97-24

PROCEEDINGS OF SPIE

SPIE—The International Society for Optical Engineering

Visual Communications and Image Processing '98

Sarah A. Rajala
Majid Rabbani
Chairs/Editors

28–30 January 1998
San Jose, California

Sponsored by
SPIE—The International Society for Optical Engineering
IS&T—The Society for Imaging Science and Technology

Volume 3309
Part One of Two Parts

Accurate segmentation and estimation of parametric motion fields for object-based video coding using mean field theory

Radhakrishnan Haridasan and John S. Baras

Institute for Systems Research &
Department of Electrical Engineering
University of Maryland
College Park, MD 20742 USA

ABSTRACT

We formulate the problem of decomposing a scene into its constituent objects as one of partitioning the current frame into objects comprising it. The motion parameter is modeled as a nonrandom but unknown quantity and the problem is posed as one of Maximum Likelihood (ML) estimation. The MRF potentials which characterize the underlying segmentation field are defined in a way that the spatio-temporal segmentation is constrained by the static image segmentation of the current frame. To compute the motion parameter vector and the segmentation simultaneously we use the Expectation Maximization (EM) algorithm. The E-step of the EM algorithm, which computes the conditional expectation of the segmentation field, now reflects interdependencies more accurately because of neighborhood interactions. We take recourse to Mean Field theory to compute the expected value of the conditional MRF. Robust M-estimation methods are used in the M-step. To allow for motions of large magnitudes image frames are represented at various scales and the EM procedure is embedded in a hierarchical coarse-to-fine framework. Our formulation results in a highly parallel algorithm that computes robust and accurate segmentations as well as motion vectors for use in low bit rate video coding.

Keywords: Motion-based segmentation, Markov Random Field, Mean Field theory, Expectation-Maximization, object-based coding

1. INTRODUCTION

Very low bit rate coding of video with a provision for content access is the focus of current research in source coding of video. Second generation coding techniques[1] were proposed as an alternative to block-based approaches to alleviate the problems occuring at low bit rates such as blocking artifacts, jerky motion etc. One such technique, namely, the object-based approach deals with extracting moving objects, representing them in terms of parameters which characterize their shape, motion and texture, and coding these parameters efficiently such that the original video frames can be synthesized at the decoder. Object-based representation also facilitates multimedia functionalities such as content based access.

The fundamental problem that needs to be solved in object-based coding is that of decomposing a scene into its constituent objects. The most important cue to segmenting a scene comes from visual motion. In order to code generic scenes, for which no explicit object model is available, implicit models that mix structure and motion information implicitly have been pointed out to be suitable.[2] Such models use the fact that 3-D real world objects are projected onto the image plane by a camera to connect successive images in the camera plane pertaining to each object and obtain a description. Image motion can be used to identify areas corresponding to differently moving objects in space and to distinguish these from the background. The objective of motion-based image segmentation is to partition the current image into regions characterized by coherent motion. Since these motion regions correspond to different moving objects in the scene, this problem is of fundamental importance in the context of object-based coding of video. As opposed to the problem of spatial (or static) image segmentation, whose goal is to determine

Other author information:(Send correspondence to R. H.)
R. H.: Email: krishnan@isr.umd.edu; Telephone: 301-405-6578; Fax: 301-314-9218; Supported by the National Science Foundation under Grant No. NSF EEC 94-02384 and the National Aeronautical & Space Administration through the NASA Cooperative Agreement NCC3-528.
J. B.: Email: baras@isr.umd.edu; Telephone: 301-405-6606; Fax: 301-314-9218

regions of uniform luminance, our problem is one of *spatio-temporal segmentation.* Computing the spatio-temporal segmentation, like its static counterpart, is a "chicken and egg" problem. The two parts of the problem, motion estimation and motion segmentation are inextricably coupled. On the one hand, an accurate estimate of the motion is required to obtain a good segmentation. On the other hand, a good segmentation is needed to precisely estimate the motion.

To overcome the above dilemma, various approaches have been adopted in the literature. A 3D segmentation based on constant luminance values does away with the correspondence problem along the time axis.[3] However, instead of motion, uniform luminance is used as the criterion for segmentation. Wang and Adelson[4] base their segmentation on the computation of optical flow. Optical flow computation itself is an ill-posed problem because of the *aperture problem**. Regularization techniques can constrain the optical flow vector, but the smoothness assumptions can distort the motion boundaries. A properly chosen discontinuity adaptive regularizer[5] can be used to preserve motion boundaries. However, in the context of coding the computation of optical flow becomes an intermediate step whose accuracy plays a crucial role in motion-based segmentation. The high cost of computation that would be incurred for obtaining an intermediate result suggests a more direct formalism than the two step one. Our approach is one such and is based on this observation.

