Video Compression Using Spatiotemporal Regularity Flow

Orkun Alatas, Omar Javed, *Member, IEEE*, and Mubarak Shah, *Fellow, IEEE*

***Abstract*—We propose a new framework in wavelet video coding to improve the compression rate by exploiting the spatiotemporal regularity of the data. A sequence of images creates a spatiotemporal volume. This volume is said to be *regular* along the directions in which the pixels vary the least, hence the entropy is the lowest. The wavelet decomposition of regularized data results in a fewer number of significant coefficients, thus yielding a higher compression rate. The directions of regularity of an image sequence depend on both its motion content and spatial structure. We propose the representation of these directions by a 3-D vector field, which we refer to as the *spatiotemporal regularity flow (SPREF)*. *SPREF* uses splines to approximate the directions of regularity. The compactness of the spline representation results in a low storage overhead for *SPREF*, which is a desired property in compression applications. Once *SPREF* directions are known, they can be converted into actual paths along which the data is regular. Directional decomposition of the data along these paths can be further improved by using a special class of wavelet basis called the *3-D orthonormal bandelet basis*. *SPREF* -based video compression not only removes the temporal redundancy, but it also compensates for the spatial redundancy. Our experiments on several standard video sequences demonstrate that the proposed method results in higher compression rates as compared to the standard wavelet based compression.**

***Index Terms*—Image and video multi-resolution processing (2-MRP), wavelet based video coding (1-COD-WAVV).**

I. Introduction

VIDEO compression is very important for many applications, such as video conferencing, video storage, and broadcasting, since their performance largely relies on the efficiency of the compression. The most popular video compression algorithms are the ones used in the industrially accepted compression standards, MPEG1 [1], MPEG2 [2], and, recently, MPEG4 [3].

In MPEG1 and MPEG2, the frames are labelled as I, B, and P. Only the I (key) frames are compressed spatially. The P frames can be interpolated from the I frames, and B frames can be interpolated from both I and P frames using block motion compensation. This interpolation removes the temporal redundancy in B and P frames. The I frame compression algorithm is built on the JPEG image compression standard, which uses the discrete cosine transform (DCT).

Recently, the DCT transform has been found to be outperformed by the wavelet transform, which offers lower image distortion and, therefore, better visual performance at very low bit rates [4], [5]. Moreover, due to the desirable properties of the wavelets, such as scalability and coding efficiency, they have become very popular in a relatively short time. For example, the latest image compression standard, JPEG2000, uses the wavelet transform in its main compression algorithm [6].

A wavelet basis [7] consists of the dilated and translated versions of a wavelet function, ψ and a scaling function, ϕ. In image compression, a 2-D wavelet basis is used to decompose the data along fixed horizontal and vertical axes.

In a recent work, Mallat and Le Pennec proposed a wavelet-based image compression framework [8], where they exploited the image geometry to achieve higher compression rates. In this work, the geometry of an image patch is approximated with a 2-D vector field, called *geometric flow*. The geometric flow shows the local directions in which the patch varies regularly. Hence, compressing the patch with a wavelet basis along these directions outperforms the standard wavelet compression that takes place along fixed axes. The authors exploit the regularity along the flow lines further by *bandeletizing* the warped wavelet basis. In this step, the scaling functions are replaced with wavelet functions at higher levels so that the number of significant coefficients are reduced. The compression of an image with this framework involves its quad tree segmentation into smaller patches. After computing the compression cost of all nodes in the tree, the final segmentation is obtained by a split-merge algorithm that optimizes the compression cost of the whole image.

Besides image compression, the wavelets have also turned out to be a very useful tool in video compression. A wavelet video coding algorithm was proposed for the first time by Karlsson and Vetterli [9]. In this algorithm, a sequence is first segmented into *group(s) of frames (gof)*. Then, each *gof* is decomposed along the three major axes: temporal, horizontal, and vertical. However, this decomposition does not take the regularity of the *gof* into account.

In the presence of global motion in a *gof*, uniform 3-D paths of regularity are defined in the temporal direction, and these paths extend along the direction of motion. The situation gets more complicated when the motion is a mixture of the local and global components. In this case, *subgroups of frames (subgofs)*

Manuscript received November 21, 2004; revised May 26, 2006. The research work for this manuscript was carried out when O. Javed was at the University of Central Florida. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. David S. Taubman.

O. Alatas, deceased, was with the School of Electrical Engineering and Computer Science at the University of Central Florida, Orlando, FL 32816 USA.

O. Javed is with ObjectVideo, Reston, VA 20191 USA (e-mail: omar.javed@objectvideo.com).

M. Shah is with the School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: shah@cs.ucf.edu).

Color versions of Figs. 1–3 and 5–7 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2006.884930

with different motion types result in multiple directions of regularity. One way of modeling this regularity is to compute the motion to find the directions, in which the *gof* is regular. Once the pixel correspondence information is accumulated over multiple frames, it can be treated as a 3-D vector field that gives the directions of regularity of the *gof.* In wavelet-based approaches, the motion-compensated (MC) wavelet coding algorithms aim to use this solution. As a matter of fact, it has been shown that the compression along the motion directions is more efficient compared to the standard wavelet decomposition [10]. However, in MC wavelet coding, the motion information also needs to be coded in order to reconstruct the sequence. Therefore, the choice of the motion model is an important factor in such algorithms, as the precision and compressibility of its parameters directly affect the bit rate.

In the literature, two of the early studies on MC wavelet coding were carried out by Ohm [11] and Taubman [12], where they considered only the camera pan motion, and added a simple image registration step before the wavelet decomposition. In [10], Marpe and Cycon used *overlapped block motion compensation*, a technique that results in fewer blocking artifacts as compared to the standard block motion estimation. Han and Podilchuk, in [13], proposed using dense motion fields modeled by Markov random fields [14] to achieve accurate motion estimates. However, since the density of the motion field increased the bit rate, they implemented a lossy coding to encode the motion information. Later, Secker and Taubman [15] used deformable triangular meshes to model the motions more accurately. Wang and Fowler [16] also used deformable mesh models for motion compensation in their scheme, and employed *redundant wavelets* [17] for video coding. The deformable mesh model estimates the motion between two consecutive frames by placing a regular mesh grid on the first frame, and then computing the displacement of the mesh nodes in the second frame. Once the motion vectors of the mesh nodes are known, they can be used to interpolate the motion at any location. One of the problems with this model is that only consecutive pairs of frames are used to compute the directions of regularity of the whole *gof.* The 3-D vector field, constructed this way may not always be smooth. Moreover, storing the node displacement information for each frame pair creates a redundancy of motion information when the motions in the consecutive frames are similar. Another disadvantage is that the number of the nodes in the mesh models is fixed. When the motion is complex, its precise computation calls for denser meshes, which require an increased number of mesh nodes. This increase in nodes boosts the overhead of storing the motion parameters.

