Appendix) is also used here. The motion and segmentation fields for the second frame ($\mathbf{d}^1$ and $\mathbf{z}^1$) are then found using $\mathbf{z}^0$ and assuming $\mathbf{d}^0 = \mathbf{0}$. The temporal motion smoothness constraint is turned off by setting $\lambda_2 = 0$. Contrary to other works [6], [14], the initial spatial segmentation is not as crucial because any objects that were not captured at $t = 0$ will be recovered at $t = 1$ (according to the detection of new objects in Section III-B), while objects that were oversegmented are merged by the motion. In fact, the overall segmentation could be implemented without segmenting the initial frame, i.e., setting $\mathbf{z}^0(\mathbf{x}) = \mathbf{0}$ for all $\mathbf{x}$.

A three-level pyramid was used for the multiresolution algorithm, using the two-step iterations as described earlier. A maximum of ten iterations at each pyramid level was sufficient, making the analysis bearable in computational time. Typically, for Miss America, seven to 15 objects were extracted, while Carphone was segmented into 14 to 21 objects. In Figs. 3
Fig. 3. Motion estimation and segmentation for Miss America. (a) Miss America frame no. 76, (b) motion segmentation motion field by (c) proposed method and (d) hierarchical block matching.

and 4, the motion estimation and segmentation results for both sequences are illustrated. The motion field is compared to that obtained by hierarchical block matching. The block size used was $16 \times 16$. We can see that the MRF model produced smooth vectors within the objects with definitive discontinuities at the boundaries. Also, it can be observed that the object boundaries more or less define the “real” objects in the scene. In Section V, it will be shown that the rate needed to transmit such a dense motion field and its segmentation (with appropriate coding) is lower than the rate needed to transmit the motion vectors resulting from block matching. This makes our estimation results suitable for video compression.

In Fig. 5, the object segments at three different time frames for Miss America are displayed. The same gray-level shading represents the consistent object labels that are linked in time. We can see that the moving objects are properly tracked in the temporal direction. Furthermore, Fig. 5 shows a good example of new objects being extracted. Initially, the hair of the lady is considered a part of the stationary dark background. This is due to its similarity with the background in both intensity and motion. When the hair starts moving, it is labeled as a separate moving object. The birth of new objects is discussed in Section III-B. Finally, it is fused into the background when its motion becomes limited again. One could argue from a semantic point of view that perhaps the hair should be labeled as a distinct object through the entire time duration. However, our notion of objects is based on coherence in motion in an effort to maximize the coding gain, and therefore such interpretations are beyond the scope of this paper.

III. REPRESENTING THE OBJECT MOTION

The motion analysis from the previous section provides us with the boundaries of the moving objects and a dense motion field within each object. In this section, we are interested in efficiently representing and coding the found object information. An affine parametric representation is found for each object’s motion by fitting the dense but smooth motion field. A new weighted least squares fitting is performed so that only the data with higher confidence are used in finding the parameters. Potential new objects are found for regions where the fit fails. By modeling the motion of the temporally linked objects with the affine parameters, the bit rate to encode the object boundaries is significantly reduced with a novel temporal updating scheme. Furthermore, uncovered regions can be extracted simply by comparing the object location and motion parameters between two frames.
Fig. 4. Motion estimation and segmentation for Carphone. (a) Carphone frame no. 88, (b) motion segmentation motion field by (c) proposed method and (d) hierarchical block matching.

Fig. 5. Object segments for Miss America. (a) Miss America frames 32, 76, and 112 and (b) corresponding object segments.
A. Parametric Motion of Objects

A two-dimensional (2-D) planar surface undergoing rigid 3-D motion yields the following affine velocity field under orthographic projection [4]:

\[
\begin{align*}
    v_x &= a_1 + a_2 x + a_3 y \\
    v_y &= a_4 + a_5 x + a_6 y
\end{align*}
\]  