Recently, techniques that simultaneously estimate segmentation and motion have been proposed. Such approaches are usually embedded in Markov Random Fields (MRF). MRF modeling provides a convenient framework to regularize the estimation problem. It is also ideally suited to introduce local and contextual constraints. Further, some amount of parallelism can be expected in the resulting algorithms. Usually, a Bayesian approach is adopted and the problem is posed as a MAP estimation of motion and segmentation given the observations.[6–8] The equivalence between MRFs and Gibbs distributions leads to global minimization of an appropriate energy function which can be performed stochastically or deterministically. While Konrad and Dubois[7] and Stiller[6] estimate dense motion fields, Bouthemy and Francois[8] use parametric modeling.

A reliable estimate of motion not only segments objects precisely but also reduces the prediction error information that needs to be sent to the decoder. Since motion information also needs to be transmitted to the decoder a succinct representation is desirable. From a coding point of view it is advantageous to have a parametric modeling of the motion field. Unlike most approaches which compute dense optical flow fields and then try to reduce the motion vector information by quantization for purposes of coding, we use motion to group pixels into objects. Pixels that move similarly can be represented by a few parameters and are considered to be arising from the same motion model. A segmentation derived from such a parametric modeling of the motion field has the advantage that it can tackle cases where there is fragmented occlusion by integrating information from pixels that are not necessarily neighbors but belong to be the same motion model. An example of such a situation would be the branches of a tree occluding a uniform background. Motion-based segmentation seeks to decompose an image sequence into constituent objects (or segments) in terms of their motion. In order to describe a scene in terms of its constituent objects we try to obtain a description of the motion field in terms of parametric motion models. To provide for occlusions, such as the one mentioned above, we model the scene by layers or regions of support. Thus, the segmentation of a scene in terms of its constituent objects should result in a set of support maps, each corresponding to the region described using the parametric model. The task of motion-based segmentation is to find these support regions automatically. Our approach is akin to the parametric mixture modeling of motion that was introduced by Darrell and Pentland.[9] Based on the observation in[2] that implicit 2-D parametric models have the best potential for use in motion-based video coding Sawhney and Ayer[10] used it to represent video in terms of layers. They used techniques from robust estimation to make their segmentation insensitive to outliers. However, Sawhney and Ayer regarded the observations to be independent and the underlying segmentation field to be i.i.d. We use a more accurate model that captures the physical constraints of the segmentation field by defining it in terms of a Markov Random Field.

2. PARAMETRIC MOTION MODELING AND SEGMENTATION

Mathematically, we formulate the problem as follows. The observation consists of two image frames, $I(t-1)$ and $I(t)$, captured by the camera at successive instances of time. $I(t)$ is taken as the current frame which needs to be partitioned into objects comprising the frame based on the motion that took place between $t-1$ and t. The information pertaining to the motion of objects is contained in a parameter vector $\mathbf{\Phi}(t)$. $\mathbf{\Phi}(t)$ relates $I(t)$ to $I(t-1)$

*Only the component of the flow vector along the intensity gradient can be recovered.

by using K motion models, each of which is a 6 or 8 parameter motion model. More particularly, each motion model is characterized by the parametric velocity vector[†], $\mathbf{u}_{\mathbf{a}^k}(t), k = 1, 2, \ldots, K$. $I(t-1)$ and $I(t)$ are specified on a collection of N sites (or pixels) where each site is indexed by a single number i, $i = 1, 2, ...N$. The set of sites is denoted by $\mathcal{S}$ and the Cartesian coordinate location of the ith pixel is denoted by $\mathbf{x}_i = (x_i, y_i)$. In order to compute motions of varying magnitudes the images are represented at multiple scales. It is assumed that I refers to any of these filtered representations of the original image. In the following discussion superscript k refers to the kth model and subscript i refers to the ith pixel location.