Another method for modeling the motion is the block motion model. This model attempts to remove the motion information redundancy by assigning a single flow vector to the blocks of image regions. However, the redundancy may still remain when the motions of (spatially and temporally) neighboring blocks are similar. The block motion model also does not result in a smooth 3-D vector field, since the motion vectors are constructed from pairs of frames. Therefore, an important issue in using motion compensation is to find a good representation that is both compact and accurate enough to model the motion well. Even when such a representation is found, the MC wavelets reduce to standard wavelets when there is no motion in the *gof.* This means that it cannot exploit the spatial regularity of the frames.

The solution to the above-mentioned problems lies in realizing that estimating the spatiotemporal directions of regularity of a *gof* requires the analysis of all frames simultaneously. When correctly formulated, this fusion of frame information can help compute smooth spatiotemporal directions in which the *gof* is regular as a whole. A general motion representation of the *gof* can also help discard the redundant (repetitive) motion information in case multiple frames have similar motion.

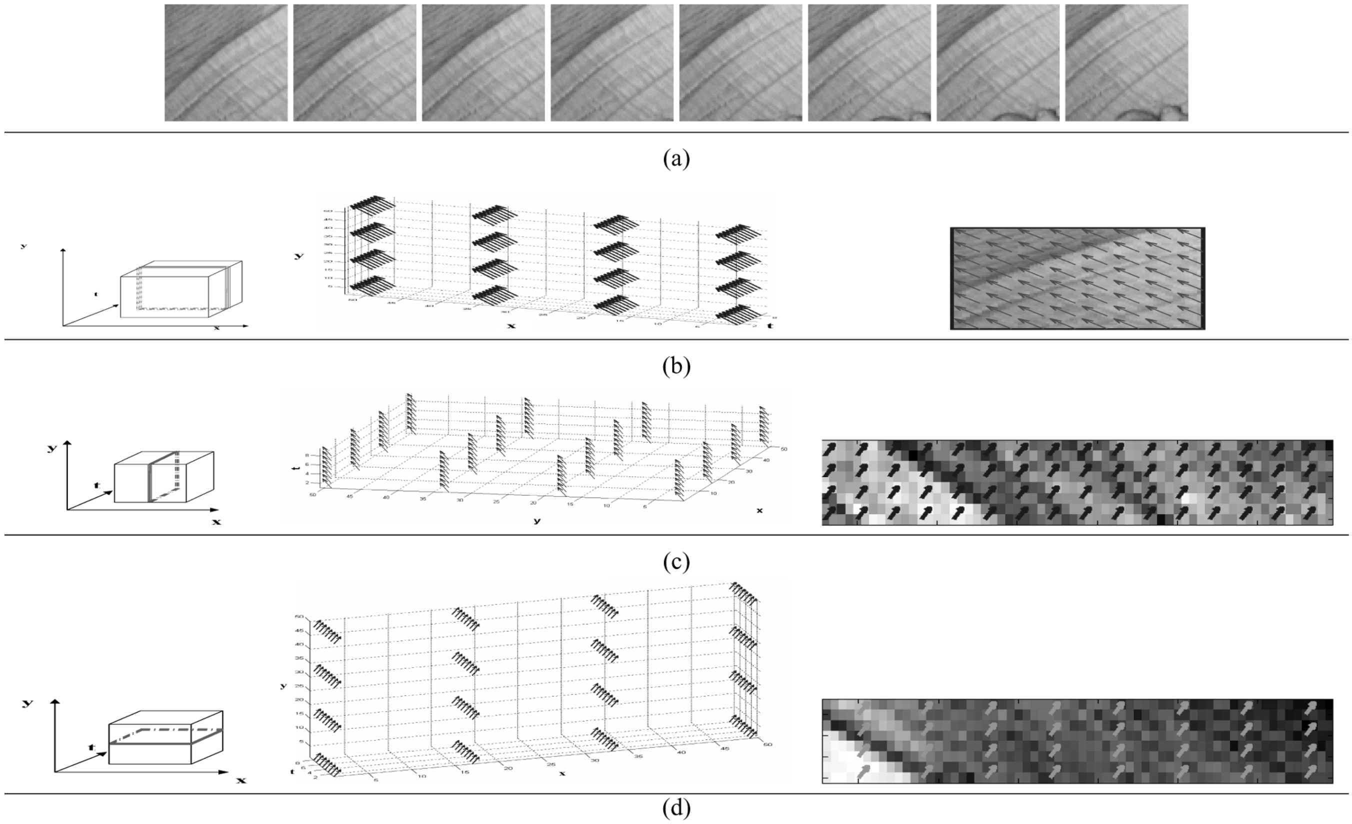
In this paper, we treat a *gof* not as a stack of frames but as a 3-D volume, and propose to model the spatiotemporal directions of regularity of this volume by a 3-D vector field, called the *spatiotemporal regularity flow (SPREF)*. The *SPREF* can be modeled in different ways, depending on whether the regularity is spatial or spatiotemporal. Once the flow is computed, a bandelet basis can be constructed to decompose the *gof* along its directions of regularity. To obtain the maximum compression, the *gof* needs to be partitioned into *subgofs*, whose regularities can be as closely modeled as possible by their respective *SPREFs*. We also propose an oct tree based algorithm to compute the segmentation of the *gof* according to its spatiotemporal properties, such that the reconstruction error and the bit rate are optimized.

The organization of the rest of the paper is as follows. In Section II, we explain the concept of *SPREF* and explain how it is computed for a spatiotemporal region. Section III deals with the construction of a bandelet basis for this region using *SPREFs*. In Section IV, we explain how this basis can be extended to compress the whole video, and present a segmentation algorithm for this purpose. After showing our results on various standard sequences in Section V, we conclude with final remarks in Section VI.

II. Spatiotemporal Regularity Flow (SPREF)

SPREF $(\zeta(x,y,t))$ is a 3-D vector field that shows the directions in which a spatiotemporal region, F, varies regularly. These directions are designed such that when a 3-D wavelet basis is warped along the *SPREF* directions, the resulting basis is also orthogonal. The orthogonality is guaranteed by a *planar (cross-sectional) parallelism* of the flow field, which is defined as all the vectors on a plane being equal in magnitude and direction. In our framework, a *cross-sectionally parallel* flow field can belong to one of the following three classes: 1) $x-y$ *parallel*, 2) $x-t$ *parallel*, and 3) $y-t$ *parallel*. In an $x-y$ *parallel* flow, the vectors on an $x-y$ cross section of the flow field for a particular t are cross-sectionally parallel. The planar parallelisms are defined similarly for the $x-t$ and $y-t$ *parallelisms*

The $x-y$ *parallel SPREF* models the directions of regularity that depend on the motion in the frames that constitute the spatiotemporal region. However, one should note that due to the *planar parallelism* constraint, the spatiotemporal regularity flow is not the same as the optical flow. The difference between these two types of flows is discussed in more detail in Section II-A. The other two classes of spatiotemporal regularity flow, i.e., $x-t$ and $y-t$ *parallel* flows, generally model the spatial regularity of F.