(12)

where \( v = (v_x(x, y), v_y(x, y)) \) refers to the apparent velocity at pixel \((x, y)\) in the \(x\) and \(y\) directions, respectively. Thus, the motion of each object can be represented by the six parameters of (12). In [32], Wang and Adelson employ a simple least squares fitting technique, and point out that this process is merely a plane fitting algorithm in the velocity space. Specifically, denote the six parameter set of (12) by

\[
p = \begin{bmatrix}
    a_1 & a_2 & a_3 \\
    a_4 & a_5 & a_6
\end{bmatrix}.
\]  

(13)

For a particular object of size \( M \), we can order the \( M \) pixels that belong to the object as \((x_1, y_1), \ldots, (x_M, y_M)\). With this notation, the least squares estimate of \( p \) is given by [32]

\[
p = \left[ \sum_{i=1}^{M} x_i \sum_{i=1}^{M} y_i \sum_{i=1}^{M} v_x(x_i, y_i) \sum_{i=1}^{M} v_y(x_i, y_i) \right]^{-1} \left[ \sum_{i=1}^{M} y_i \sum_{i=1}^{M} x_i \sum_{i=1}^{M} x_i v_x(x_i, y_i) \sum_{i=1}^{M} x_i v_y(x_i, y_i) \right] \times \left[ \sum_{i=1}^{M} y_i \sum_{i=1}^{M} x_i \sum_{i=1}^{M} y_i v_x(x_i, y_i) \sum_{i=1}^{M} y_i v_y(x_i, y_i) \right].
\]  

(14)

We improve on this method by introducing a weighting matrix \( W \) so that data with higher confidence can be given more weight in the fitting process. We can do this by defining a weighting matrix \( W \)

\[
W = \text{diag}\left[w_1, w_2, \ldots, w_M\right]
\]

where \( w_i (i = 1, 2, \ldots, M) \) corresponds to the weight given for pixel \( i \). Then, the weighted least squares solution becomes

\[
p = \left[ \sum_{i=1}^{M} w_i^2 \sum_{i=1}^{M} w_i^2 x_i \sum_{i=1}^{M} w_i^2 y_i \right]^{-1} \left[ \sum_{i=1}^{M} w_i^2 x_i \sum_{i=1}^{M} w_i^2 y_i \sum_{i=1}^{M} w_i^2 x_i y_i \right] \times \left[ \sum_{i=1}^{M} w_i^2 y_i \sum_{i=1}^{M} w_i^2 x_i \sum_{i=1}^{M} w_i^2 x_i y_i \right] \times \left[ \sum_{i=1}^{M} w_i^2 y_i \sum_{i=1}^{M} w_i^2 y_i x_i \sum_{i=1}^{M} w_i^2 y_i x_i y_i \right].
\]  

(15)

We design the weighting matrix \( W \) based on experimental observations on our motion field data. For one thing, we would like to eliminate (or lessen) the contribution of inaccurate data on the least squares fitting. This can be done by measuring the displaced frame difference (DFD) at each point \( x_i = (x_i, y_i) \). The DFD is defined by

\[
\text{DFD}(x_i) = I^i(x_i) - I^{i-1}(x_i - d(x_i)).
\]  

(16)

Obviously, we want pixels with higher DFD to have a lower weight, and thus define

\[
\alpha_i = 1 - \text{DFD}(x_i)/\text{MAX.DFD}
\]  

(17)

where

\[
\text{MAX.DFD} = \max_{x_i} \text{DFD}(x_i).
\]

Next, we incorporate the fact that in image regions with almost homogeneous gray-level distribution, the mean field solution favored motion vectors of value 0. Thus, for relatively large moving objects with little texture, the regular least squares fitting of (14) failed to capture the proper motion. This problem was solved by measuring the busyness in the search region of each pixel. This was represented by defining a binary weight function

\[
\beta_i = \begin{cases} 
0, & \text{not busy} \\
1, & \text{busy}
\end{cases}
\]

The busyness is decided by comparing the gray-level variance in the search region of pixel \( i \) against a predetermined threshold. The final weight for pixel \( i \) is determined as

\[
w_i = \lambda \alpha_i + (1 - \lambda) \beta_i \quad (0 < \lambda < 1)
\]  

(18)

to ensure that \( w_i \in [0, 1] \). In conclusion, motion vectors that give small DFD and those that were found in areas with textural information are given the most weight in the least squares fitting process. Fig. 6 illustrates the weights for a particular object in Miss America.