To model the intensity $I(\mathbf{x}_i, t)$ at pixel $\mathbf{x}_i$ and time t in terms of the parameter vector $\mathbf{\Phi}(t)$ we introduce K model prediction images, $\{\tilde{I}^k(t)\}_{k=1}^K$, each of which can be predicted from $I(t-1)$ using the relation

$$\tilde{I}^k(\mathbf{x}_i, t) = I(\mathbf{x}_i - \mathbf{u}_{\mathbf{a}^k}(\mathbf{x}_i, t), t-1), \qquad k = 1, 2, \ldots, K. \qquad \forall i \in \mathcal{S} \tag{5}$$

To assign each pixel to a unique model, we introduce a K dimensional support vector $\mathbf{s}_i(t)$ where

$$\mathbf{s}_i(t) = [s^1(\mathbf{x}_i, t) \quad s^2(\mathbf{x}_i, t) \quad \ldots \quad s^K(\mathbf{x}_i, t)]^T \tag{6}$$

where the **binary** variables $s^k(\mathbf{x}_i, t)$ $(= s_i^k(t))$ are defined as

$$s^k(\mathbf{x}_i, t) = \begin{cases} 1 & : \quad I(\mathbf{x}_i, t) \in \tilde{I}^k(t) \\ 0 & : \quad I(\mathbf{x}_i, t) \notin \tilde{I}^k(t) \end{cases} \tag{7}$$

To allow for illumination changes, impulse noise, environment clutter etc. we model the intensities of $I(t)$ by means of a p.d.f for each motion model

$$\begin{aligned} p^k(I(\mathbf{x}_i, t)|I(t-1), \mathbf{\Phi}(t)) &= p^k(I(\mathbf{x}_i, t)|\tilde{I}^k(\mathbf{x}_i, t), \sigma^k(t)) \\ &= p^k(I(\mathbf{x}_i, t)|I(t-1), \mathbf{u}_{\mathbf{a}^k}, \sigma_k(t)) \end{aligned} \tag{8}$$

where $k = 1, 2, \ldots, K$. Each $p^k(I(\mathbf{x}_i, t)|\tilde{I}^k(\mathbf{x}_i, t), \sigma_k(t))$ is assumed to be a Gaussian with mean $\tilde{I}^k(\mathbf{x}_i, t)$ and variance σ^k, i.e.,

$$p^k(I(\mathbf{x}_i, t)|I(t-1), \mathbf{\Phi}(t)) \sim \mathcal{N}(\tilde{I}^k(\mathbf{x}_i, t), \sigma^k(t)) \qquad k = 1, 2, \ldots, K \tag{9}$$

If we define the prediction error or residual for each of the k motion models at each pixel i, at time t and denote it by $r^k(\mathbf{x}_i, t)$, then

$$r^k(\mathbf{x}_i, t) = I(\mathbf{x}_i, t) - \tilde{I}^k(\mathbf{x}_i, t) \tag{10}$$

and $r^k(\mathbf{x}_i, t)$ is $\mathcal{N}(0, \sigma^k)$ Gaussian distributed $\forall k = 1, 2, \ldots, K$. We shall use $\mathbf{r}^k$ to denote the N dimensional vector $[r^k(\mathbf{x}_1, t), r^k(\mathbf{x}_2, t), \ldots, r^k(\mathbf{x}_N, t)]^T$ and $\mathbf{r}$ to denote $\{\mathbf{r}^k\}_{k=1}^K$. The parameter vector $\mathbf{\Phi}(t) = [\mathbf{\Theta}(t) \quad \mathbf{\Sigma}(t)]^T$ where $\mathbf{\Sigma}(t) = [\sigma^1 \sigma^2 \ldots \sigma^K]^T(t)$, and $\mathbf{\Theta}(t) = [\theta^1 \theta^2 \ldots \theta^K]^T(t)$ with $\theta^k = \mathbf{a}^k(t)$.

The objective of motion-based segmentation is to automatically assign to each pixel, $\mathbf{x}_i$, the vector $\mathbf{s}_i(t)$ on the basis of motion. Since motion information is implicit in the observations, the parameter $\mathbf{\Phi}(t)$ needs to be estimated.