(a)

(b)

(c)

(d)

Fig. 1. Three types of *SPREF* fields for a *gof* that has a global motion along the diagonal. (a) The original synthetic sequence (eight frames). (b) (Left) The $x-y$ cross section of a *gof*. (Middle) The flow field shown for the $x-y$ *parallel* flow. (Right) The $x-y$ cross section of the flow at $t=1$, superimposed on the first frame of the *gof*. Similar explanations apply for (c) $y-t$ *parallel* flow and (d) $x-t$ *parallel* flow.

The regularity condition that *SPREF* needs to satisfy can also be perceived as a requirement to follow the directions, in which the sum of the directional gradients is minimum. Describing the problem in this way allows us to write a flow energy equation, defined over V (the support of F), for a ζ as

$$E(\zeta)=\int_V\left|\frac{\partial(F\star H)(x,y,t)}{\partial\zeta(x,y,t)}\right|^2 dxdydt \qquad (1)$$

where H is a regularizing filter, such as a Gaussian.

This continuous flow energy equation can be discretized, and then tailored to different types of parallelism depending on how ζ is defined. If the flow is $x-y$ *parallel*, then ζ is defined as $\zeta(x,y,t)=\zeta_{xy}(t)=(c_1'[t],c_2'[t],1)$, resulting in

$$E(\zeta_{xy})=\sum_V\left|\left(F\star\frac{\partial H}{\partial x}\right)c_1'[t]+\left(F\star\frac{\partial H}{\partial y}\right)c_2'[t]+F\star\frac{\partial H}{\partial t}\right|^2. \qquad (2)$$

Notice that the formulation of $\zeta_{xy}(t)$ implies that the x and y components of the flow, $(c_1'[t],c_2'[t])$, are functions of time only, which is constant for all the pixels of a certain frame, i.e., $x-y$ cross section of the *gof*. Fig. 1(a) shows the frames of a synthetic *gof*, which has been sampled from the Lena image by imitating a global translational motion in the diagonal direction. Hence, the direction of motion for all frames is uniform. Fig. 1(b), from left to right, shows a typical $x-y$ cross section of a *gof*, the subsampled $x-y$ *parallel* flow field (shown by the blue arrows), and, finally, the $x-y$ cross section of the flow at $t=1$, superimposed on the corresponding cross section of the *gof*.

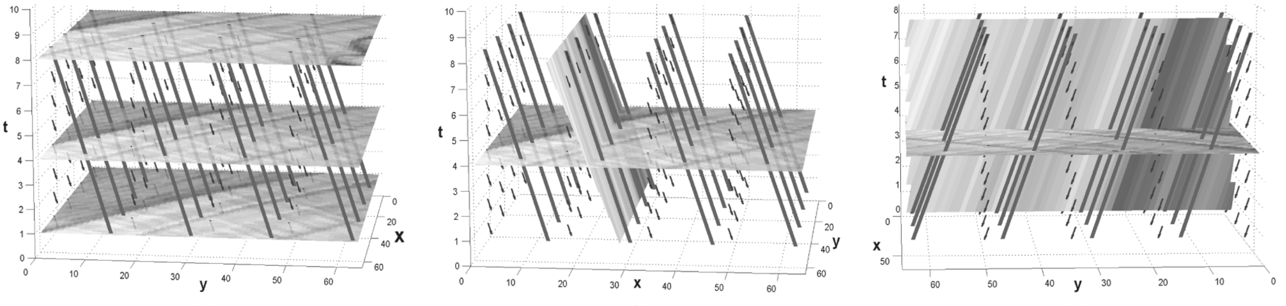
If the flow is $y-t$ *parallel*, then $\zeta(x,y,t)$ is formulated as $\zeta_{yt}(x)=(1,c_2'[x],c_3'[x])$. The vector, $(c_2'[x],c_3'[x])$, for a given x, is the same for all the pixels on the $y-t$ cross section of the flow. This definition of ζ results in the following flow energy equation:

$$E(\zeta_{yt})=\sum_V\left|F\star\frac{\partial H}{\partial x}+\left(F\star\frac{\partial H}{\partial y}\right)c_2'[x]+\left(F\star\frac{\partial H}{\partial t}\right)c_3'[x]\right|^2. \qquad (3)$$

Fig. 1(c) shows $\zeta_{yt}(x)$, and its first $y-t$ cross section at $x=1$, superimposed on the corresponding cross section of the *gof*. Notice that the flow directions still follow the motion.

For the $x-t$ *parallel* flow, ζ is defined as $\zeta(x,y,t)=\zeta_{xt}(y)=(c_1'[y],1,c_3'[y])$, and the expansion of (1) with this definition results in

$$E(\zeta)=\sum_V\left|\left(F\star\frac{\partial H}{\partial x}\right)c_1'[y]+F\star\frac{\partial H}{\partial y}+\left(F\star\frac{\partial H}{\partial t}\right)c_3'[y]\right|^2. \qquad (4)$$



(a) (b) (c)

Fig. 2. Three classes of *SPREF* curves for the *gof* in Fig. 1: (a)–(c) $x - y$ *parallel* Flow. Blue arrows show the *SPREF* directions, and the red curves show the *SPREF* curves. In (a), the 3-D view of the first, fourth, and eighth frames of the sequence in Fig. 1(b) are shown. (b) The same spatiotemporal region intersected along the flow curves. Finally, (c) is the frontal view of the intersecting plane. Notice how the *SPREF* curves extend in the directions, along which pixels change the least.

Fig. 1(d) shows $\zeta_{xt}(y)$, and the superimposition of its $x - t$ cross section on the *gof*. Similar arguments about the regularity of the flow apply here as well.

In order to convert the flow directions into actual spatiotemporal paths of regularity, the *SPREF curves* need to be computed. A spatiotemporal regularity flow curve, $c[u]$, is an integral curve, whose tangents are parallel to ζ. It defines the spatiotemporal paths, on which F varies regularly, and it can be used to warp the wavelet basis along the directions of regularity. The *SPREF* curve in the discrete domain is defined by the equation,

$$c_w[u] = \sum_{k=1}^{u} c'_w[k] \tag{5}$$

where $w = 1, 2, 3$. The coordinates of an $x - y$ *parallel* flow curve are given as, $(x + c_1[t], y + c_2[t], t)$ for a constant (x, y) and a varying t. If the flow is $x - t$ *parallel*, the flow curve coordinates are $(x + c_1[y], y, t + c_3[y])$ for constant (x, t) and varying y. Finally, if the flow is $y - t$ *parallel*, then $(x, y + c_2[x], t + c_3[x])$ for constant (y, t) and varying x gives the flow curve coordinates.