B. Appearance of New Regions

To extract meaningful new objects, additional processing was necessary based on the least squares fitting. The basic idea is taken from the "top-down" approach of Musmann et al. [3], in which regions where the motion parametrization fail are assigned as new objects. However, we incorporate additional criteria to extract more meaningful objects. Initially, each object found from our MRF-based estimation is split up into subregions using an intensity-based segmentation. Again,
the same region-growing algorithm from the Appendix is used. Then, a subregion is labeled as a new object only if all three of the following conditions are met.

1) The norm difference between the synthesized motion vectors and the original dense motion vectors is large.
2) The prediction error resulting from the split is significantly reduced.
3) The prediction error within the subregion is high without a split.

Because of the smoothness constraint, splitting within objects merely based on the affine fit failure did not produce meaningful objects. A norm difference of one pixel was used in testing condition 1). The second condition ensures that the splitting process decreases the overall coding rate. Note that only a slight decrease in the prediction error is not good enough because it will not make up for the increased bits needed to code the contour of the new region. We determined heuristically the threshold of this reduction rate to be 15%. Finally, the third condition guards against splitting the object when there is no need to in terms of coding gain. We used \( \mu = 20 \) [6] as the prediction error threshold. In summary, candidates for new objects are first found by color segmentation, and only those with distinct parametrizable motion are assigned new object labels.

Our method of detecting new objects is unique from recent works which address this issue. In [6] and [14], new objects can only be found in regions uncovered by object motion. The technique in [13] requires the motion estimation to be recalculated, and does not incorporate condition 2). Finally, contrary to [17], [20], the labeling of new objects is taken out of the MRF formulation in our approach. Our experiments showed that the MRF model produced new objects which are small and spurious, not adhering to the true objects in the scene. Consequently, they could not be properly tracked in subsequent frames, making them unwarranted for coding purposes.

C. The Problem of Covered/Uncovered Regions

Using our object segmentation and motion information, the uncovered regions can be extracted rather easily. Because our objects are linked in time, the extraction merely involves projecting our synthesized motion vectors in time and comparing labels. More specifically, to find the uncovered regions in frame \( f \), each pixel is projected back to frame \( f-1 \) according to its synthesized motion vector. The uncovered pixels are simply those whose object labels do not match along the trajectory. Fig. 7 illustrates some examples.

D. Coding the Object Boundaries

We have already seen that temporally linked objects in an object-based coding environment offer various advantages. However, the biggest advantage comes in reducing the contour information rate. Using the object boundaries from the previous frame and the affine transformation parameters, the boundaries can be predicted with a good deal of accuracy. In fact, the boundaries are made to behave in this manner by our MRF model. This is in sharp contrast to the motion-compensated partition coding techniques of [8] and [9], in which the temporally linked object segmentation is found first, and then the motion parameters are found to describe the predetermined boundaries. This creates the problem that separate motion parameters are needed for the texture and partition prediction. The motion parameters from our MRF model attempt to maximize the prediction of both.

The actual partition coding is similar to [9], with some modifications to better suit our motion-based boundaries. We adopt the backward prediction approach [9]. Let us assume that \( N \) objects exist in frame \( f-1 \) whose boundaries are defined by \( z^{f-1} \). Furthermore, the affine parameters \( p_i \), \( i = 1, 2, \ldots, N \) are assumed known. Based on \( z^{f-1} \) and the \( p_i \)'s, we build a prediction partition of \( z^f \) as follows. At each pixel, try the motion vectors synthesized from each of the \( p_i \)'s. If the pixel is projected back to label \( i \) using \( p_i \), and this match is unique, then that pixel is given the label \( i \). Inevitably, pixels with more than one match (overlapping regions) and those without any (empty regions) arise. In [9], an extra ordering information is transmitted to resolve these conflicts. However, this extra

---

**Fig. 6.** Least squares weights for Miss America’s right shoulder: pixels with lighter gray-level values are given more weight.