[†]At any pixel $\mathbf{x}_i$, suppressing the dependence on time, the parametric velocity vector can be expressed as

$$\mathbf{u}_{\mathbf{a}}(\mathbf{x}_i) = \mathbf{X}(\mathbf{x}_i)\mathbf{a} \tag{1}$$

where

$$\mathbf{u}_{\mathbf{a}}(\mathbf{x}_i) = \begin{bmatrix} u(x_i, y_i) \\ v(x_i, y_i) \end{bmatrix} \tag{2}$$

$$\mathbf{X}(\mathbf{x}_i) = \begin{bmatrix} 1 & x_i & y_i & 0 & 0 & 0 & x_i^2 & x_i y_i \\ 0 & 0 & 0 & 1 & x_i & y_i & x_i y_i & y_i^2 \end{bmatrix} \tag{3}$$

$$\mathbf{a} = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \quad a_8]^T \tag{4}$$

We use Maximum Likelihood estimation to compute the ML estimate of $\Phi(t)$, denoted by $\hat{\Phi}_{ML}(t)$.

$$\begin{aligned}\hat{\Phi}_{ML}(t) &= \arg\max_{\Phi(t)} \Big\{ p(I(t)|I(t-1), \Phi(t)) \Big\} \\ &= \arg\max_{\Phi(t)} \Big\{ \log[p(I(t)|I(t-1), \Phi(t))] \Big\} \end{aligned} \tag{11}$$

The ML estimate $\hat{\Phi}_{ML}(t)$ can be interpreted as the value of $\Phi(t)$ which best explains the observation - $I(t)$ and $I(t-1)$.

Equation (11) is more general than the classical ML estimation problem, in that, part of the data (corresponding to $\mathbf{s}_i(t)$) is unobservable or hidden and hence is an instance of an "incomplete data problem". Expectation-Maximization[11] (EM) algorithm is a formal procedure to solve such incomplete data problems. The underlying assumption of the procedure is that the "complete" data includes not only the observed data, $I(t)$ and $I(t-1)$, but also "hidden" data, consisting of labels $\mathbf{s}_i(t), \forall i = 1, 2, \ldots, N$. Although this information is hidden we know that these variables should exhibit certain characteristics. In order to force the output of our segmentation procedure to reflect these characteristics we can introduce appropriate prior distributions. Specifically, we are interested in obtaining spatio-temporal segmentations that reflect the following two facts.

1. Neighboring pixels usually belong to the same object and hence should have the same labels.
2. Object boundaries, i.e., sites where the neighboring pixels of $\mathbf{s}(t)$ are not assigned the same labels, usually coincide with static intensity segmentation boundaries of $I(t)$.

In other words, $\mathbf{s}(t)$ should be modeled as a Markov Random Field (MRF). To bias the output segmentations appropriately, we use the following prior.

$$\begin{aligned} p(\mathbf{s}(t)) &= \frac{1}{Z} \exp\Big[-\beta \sum_{i=1}^{N} \Big(\lambda_1 \sum_{j \in \mathcal{N}_i} [1 - 2\delta(\mathbf{s}_i(t) - \mathbf{s}_j(t))] \delta(z_i(t) - z_j(t)) \Big) \Big] \\ &= \frac{1}{Z} \exp[-\beta U(\mathbf{s}(t)] \end{aligned} \tag{12}$$

where δ is the Kronecker delta function and $z(t)$ denotes the assigned label field for the static (gray level) image segmentation of the image frame $I(t)$. We shall assume that we can compute the static image segmentation fast enough to make it available to the motion-based segmentation procedure. As was pointed out earlier, we can use EM to simultaneously estimate the motion parameter vector $\Phi(t)$ as well as the segmentation $\mathbf{s}(t)$.

2.1. Recovering Motion Vectors and Support Regions Using EM Algorithm

The EM procedure compensates for the lack of "hidden" data by replacing them with their conditional expected values. To illustrate how the motion-based segmentation problem can be formulated as an incomplete data problem we introduce additional notation and reinterpret those defined before. The new notation subsumes the explicit dependence on time t and allows us to concentrate on developing the segmentation algorithm based on EM. Let

$\mathbf{h} = \{h_i, i \in \mathcal{S}\}$ denote hidden variables with prior distribution $p(\mathbf{h}|\Phi_h)$

$\mathbf{o} = \{o_i, i \in \mathcal{S}\}$ denote observations with likelihood $p(\mathbf{o}|\Phi_o, \mathbf{h})$

Let $\Phi = (\Phi_o, \Phi_h)$ denote the parameter vector. Further, we assume that the parameter vectors Φ_o and Φ_h are separable, i.e. $\Phi_o \bigcap \Phi_h = \phi$ where ϕ is the null set.