Fig. 2(a) shows the first, fourth, and eighth frames of the sequence in Fig. 1(a), and its $x - y$ *parallel SPREF* directions (blue arrows) in 3-D. The red lines show the *SPREF* curves, on which the *gof* varies regularly. Notice that the curves extend along the direction of motion. In Fig. 2(b), the side view of the *gof*, intersected along the flow curves is shown. Finally, in Fig. 2(c), the intersecting plane is viewed from the front. It is clearly visible that the flow curves show the directions on which the pixels change the least. In Fig. 2(d) and (e), the *SPREF* curves of the flows in Fig. 1(d) and (f) are shown. Notice that the curves extend along the direction of motion regardless of the flow type since motion is the factor that determines the regularity in this sequence.

One of the most important requirements for the flow representation is that it should be compact, such that the overhead it will introduce to the compression is minimum. In order to satisfy this condition, and also to obtain a smooth flow, the directions, $c'_m[u]$ $(\mathrm{m} \in \{1, 2, 3\})$, are approximated with the translated box spline functions of the first degree, $S(u)$, as

$$c'_m[u] = \sum_n \alpha_n^m S(2^{-l}u - n) \tag{6}$$

where α_n $(n = 1 \ldots 2^l)$ is the nth spline coefficient, $l = 1 \cdots k$ is a scale factor, 2^k is the width of F_i on the axis along which the flow is not planarly parallel, and u is an index of this axis $(u \in \{x, y, t\})$. With this representation, the whole spatiotemporal regularity flow can be recovered by storing only the spline control point values. The spline function, $S(u)$, we used in our experiments is formulated as

$$S(u) = \begin{cases} 1 - |u|, & \text{if } |u|\ 1 \\ 0, & \text{otherwise.} \end{cases}$$

The coefficients, α_n, are solved for by quadratic minimization of the energy functions (2), (3), or (4), the choice depending on the parallelism class. In the final step, these coefficients are quantized. The selection of the quantization parameter depends on the precision requirement of the flow.

A. $x - y$ *Parallel SPREF versus Optical Flow*

When the spatiotemporal regularity flow is $x - y$ *parallel*, its directions and magnitudes resemble those of the optical flow but there are some differences between the two types of flows. The true optical flow $(u[x, y], v[x, y])$ gives the directions of highest regularity between two frames as a function of the spatial location. These directions for all the frames of a *subgof* can form a 3-D flow field, such that $\zeta(x, y, t) = (u[x, y, t], v[x, y, t], 1)$. However, the spatiotemporal components of the true optical flow field are not regularized; hence, they do not necessarily provide a one-to-one mapping of the pixels in consecutive frames in 3-D. When this is the case, they cannot be used to form a path of decomposition that will result in a perfect reconstruction of the data. On the other hand, the $x - y$ *parallel SPREF* imposes a regularity condition on the flow such that the pixels are one-to-one mapped through optimal translations

$(c_1[t], c_2[t])$ for a *gof*, such that the cumulative spatiotemporal regularity flow error is minimized. If all the pixels in a frame have the same motion, then the *SPREF* and the optical flow are the same. However, if the true optical flow is a function of the spatial location, i.e., when the motion is a rotation or zooming in/out, then *SPREF* tries to find the best translations for each frame of the *gof* that can approximate those motions. Moreover, *SPREF* is more suitable than optical flow since it contains much less redundancy due to the spline representation. Further reduction of this redundancy is also possible by a final segmentation process, which will be described later in detail, in Section IV.

III. Orthonormal Spatiotemporal Bandelet Basis

SPREF is designed so that it can be used to warp the 3-D wavelet basis along the directions which a spatiotemporal region, F, is regular. Once the wavelet basis is warped, it can be converted into a *bandelet* basis to take further advantage of the regularity along the flow directions. In this section, we will explain how a 3-D orthonormal bandelet basis can be constructed with a given *SPREF*, and describe an algorithm for the bandelet decomposition of F.

A. Constructing the Orthonormal Bandelet Basis

The standard orthonormal 3-D wavelet basis decomposes the data in temporal, vertical and horizontal directions. The order of these directions is not important due to the separability of the basis. This basis can be written as

$$\left\{\begin{array}{l}
\psi_{j,m_1}(x)\psi_{j,m_2}(y)\phi_{j,m_3}(t)\\
\psi_{j,m_1}(x)\psi_{j,m_2}(y)\psi_{j,m_3}(t)\\
\phi_{j,m_1}(x)\psi_{j,m_2}(y)\phi_{j,m_3}(t)\\
\phi_{j,m_1}(x)\psi_{j,m_2}(y)\psi_{j,m_3}(t)\\
\psi_{j,m_1}(x)\phi_{j,m_2}(y)\phi_{j,m_3}(t)\\
\psi_{j,m_1}(x)\phi_{j,m_2}(y)\psi_{j,m_3}(t)\\
\phi_{j,m_1}(x)\phi_{j,m_2}(y)\psi_{j,m_3}(t)
\end{array}\right\}_{(j,m_1,m_2,m_3)\in I_V}$$

where $\psi_{j,m}(u) = (1/\sqrt{2^{j)}}\psi(u-2^j m/2^j)$ is the wavelet function, ϕ is the scaling function, $\phi_{j,m}(u) = 1/\sqrt{2^j}\phi(u-2^j m/2^j)$, j and m are the scale and translation factors. V is the support of F and is a subset of $\mathbb{R}^3$. I_V is an index set that depends upon the geometry of the boundary of V.

This standard set of decomposition directions can be described by certain values of the *SPREF* classes such as $\zeta_{xy}(t) = (0,0,1)$, $\zeta_{xt}(y) = (0,1,0)$, or $\zeta_{yt}(x) = (1,0,0)$. Notice that, ideally, these values of ζ are suitable only when the frames are regular along the purely temporal, horizontal or vertical directions. However, when the directions of regularity depend on the motion and/or the spatial structure of the scene, which they always do, the optimal decomposition might lie along different directions. If the true flow, ζ, of a *subgof* is known, then the decomposition of the data along the flow directions can be carried out using the orthonormal bandelet basis. The construction of the bandelet basis consists of two steps: 1) warping the standard 3-D wavelet basis and 2) *bandeletization*.