**Fig. 7.** Extraction of uncovered regions: (a) one moving object and (b) two moving objects.
complexity is unnecessary for our motion-based partitions because the overlapping and empty pixels are very small in number and occur only on the boundaries. Thus, it suffices to group such pixels into connected regions and transmit 1-bit flags (assuming the ambiguity was between two objects) to denote which object the regions belong to. The prediction error contours were coded by classical chain coding, as were the new object contours. In our approach, the contours of new objects were encoded after the predicted contours were built. In [9], the contours of new objects are coded before the contours in prediction mode. This is inefficient because, in many cases, a portion of the new object contour shares a boundary with continuing objects.

Using this temporal updating scheme, the object segmentation for Miss America was coded at 0.94 kbits/s. This figure includes the chain coding that was implemented on new objects. This corresponds to a rate of 0.28 bits/contour point (bpcp), which compares favorably to the 0.3 bpcp reported in [5], where a lossy temporal updating scheme is implemented using spline and polygon approximations.

E. Object Motion/Segmentation Coding

The overall encoding and decoding scheme for the object motion and contour information is illustrated in Fig. 8. The input to the coder is $d$ and $z$, the output from our joint motion estimation/segmentation algorithm.

IV. ADAPTIVE CODING OF OBJECTS

The coding of the objects’ interior is performed by adaptive coding. In short, objects that can be described well by the motion are encoded by motion-compensated predictive (MCP) coding, and those that cannot are encoded in “intra” mode. The coding is done independently on each object using spatial subband/wavelet coding. Since the objects are arbitrarily shaped, the efficient signal extension method proposed by Barnard [16] is applied.

Although the motion compensation was relatively good for most objects at most frames, the flexibility to switch to intramode (I mode) in certain cases is inevitable. For instance, when a new object appears from outside the scene, it cannot
be properly predicted from the previous frame. Thus, these new objects must be coded in the \( I \) mode. This includes the initial frame of the image sequence, where all of the objects are considered new. However, not all new objects were set to the \( I \) mode since some of them could be predicted in time. This corresponds to objects that “stay still” in the scene for awhile, and then start to move at an intermediate time. Also, even for “continuing” objects, the motion might be too complex at certain frames for our model to describe properly, resulting in poor prediction. This is another case when objects are encoded in the \( P \) mode. Such classification of objects into \( I \) objects and \( P \) objects is analogous to blocks and blocks in the current video standards such as MPEG and H.263 [33].

Uncovered regions must also be coded in the \( I \) mode since they cannot be predicted from the previous frame in general. However, in many instances, such as when an object moves against a uniform background, the uncovered region can be extrapolated from its spatial neighboring region. Thus, an uncovered region that was similar in intensity to the rest of the object it belonged to was simply merged with the object. Those that did not belong to any neighboring region were given new object labels and coded in the \( P \) mode.

In summary, at each frame, every object was classified into one of three categories: \( I \) object (case \( I \)), motion-only \( P \) object (case \( P-1 \)), and motion+residual \( P \) object (case \( P-2 \)). The decision between \( I \) mode or \( P \) mode is made by comparing the original intensity variance (\( \text{VAR}_I \)) and the motion-compensated error (\( \text{VAR}_P \)) within the region. Thus, an object is classified as \( I \) mode if \( \text{VAR}_I < \alpha \text{VAR}_P \). The value \( \alpha = 0.5 \) was used in our experiments because the number of \( I \) objects needed to be kept at a minimum under our restricted bit budget. Further distinction between case \( P-1 \) and \( P-2 \) was made by comparing \( \text{VAR}_P \) against a threshold \( \mu \). Again, \( \mu = 20 \) was used for this purpose.

The rate allocation was done to simulate constant bit-rate (CBR) transmission at the frame level. Except for the first \( I \) frame, the bits were thus allocated equally among the frames. For example, coding at 16 kbits/s with 7.5 frames/s, each frame has roughly 2000 bits to use. The motion parameters are encoded by fixed-length coding, while the contours are coded as described in Section III-D. The remaining bits are used for the spatial subband/wavelet coding. We employ a uniform threshold quantizer followed by arithmetic coding [16] to achieve the target bit rate. The residual bits were distributed among the \( P-2 \) objects based on size and \( \text{VAR}_P \). As of now, the segmentation and coding are performed independently of one another. A more mature segmentation-based rate control [34] should improve our overall results (quantitatively, at least).