In our case,

$$\begin{aligned} \mathbf{h} &= \mathbf{s}(t) = \{\mathbf{s}_i(t), i \in \mathcal{S}\} \\ \mathbf{o} &= \{I(t), I(t-1)\} \\ \Phi_o &= \Phi(t) \end{aligned} \tag{13}$$

The complete data vector is denoted by $\mathbf{c}$, and it includes both the observed data, $\mathbf{o}$ and the hidden data, $\mathbf{h}$, i.e. $\mathbf{c} = \{\mathbf{o}, \mathbf{h}\}$. The EM algorithm attempts to solve the ML estimation problem:

$$\hat{\Phi}_{ML} = \arg\max_{\Phi} \log p(\mathbf{o}|\Phi) \tag{14}$$

To do this, it alternates between the following two steps

E step: Compute

$$Q\left(\Phi|\Phi^{(p)}\right) = E\{[\log p(\mathbf{o}|\mathbf{h}, \Phi) + \log p(\mathbf{h}|\Phi)]|\mathbf{o}, \Phi^{(p)}\} \tag{15}$$

M step: Compute

$$\Phi^{(p+1)}(t) = \arg\max_{\Phi} \quad Q\left(\Phi|\Phi^{(p)}\right) \tag{16}$$

where $E[\cdot]$ denotes expectation and p denotes the pth iteration. It has been shown that under some moderate regularity conditions, the estimates converge to ML estimates, at least locally.[12]

2.2. E Step: Computation of Expected Support by Mean Field Approximation

The E step of the EM procedure compensates for the lack of "hidden" data $\mathbf{s}(t)$ by replacing it with its conditional expected value denoted by $\mathbf{g}(t)$, where $g^k(\mathbf{x}_i, t) \in [0, 1]$ although $s^k(\mathbf{x}_i, t) \in \{0, 1\}$. The expected values are based on the current parameter estimate $\mathbf{\Phi}^{(p)}(t)$ and the observations - $I(t)$ and $I(t-1)$.

$$\begin{aligned} g^k(\mathbf{x}_i, t) &= E[s^k(\mathbf{x}_i)|I(t), I(t-1), \mathbf{\Phi}^{(p)}(t)] \\ &= Prob[I(\mathbf{x}_i, t) \in \tilde{I}_k(t)|I(t), I(t-1), \mathbf{\Phi}^{(p)}(t)] \end{aligned} \tag{17}$$

In terms of notation introduced in the previous subsection, $E[\mathbf{h_i}|\mathbf{o}, \Phi^{(p)}]$ needs to be computed. Since we have modeled $\mathbf{h}$ as an MRF, it can be shown[13] using results from Mean Field (MF) theory that $\mathbf{h}|\mathbf{o}, \Phi^{(p)}$ is an MRF. Hence, we can express

$$E[h_i|\mathbf{o}, \Phi^{(p)}] = Z_i^{mf'} \sum_{h_i} h_i \exp(-\beta U_i^{mf'}) \tag{18}$$

where $U_i^{mf'}$ is the local energy given by

$$U_i^{mf'} = h_i^T [\frac{-1}{\beta} W_1(o_i, \Phi^{(p)}) + V_1(\Phi^{(p)})] + \sum_{j \in \mathcal{N}_i} h_i^T V_2(\Phi^{(p)}) E[h_j|\mathbf{o}, \Phi^{(p)}] \tag{19}$$

and $Z_i^{mf'}$ is the local partition function. Note that in order to find the the mean field at i we need to find the mean field at the neighbors of i. Since both the E and the M steps are embedded in an iterative procedure the mean field is computed iteratively. The calculation of the mean field can be decomposed into local computations using Besag's coding method and can be implemented in parallel.