The standard wavelet basis can be warped along the flow curves with the operator W, which is defined as $W_{xy}(F(x,y,t)) = F(x+c_1[t], y+c_2[t], t)$ for the $x-y$ *parallel* flow, $\zeta_{xy}(t)$. Alternatively, F can be warped instead of the basis itself, and the standard wavelet basis can be defined in the warped domain. Then, the transformation of this basis back to the original domain produces the *warped wavelet basis*

$$\left\{\begin{array}{l}
\psi_{j,m_1}(x-c_1[t])\psi_{j,m_2}(y-c_2[t])\phi_{j,m_3}(t)\\
\psi_{j,m_1}(x-c_1[t])\psi_{j,m_2}(y-c_2[t])\psi_{j,m_3}(t)\\
\phi_{j,m_1}(x-c_1[t])\psi_{j,m_2}(y-c_2[t])\phi_{j,m_3}(t)\\
\phi_{j,m_1}(x-c_1[t])\psi_{j,m_2}(y-c_2[t])\psi_{j,m_3}(t)\\
\psi_{j,m_1}(x-c_1[t])\phi_{j,m_2}(y-c_2[t])\phi_{j,m_3}(t)\\
\psi_{j,m_1}(x-c_1[t])\phi_{j,m_2}(y-c_2[t])\psi_{j,m_3}(t)\\
\phi_{j,m_1}(x-c_1[t])\phi_{j,m_2}(y-c_2[t])\psi_{j,m_3}(t)
\end{array}\right\}_{(j,m_1,m_2,m_3)\in I_{V'}}$$

where V' is the support for the warped region. This basis decomposes F along the directions of regularity, where the entropy of the data is lower. Hence, the resulting number of significant coefficients is also lower than those of the standard wavelet basis, which implies a higher compression rate. The warped wavelet bases for the $y-t$ and $x-t$ *parallel* spatiotemporal regularity flows can be written in the same manner using the warping operators $W_{yt}(F(x,y,t)) = F(x, y+c_2[x], t+c_3[x])$ and $W_{xt}(F(x,y,t)) = F(x+c_1[y], y, t+c_2[y])$, respectively.

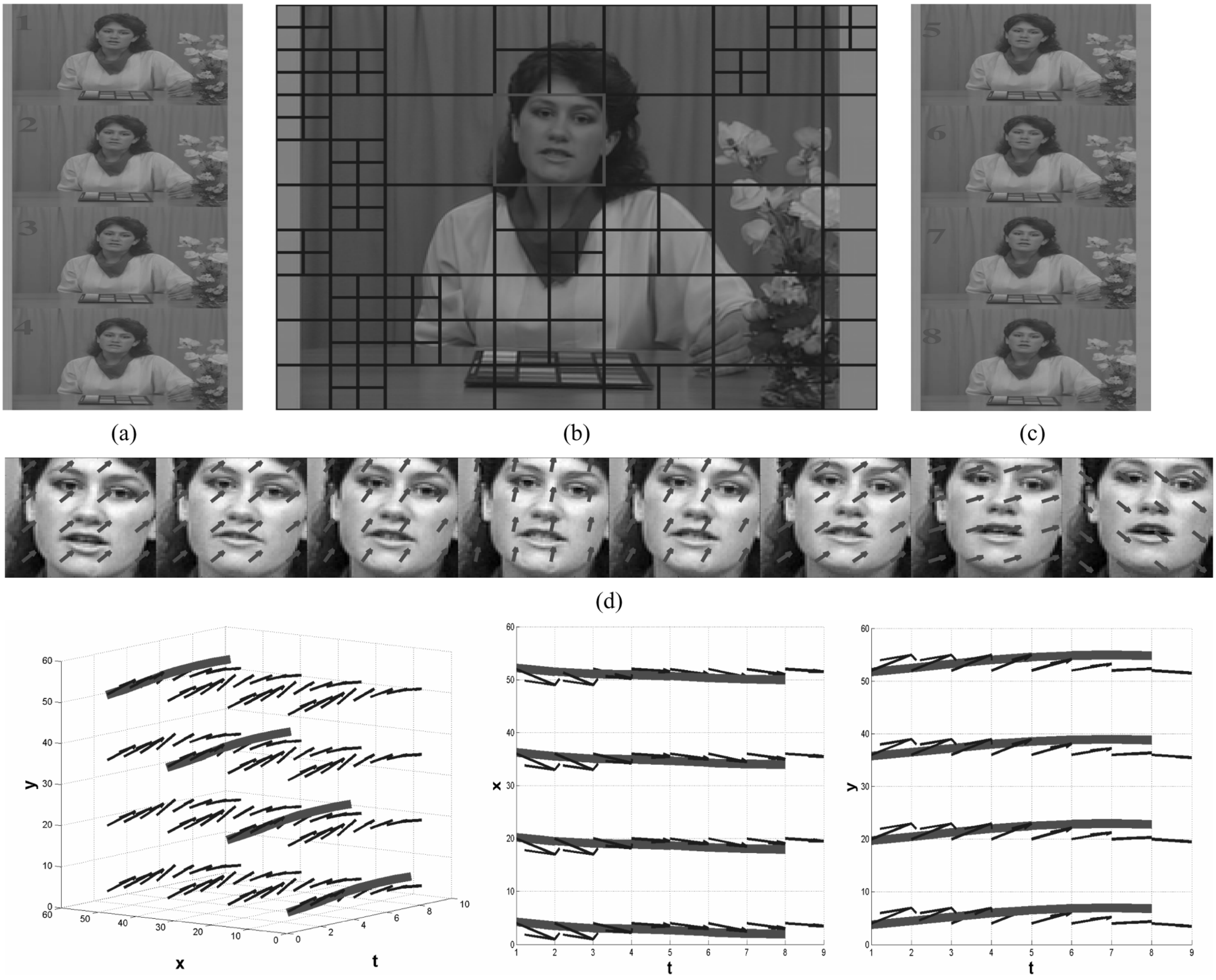
After warping the wavelet basis, the last step is the *bandeletization* [8]. The wavelet function family $\{\psi(t)\}_{j,m3}$ consists of high-pass filters and it has a vanishing moment at lower resolutions. The scaling function family $\{\phi(t)\}_{j,m3}$, however, consists of low-pass filters, and it does not have a vanishing moment at lower resolutions. Hence, it cannot take advantage of the regularity of the *gof* along the flow curves. In order to solve this problem, the warped wavelet basis is *bandeletized* by replacing $\{\phi(t)\}_{j,m3}$ with $\{\psi(t)\}_{l,m3}$ for $l > j$. After the *bandeletization*, the orthonormal bandelet family is written as follows:

$$\left\{\begin{array}{l}
\psi_{j,m_1}(x-c_1[t])\psi_{j,m_2}(y-c_2[t])\underline{\psi_{l,m_3}(t)}\\
\psi_{j,m_1}(x-c_1[t])\psi_{j,m_2}(y-c_2[t])\psi_{j,m_3}(t)\\
\phi_{j,m_1}(x-c_1[t])\psi_{j,m_2}(y-c_2[t])\underline{\psi_{l,m_3}(t)}\\
\phi_{j,m_1}(x-c_1[t])\psi_{j,m_2}(y-c_2[t])\psi_{j,m_3}(t)\\
\psi_{j,m_1}(x-c_1[t])\phi_{j,m_2}(y-c_2[t])\underline{\psi_{l,m_3}(t)}\\
\psi_{j,m_1}(x-c_1[t])\phi_{j,m_2}(y-c_2[t])\psi_{j,m_3}(t)\\
\phi_{j,m_1}(x-c_1[t])\phi_{j,m_2}(y-c_2[t])\psi_{j,m_3}(t)
\end{array}\right\}_{(j,l>j,m_1,m_2,m_3)}$$

where the underlined are the replaced functions in the warped wavelet basis. A similar process is followed for bandeletization if the flow is $y-t$ *parallel*, i.e., the bandeletization is done by replacing the scaling function family $\{\phi(x)\}_{j,m1}$ with the wavelet function family $\{\psi(x)\}_{l,m1}$ at lower resolutions, $l > j$. if the flow is $x-t$ *parallel*, then the bandeletization requires replacing the family $\{\phi(y)\}_{j,m2}$ with $\{\psi(y)\}_{l,m2}$ at lower resolutions, $l > j$.

B. Orthonormal Bandelet Decomposition

Decomposing F using an orthonormal bandelet basis can be implemented by an algorithm that is very similar to the discrete



(a) (b) (c)

(d)

(e)

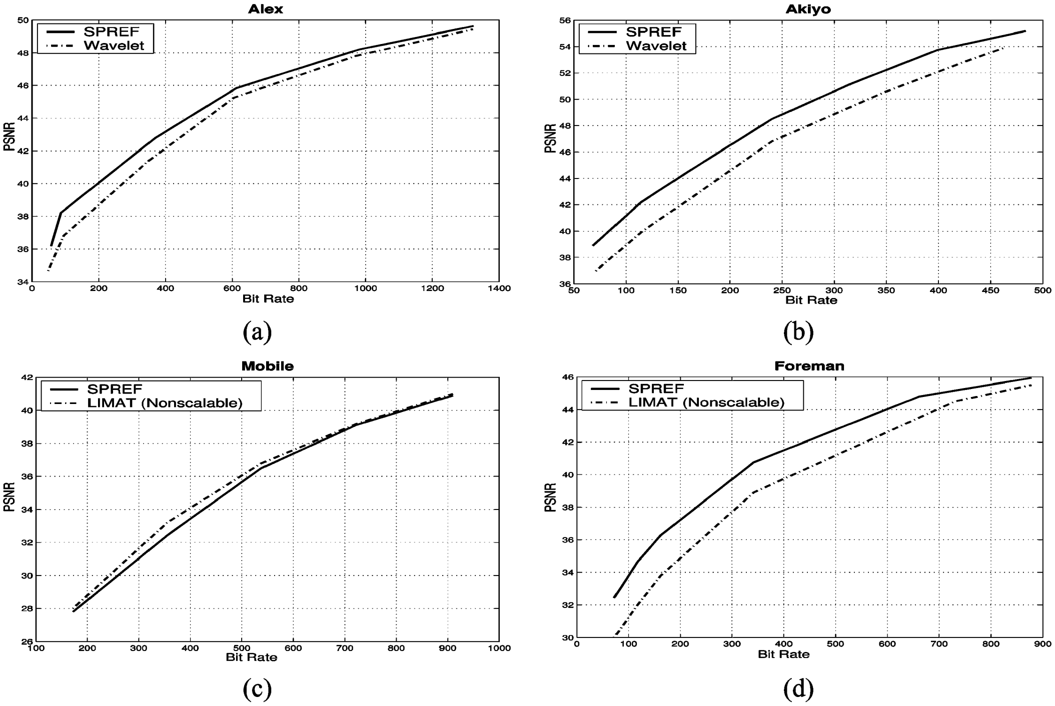
Fig. 3. Results for the frames 98–105 of the Alex sequence at 1000 kbps. [Columns (a) and (c)] The original frames. (b) The optimal segmentation of the frames at 1000 kbps. The *SPREF* of the region marked with a red rectangle is analyzed further in (d) and (e). In (d), the *SPREF* is superimposed on the mini frames. The correctness of the motion directions can be observed by tracing Alex's chin (y) and left ear (x). In (e), the *SPREF* is shown as a 3-D vector field, where it is shown from oblique, top and side views, respectively.

wavelet transform. The main steps of the orthonormal bandelet decomposition can be summarized as follows.

1) Compute the *SPREF* of the region, F, according to (2), (3), or (4). A 3-D Gaussian filter can be used as the regularizing filter, H.

2) Compute the warped wavelet transform coefficients of F with the quantized flow parameters.

   (a) F can be decomposed with a warped wavelet basis by a subband filtering that goes along the flow curves. The subband filtering can be implemented by using the lifting scheme [18], which requires that the right and left neighbors of a point be known. In the standard wavelet decomposition, the temporal neighbors of a pixel at the location (x,y,t) are located at $(x,y,t-1)$ and $(x,y,t+1)$. However, in the warped wavelet filtering, these neighborhoods have to be defined on the flow curves. In order to establish this neighborhood relationship, the curve coordinates are stored in a grid $G(x,y,t)$. If the parallelism is of the $x-y$ class, then $G(x,y,t)=(x+c_1[t],y+c_2[t],t)$ if $(x+c_1[t],y+c_2[t],t)\in V_i$ and is null otherwise. Since the $x-y$ *parallel* flow curves are the sets of points with constant (x,y) and varying t, the pixels stored at the locations $G(x,y,t-1)$, $G(x,y,t)$ and $G(x,y,t+1)$ are temporal neighbors on the same curve. The spatial neighborhoods of the pixels, on the other hand, are still defined based on their spatial coordinates, not the flow curves. Similarly, if the parallelism is of the $x-t$ class, then G defines the horizontal neighborhoods, i.e., $G(x,y,t)=(x+c_1[y],y,t+c_3[y])$ if $(x+c_1[y],y,t+c_3[y])\in V.G$ is null otherwise. The neighbors of the pixel stored in $G(x,y,t)$ are stored in $G(x,y-1,t)$ and $G(x,y+1,t)$. Finally, if the parallelism is of the $y-t$ class, then G is defined as



(a) (b) (c) (d)

Fig. 4. Bit-rate versus PSNR plots of (a) Alex, (b) Akiyo, (c) mobile, and (d) foreman sequences. The plots (a) and (b) show the comparison of *SPREF*-based compression and standard (non motion-compensated) wavelet video compression. The plots (c) and (d) show a comparison of *SPREF*-based compression and the *LIMAT* (motion-compensated) framework.