The overall object-based coder block diagram is shown in Fig. 9.

V. CODING RESULTS

The proposed object-based coding scheme was applied to two test sequences, Miss America and Carphone. To simulate low bit rates, QCIF versions (spatial size 176 x 144 pixels)
were used, and coding was done at a frame rate of 7.5 frames/s. The low frame rate meant that the motion between two frames was relatively large, thus making the motion analysis and coding quite challenging.

The first frame was coded by region-based subband/wavelet coding [16] with the segmentation found as described in Section II-F. Thus, for the first frame, all of the objects were considered to be in the I mode. The subsequent frames were coded according to Fig. 9. The joint motion estimation/segmentation and the resulting parametric fitting were performed on the original frames. Because the objects were temporally linked in time, good initial estimates were found at each frame, thus leading to faster convergence. Our coding results were compared to the Telenor research group’s H.263 implementation.\(^1\) H.263 [33] is a block-based hybrid coding system that improves on the H.261 codec by featuring overlapped block matching, among others. These features enable the H.263 to perform optimally at very low bit rates.

\(^1\)http://www.nta.no/brukere/DVC/.
TABLE I
DECODED AVERAGE PSNR FOR Miss America, FRAMES 0–149

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>component</th>
<th>H.263</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 kbps</td>
<td>Y</td>
<td>35.87</td>
<td>35.98</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>37.42</td>
<td>37.52</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>35.37</td>
<td>35.48</td>
</tr>
<tr>
<td>12 kbps</td>
<td>Y</td>
<td>37.20</td>
<td>37.09</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>37.77</td>
<td>37.88</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>36.47</td>
<td>36.33</td>
</tr>
<tr>
<td>16 kbps</td>
<td>Y</td>
<td>38.23</td>
<td>38.01</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>38.43</td>
<td>38.23</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>37.59</td>
<td>37.20</td>
</tr>
</tbody>
</table>

For Miss America, simulations were performed at three bit rates: 8, 12, and 16 kbits/s. The average reconstructed PSNR’s are summarized in Table I. Roughly 15% of the total bits were spent on the object motion and boundary coding. A plot of the PSNR and rate allocation is shown in Fig. 10. We can see that the CBR constraint is met fairly well for our object-based coder, while the H.263 coder assigned more bits for active frames. This explains why the object-based coder gave poorer numerical results at these frames. A variable bit-rate (VBR) implementation of our coder could improve the results in this respect. We can also observe from Fig. 10 that the motion rate (this includes both motion parameter and contour coding) is smaller for our object-based coder.

We can see that in terms of PSNR, the proposed object-based coder was comparable in performance to the conventional technique at very low bit rates. However, more importantly, the object-based coder produced visually more pleasing video. The annoying blocking artifacts that dominate the block-based methods at low bit rates are not present. Also, the object-based coder gave much clearer reconstructed frames with less blurriness. In Fig. 11, decoded frames from both methods are displayed. Here, for this untypical frame, even though the reconstruction PSNR favors the H.263 coder by over a full decibel, the object-based result is visually more pleasing.

Simulations were also performed on the more complex test sequence Carphone at 24 and 32 kbits/s. The average reconstructed PSNR values are summarized in Table II. Here, roughly 25% of the total rate was devoted to the object motion and boundary encoding. A typical reconstructed frame is displayed in Fig. 12. Again, the object-based coder resulted in a slightly worse decoded PSNR for this particular frame, but the picture is devoid of block artifacts and is less blurry.
TABLE II