2.3. M Step: Estimating Model Parameters

The M step performs the maximization of the Q function to obtain the parameter estimate $\mathbf{\Phi}^{(p+1)}(t)$ for the next iteration. Since $\mathbf{s}_i$ was introduced in Section 2 to assign each pixel $\mathbf{x}_i$ to a single model we need to recover the **binary labeling**. We introduce the K dimensional vector $\mathbf{l}_i = [l_i^1 \quad l_i^2 \ldots \quad l_i^K]^T$ at each site i. Each label $l_i^j \in \{0, 1\}$ is binary and follows the notation

$$l_i^j = 1 \qquad \text{iff} \qquad j = \arg\max_{k \in 1,2,..K} E[h_i^k|\mathbf{o}, \Phi^{(p)}] \tag{20}$$

In other words, the labels $l_i^k, i = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, K$, are used to assign each observation (at site i) to a motion model k. This information is crucial for updating $\{\sigma^k\}_{k=1}^K$. To make the parameter update step robust, i.e., insensitive to outliers we use the robust estimate[14]

$$\sigma^{k,(p+1)} = 1.4826 \quad \text{median}_{i:l_i^k=1}|r^k(\mathbf{x}_i, t)| \tag{21}$$

In our problem of motion-based segmentation we also need to update the motion vector, $\mathbf{u}_{\mathbf{a}^k}$ (or corresponding parameter vector θ^k), $k = 1, 2, \ldots, K$, at each M step. From Equation (16) we can write

$$\Theta^{(p+1)}(t) = \arg\max_{\Theta(t)} \sum_{i=1}^{N}\sum_{k=1}^{K} E[s_i^k|\mathbf{r}, \Phi^{(p)}(t)] \quad \log p^k(r^k(\mathbf{x}_i, t)|\Phi(t)) \tag{22}$$

which can be rewritten as

$$\Theta^{(p+1)}(t) = \arg\min_{\Theta(t)} \sum_{i=1}^{N}\sum_{k=1}^{K} E[s_{ik}|\mathbf{r}, \Phi^{(p)}(t)]\Big(-\log p^k(r^k(\mathbf{x}_i, t)|\Phi(t))\Big) \tag{23}$$

To compute a robust estimate, we replace the negative log-likelihood by a robust function ρ which is related to the likelihood function through the choice of the distribution of residual or prediction error Thus, we have K minimizations to perform, one for each motion model, each of which is a weighted nonlinear minimization and can be expressed as

$$\theta_k^{(p+1)} = \arg\min_{\theta_k} \sum_{i=1}^{N} E[s_i^k|\mathbf{r}, \Phi^{(p)}(t)] \quad \rho(r^k(\mathbf{x}_i, t), \sigma_k) \qquad \forall k = 1, 2, ..., K \tag{24}$$

3. ROBUST MOTION-BASED SEGMENTATION

Atypical observations, or outliers, arise in the context of modeling when the brightness constancy constraint, implicit in Equation (5) is violated. This occurs when there is multiple motion at a pixel (Eg. overlays) or when there is occlusion/dis-occlusion of pixels. Regardless of the reason why outliers manifest themselves, atypical observations must be detected and removed from the data so that they do not corrupt the segmentation procedure. To do this, we apply Chebyshev bounds and validate observations at each iteration. We discard outliers and use only pertinent observations for the EM procedure.

3.1. The Algorithm

By specializing the EM algorithm to our problem and using a robust estimation formulation we can obtain a robust motion-based segmentation algorithm. As mentioned before, we embed the EM algorithm in a hierarchical coarse-to-fine framework.

Given: Current image frame $I(t)$ and previous image frame $I(t-1)$ both of dimension $X \times Y$, where usually $X = Y = 2^b$ for some positive integer b.

Preprocessing: Compute the multi-resolution image representation for $I(t)$ and $I(t-1)$. Let m represent scale or the level of the multi-resolution pyramid, with $m = MAXLEVEL$ referring to the coarsest level, and $m = 1$ referring to the finest level (corresponding to the original resolution of the frames). The static (gray level) segmentation is computed at each scale. The initialization step is carried out only at the coarsest level, i.e., $m = MAXLEVEL$.

Initialization: We set initial values for σ^k and $\theta^k = \mathbf{a}^k, k = 1, 2, \ldots, K$ where $\mathbf{a}$ is of dimension 6 (for affine motion model) and dimension 8 (for quadratic motion model). In particular, we choose each σ^k to be a large value and each θ^k to be the zero vector. These values are used to set $\Phi^{(0)}$.