$(x, y + c_2[x], t + c_3[x])$ if $(x, y + c_2[x], t + c_3[x]) \in V.G$ is null otherwise. The neighboring pixel coordinates are stored in $G(x-1, y, t)$ and $G(x+1, y, t)$. Note that when $G(x, y, t)$ is a noninteger value, the data has to be carefully sampled at that point. In our experiments, we used nearest-neighbour interpolation technique to prevent loss of data during sampling.

3) Bandeletize the warped wavelet coefficients.
   (a) The coefficients resulting from the scaling function, $\phi(u)$ $(u \in \{x, y, t\}$, depending on the parallelism class) are further decomposed, using subband filtering at the lower resolutions. This concludes the bandelet transformation of the *subgof*.

The reconstruction of the *gof* is implemented by inverting the decomposition steps. In order to do this, the coordinate grid, G, needs to be reconstructed. Once G is known, the rest is simply inverse filtering of the data along the flow curves.

1) Compute the grid, G, from the quantized spline coefficients.
2) Unbandeletize the quantized bandelet coefficients by inverse subband filtering to recover the warped wavelet coefficients.
3) Apply inverse subband filtering steps once more along the flow curves to reconstruct the *subgof*.

IV. Video Compression

In wavelet video compression, a video sequence is first partitioned into *gof*, then each *gof* is compressed separately. Our aim is to achieve a higher compression rate by analyzing the directions of regularity of the *gof*, which are modeled by *SPREF*. The direct extension of the compression we described in the previous section to a video sequence does not always work because most of the times the directions of regularity of a *gof* are not uniform. This becomes the case when the *gof* has multiple directions of regularity due to different types of motions taking place in the video, and/or the different spatial arrangements in it.

The solution to this problem is segmenting a *gof* into *subgofs*, such that the directions of regularity of each *subgof* is as closely estimated as possible by its corresponding *SPREF*. The challenge here is finding the optimal segmentation of the *gof*'s support, V, into V_is, $(V = \bigcup_i V_i)$, so that the compression cost

$$D + \lambda R = \sum_i D_i + \lambda R_i \tag{7}$$

is minimized, where D_i

$$D_i = \sum_x \sum_y \sum_t (F_{i,\text{original}}(x, y, t) - F_{i,\text{reconstructed}}(x, y, t))^2 \tag{8}$$

is the sum of squared reconstruction error of F_i, R_i is the bit cost of the encoded decomposition coefficients and *SPREF* parameters, and λ is a Lagrange multiplier. In order to achieve this segmentation, we initially partition the *gof* into rectangular prisms (cuboids) using an *oct tree* data structure,The width of each dimension of a cuboid is 2^{k_j}, where $j \in \{1, 2, 3\}$ denotes the particular dimension. The bit cost (R_i) of a *subgof* consists of three different types of costs: $R_{c,i}$ (the position and size of F_i), $R_{s,i}$ (the *SPREF* coefficients) and $R_{b,i}$ (the quantized bandelet/wavelet coefficients). $R_{s,i}$ also encodes whether a certain type of spatiotemporal regularity flow exists or not. If a *subgof*,