<table>
<thead>
<tr>
<th>Bit-rate</th>
<th>component</th>
<th>II.263</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 kbps</td>
<td>Y</td>
<td>30.08</td>
<td>30.01</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>35.83</td>
<td>35.60</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>36.38</td>
<td>36.42</td>
</tr>
<tr>
<td>32 kbps</td>
<td>Y</td>
<td>31.16</td>
<td>30.79</td>
</tr>
<tr>
<td></td>
<td>U</td>
<td>36.54</td>
<td>36.15</td>
</tr>
<tr>
<td></td>
<td>V</td>
<td>37.16</td>
<td>36.89</td>
</tr>
</tbody>
</table>

APPENDIX

A spatial intensity segmentation is needed in several phases of the proposed coder. One is the segmentation of the first and subsequent (if any) intraframes, in which no motion is available. Also, the deterministic field $s$ needs to be calculated in Section II-C. Finally, intensity segmentation is needed in finding candidate new objects as described in Section III-B. We implement a modified version of the centroid linkage formation. These regions are merged into the closest (in YUV Euclidean distance) neighboring region.

VI. CONCLUSION

We have presented an object-based video coding system with improved coding performance from a visual perspective. An improved motion estimation/segmentation algorithm enables the extraction of moving objects that correspond to the true scene. By following the objects in time, the object motion and contour can be encoded efficiently with temporal updating. The interiors of the objects are encoded by predictive 2-D subband/wavelet analysis. The objects are encoded adaptively based on scene content. No a priori assumptions about the image or motion are needed. We can conclude from our results that object-based coding is indeed a viable alternative to the block-based standards for very low bit-rate applications.

ACKNOWLEDGMENT

The authors would like to thank Dr. L. Böröczky, now with IBM Corporation, for her contributions to the early stages of this work.

REFERENCES

Soo-Chul Han was born in Seoul, Korea, in 1967. He received the B.S. degree in electronic engineering from Yonsei University, Seoul, in 1990, the M.S. degree in electrical engineering from Rensselaer Polytechnic Institute (RPI), Troy, NY, in 1992, and the M.S. degree in mathematics from RPI in 1996. He is pursuing the Ph.D. degree in computer and systems engineering at RPI. Presently, he is a Research Assistant in the Center for Image Processing Research at RPI. He has worked as a Research or Teaching Assistant for his entire stay at RPI. From 1993 to 1994, he served in the Korean Army as a Communications Specialist. In the summer of 1995, he was with Texas Instruments, Dallas, TX, where he worked on MPEG-4 video coding. His research interests include image processing, information theory, and computer vision.


Since 1976, he has been with the ECSE Department, Rensselaer Polytechnic Institute, Troy, NY, where he is currently a Professor and Associate Director of the Center for Image Processing Research. He has taught courses in digital signal processing, probability, random processes, digital image processing, and estimation theory. His research interests include image and video estimation, restoration, filtering, and coding. He has authored or coauthored over 80 papers in these fields. He has coauthored one text in the area of probability and random processes (2nd ed., 1994), was Editor of the 1990 book, Subband Image Coding, and has just coedited a new book, with Ming Hang, entitled, Handbook of Visual Communications. During the academic year 1985–1986, he was a Visiting Professor in the Information Theory Group, Delft University of Technology, The Netherlands. He directed the Circuits and Signal Processing Program at the National Science Foundation, Washington, DC, in 1987–1988.

Dr. Woods was corecipient of 1976 and 1987 Senior Paper Awards of the now IEEE Signal Processing (SP) Society. He is a former Associate Editor of the TRANSACTIONS ON SIGNAL PROCESSING. He is a former Chairman of the Society Technical Committee on Multidimensional Signal Processing. He is a former member of the Administrative Committee of the now SP Society. He was founding Chairman of the Signal Processing Society Education Committee. He served on the Editorial Board of Graphical Models and Image Processing, and was Chairman of the Seventh Workshop on Multidimensional Signal Processing in 1991. He received the Signal Processing Society Meritorious Service Award in 1990. He received an SP Society Technical Achievement Award in 1994. He is currently on the Editorial Board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is a member of Sigma Xi, Tau Beta Pi, Eta Kappa Nu, and the AAAS. He was Cochairman of the 1st Technical Program Committee of the 1st IEEE International Conference on Image Processing, Austin, TX, November 1994.