Iterative Phase: At each resolution level, $m = MAXLEVEL, \ldots, 2, 1$:

FOR $p = 0$ to P do

1. Compute K *prediction* frames by warping the previous frame towards the current frame,
$$\tilde{I}^k(\mathbf{x}_i, t) = I(\mathbf{x}_i - \mathbf{u}_{\mathbf{a}^k}^{(p)}(\mathbf{x}_i, t), t-1), \qquad k = 1, 2, ..., K. \tag{25}$$
2. Define K *residual* frames in terms of the pixel residues using
$$r^k(\mathbf{x}_i, t) = I(\mathbf{x}_i, t) - \tilde{I}^k(\mathbf{x}_i, t), \qquad k = 1, 2, \ldots, K. \tag{26}$$
3. Validate data using Chebyshev bounds.
4. **E step:** Compute K dimensional *expected support* vector $E[\mathbf{s}_i(t)|\mathbf{r}, \Phi^{(p)}]$ at each site i using MF equations.
5. **M step:**
   (a) Compute K dimensional *label* vector $\mathbf{l}_i(t) = [l_i^1(t) \quad l_i^2(t) \ldots \quad l_i^K(t)]^T$ at each site i using Equation (20).
   (b) Compute updates for $\mathbf{\Sigma}(t)$ and $\mathbf{\Theta}(t)$ using Equation (21) and solving Equation (23).

ENDFOR

Project results to next level.

Analyzing the above algorithm reveals steps 4 and 5(b) to be computationally intensive. However, both steps can be parallelized easily. Hence, the algorithm can be implemented parallely in hardware for real-time applications. Results obtained through software implementation of the above algorithm are discussed next.

4. RESULTS

In order to test the algorithm we synthesize frames with known motion parameters as the input. We use an input sequence consisting of a moving block (foreground) against a static background. We start with 4 motion layers ($K = 4$) and let the algorithm decide the number of motion layers needed. Fig. 1 shows the two frames of the input sequence, $I(t-1)$ and $I(t)$. The two frames have been generated synthetically, using a motion vector of $[6 \quad 4]^T$ for a bright (textured) block moving against a dark (textured) background. Fig. 2 shows the support layers obtained. It can be seen that the algorithm converges to 2 motion layers - one each for the foreground and the background. The pixels belonging to the uncovered background manifest themselves as outliers. The results shown were obtained after just 10 iterations. The computed motion vector parameters are shown in Table 1.

Object	Background
5.9897	0
0.0003	0
0.0001	0
3.9538	0
0.0003	0
0.0016	0

Table 1. Computed parameters

Fig. 3 shows the difference between the current frame $I(t)$ and the predicted frame, $\hat{I}(t)$ based on support layering and motion modeling.

$$\begin{aligned} \hat{I}(t) &= \sum_{k=1}^{K} \hat{I}^k(t) \\ &= \sum_{k=1}^{K} \mathbf{s}^k(t) \odot \tilde{I}^k(t) \end{aligned} \tag{27}$$

where $\odot$ denotes pixelwise multiplication.

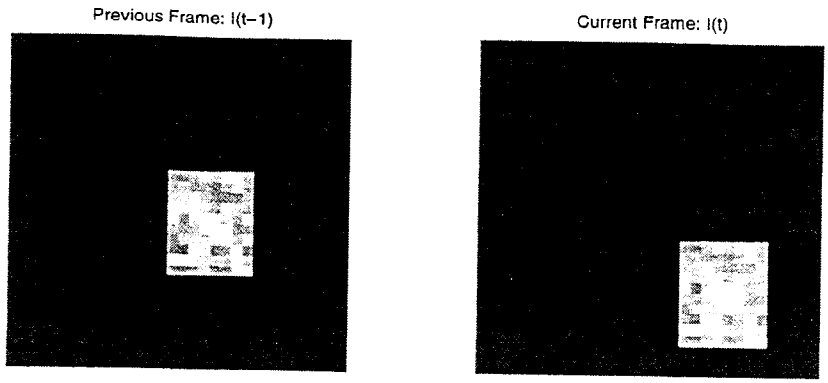

Figure 1. Two frames of the input sequence

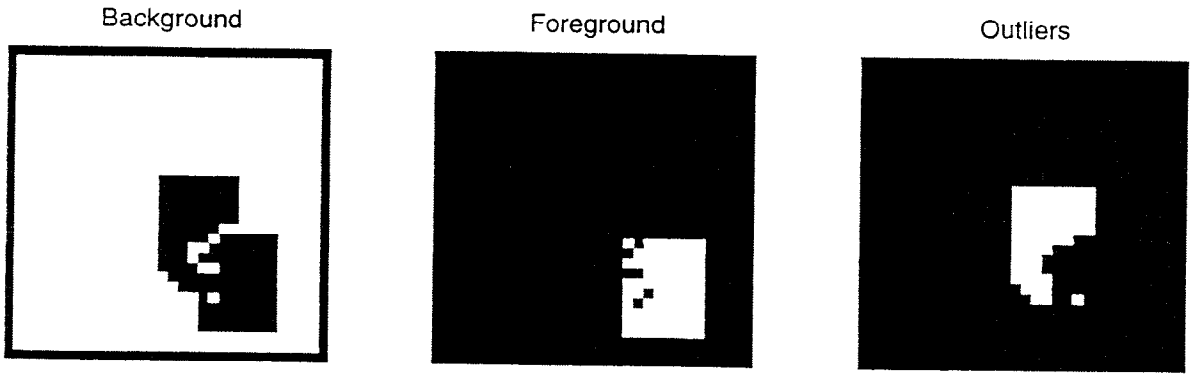

Figure 2. Support layers

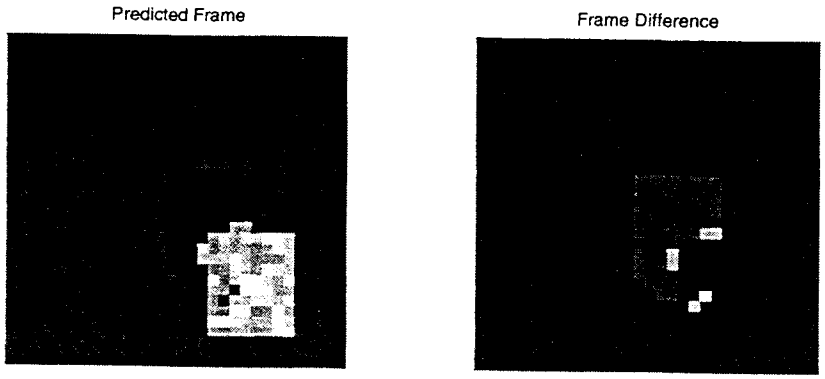

Figure 3. Predicted frame and frame difference

5. CONCLUSIONS

This paper provides a formalism for decomposing video frames into objects based on support layers and parametric motion models. It establishes the feasibility of using Mean Field approximation within the EM framework to compute segmentations and motion vectors for use in low bit rate video coding. The results shown here are preliminary and further research is underway to demonstrate the effectiveness of the algorithm in the case of natural sequences.

REFERENCES

1. M. Kunt, A. Ikomonopoulos, and M. Kocher, "Second generation image coding techniques," *Proc. of the IEEE* , pp. 549–574, Apr. 1985.
2. N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Processing: Image Communication, Vol. 3* , pp. 23–56, 1991.
3. J. K. Aggarwal, L. S. Davis, and W. N. Martin, "Correspondence processes in dynamic scene analysis," *Proceedings of IEEE, Vol. 69, No. 5* , pp. 562–572, May 1981.
4. J. Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Transcations on Image Processing, Vol. 3, No. 5* , pp. 625–638, Sept. 1994.
5. S. Z. Li, *Markov Random Field Modeling in Computer Vision*, Springer, 1995.
6. C. Stiller, "Object-based estimation of dense motion fields," *IEEE Transactions on Image Processing, Vol. 6, No. 2* , pp. 234–250, Feb. 1997.
7. J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-14* , pp. 910–927, Sept. 1992.
8. 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.
9. T. Darrell and A. Pentland, "Motion segmentation and qualitative dynamic scene analysis form an image sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 5* , pp. 474–487, May 1995.
10. H. Sawhney and S. Ayer, "Compact representations of videos through dominant and multiple motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 8* , pp. 814–830, 1996.
11. A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from incomplete data via the EM algorithm," *Journal of the Royal Society of Statistics, Ser. B, No. 1* , pp. 1–38, 1977.
12. R. A. Redner and H. F. Walker, "Mixture densities, Maximum Likelihood and the EM algorithm," *SIAM Review, Vol. 26* , pp. 195–239, 1984.
13. J. Zhang, "The Mean Field theory in EM procedures for Markov Random Fields," *IEEE Transactions on Image Processing, Vol. 40, No. 10* , pp. 2570–2583, Oct. 1992.
14. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley and Sons, New York, 1987.