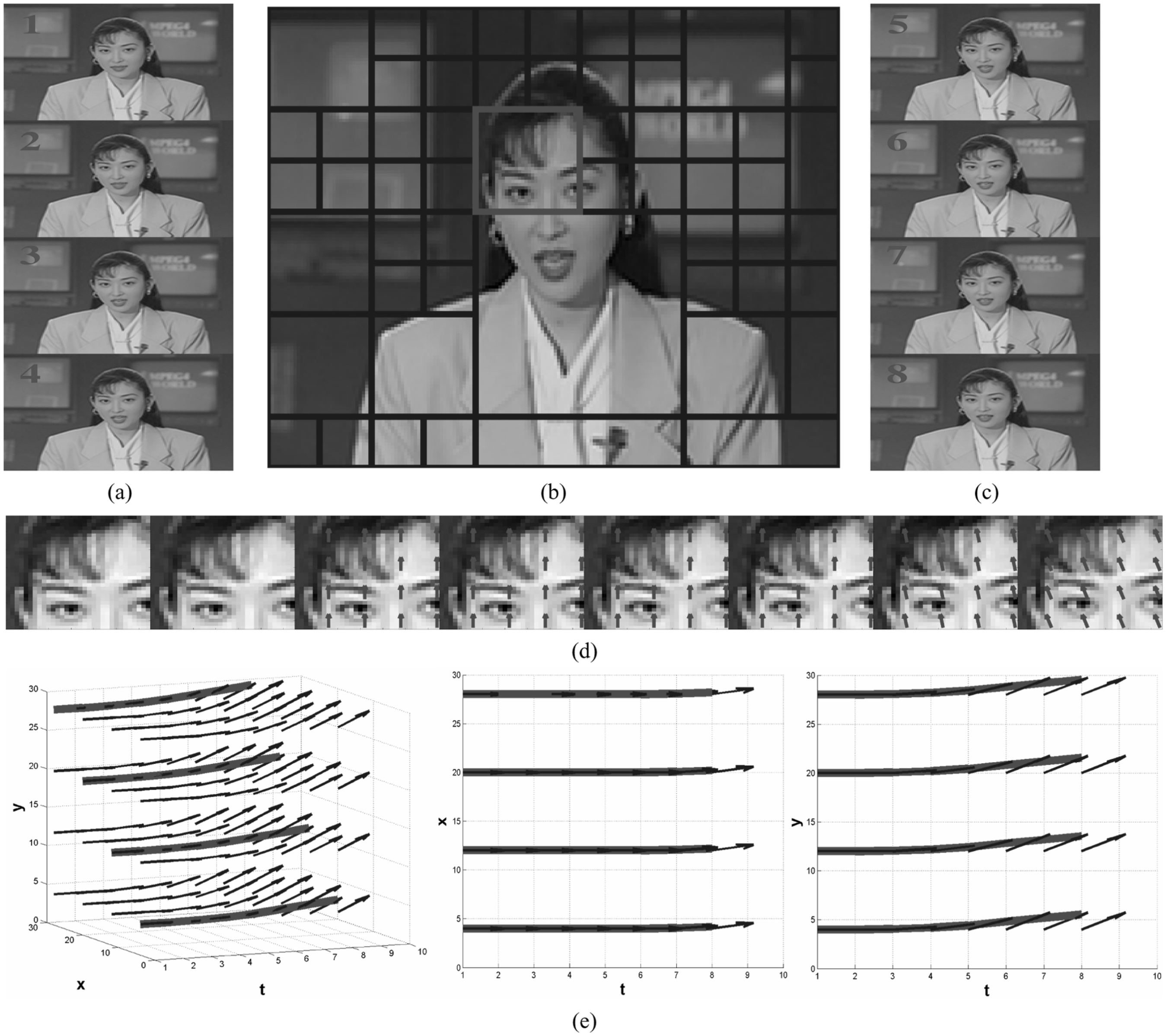
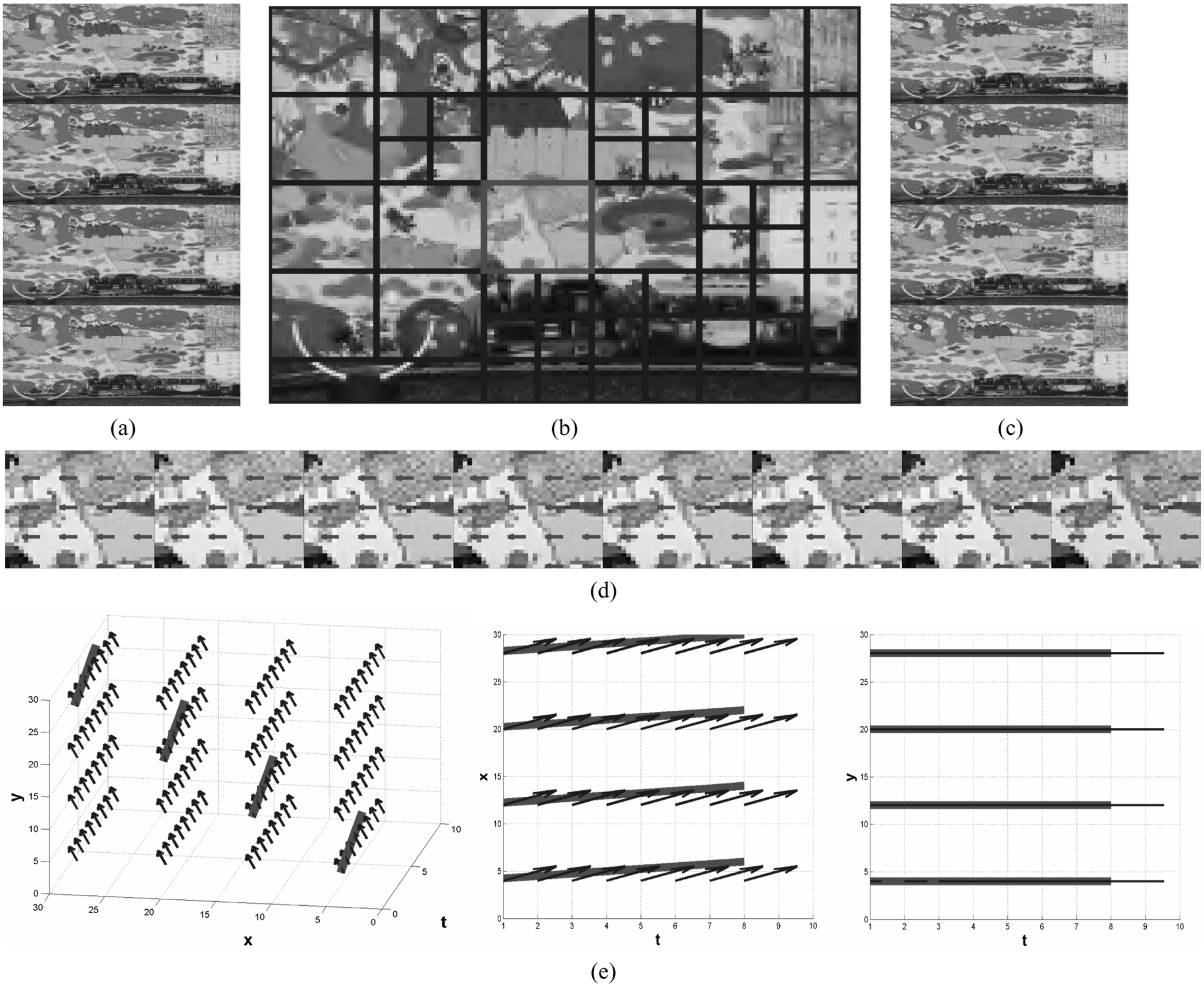


Fig. 5. Results for frames 11–18 of the Akiyo sequence compressed at 480 kbps. (a), (c) Original frames; (c) shows the optimal segmentation at this bit rate. The motion of Akiyo's head results in segmentation of the region that contains both the static background and her head. Moreover, even the static regions are segmented according to their regularity. For example, most of the corners in the background are segmented where they contain both horizontal and vertical edges. The *SPREF* analysis of the region marked with red rectangles is shown in (d) and (e). In (d), the global motion of Akiyo's head can be seen by following her brows. This motion is captured by the *SPREF* and shown by the red arrows superimposed on the frames. In (e), the same *SPREF* is shown from oblique, top and side views.

F_i, consists of spatially smooth frames with no motion, then its directions of regularity are not well defined; hence, there exists no *SPREF*. In this case, F_i is compressed with the standard wavelets. If the flow is defined, then it is represented by $n = 2^{(k_j - l_{a_1})} + 2^{(k_j - l_{a_2})}$ quantized spline parameters (α_n), where a_1 and a_2 are the axes that constitute the plane of the flow parallelism. The scale parameter, l_{a_j}, and the flow coefficients are coded with fixed length codes. The bandelet/wavelet coefficients to be stored in $R_{b,i}$ are encoded using 3-D SPIHT encoding [19], followed by a run-length encoding step.

The minimization of the sum in (7) starts with computing the compression cost for all cuboids in the oct tree. The cost, $(D_i + \lambda R_i)$, can be minimum for only one of the four flow hypotheses, including the no-flow case. When computing the cost of the *SPREF*s for a certain parallelism, the optimal scale parameter $l (1 \leq l \leq k)$ in (6) is found by trying all possible values of l, and selecting the one that results in the smallest compression cost. In the end, the flow that results in the minimum cost determines the flow type of F_i. The optimal segmentation of V is found by a split/merge algorithm starting from the leaf nodes (the smallest cuboids) of the oct tree. At each level, eight child nodes are merged into a single node if their cumulative cost is greater than the parent's cost, otherwise they stay split. This merging constraint can be formulated as

$$D_i + \lambda R_i < \sum_{q=1}^{8} D_{i,q} + \lambda R_{i,q} \qquad (9)$$



(a) (b) (c)

(d)

(e)

Fig. 6. Results for a *gof* of the Mobile sequence (frames 99–106) at 350 kbps. (a), (c) Original frames; (b) shows the segmentation at this bit rate, with a marked region, whose *SPREF* is analyzed in (d) and (e). Notice that the train, calendar, and the moving background have different motions; hence, the *gof* is segmented in these regions so that each region can contain a certain type of motion. The motion of the ball and the swinging toy cannot be modeled well with *SPREF*; hence, those regions stay unsegmented. In (d), it can be seen that the horizontal motion of the background is captured by *SPREF* and superimposed on the mini frames. (e) *SPREF* from oblique, top, and side views, respectively.

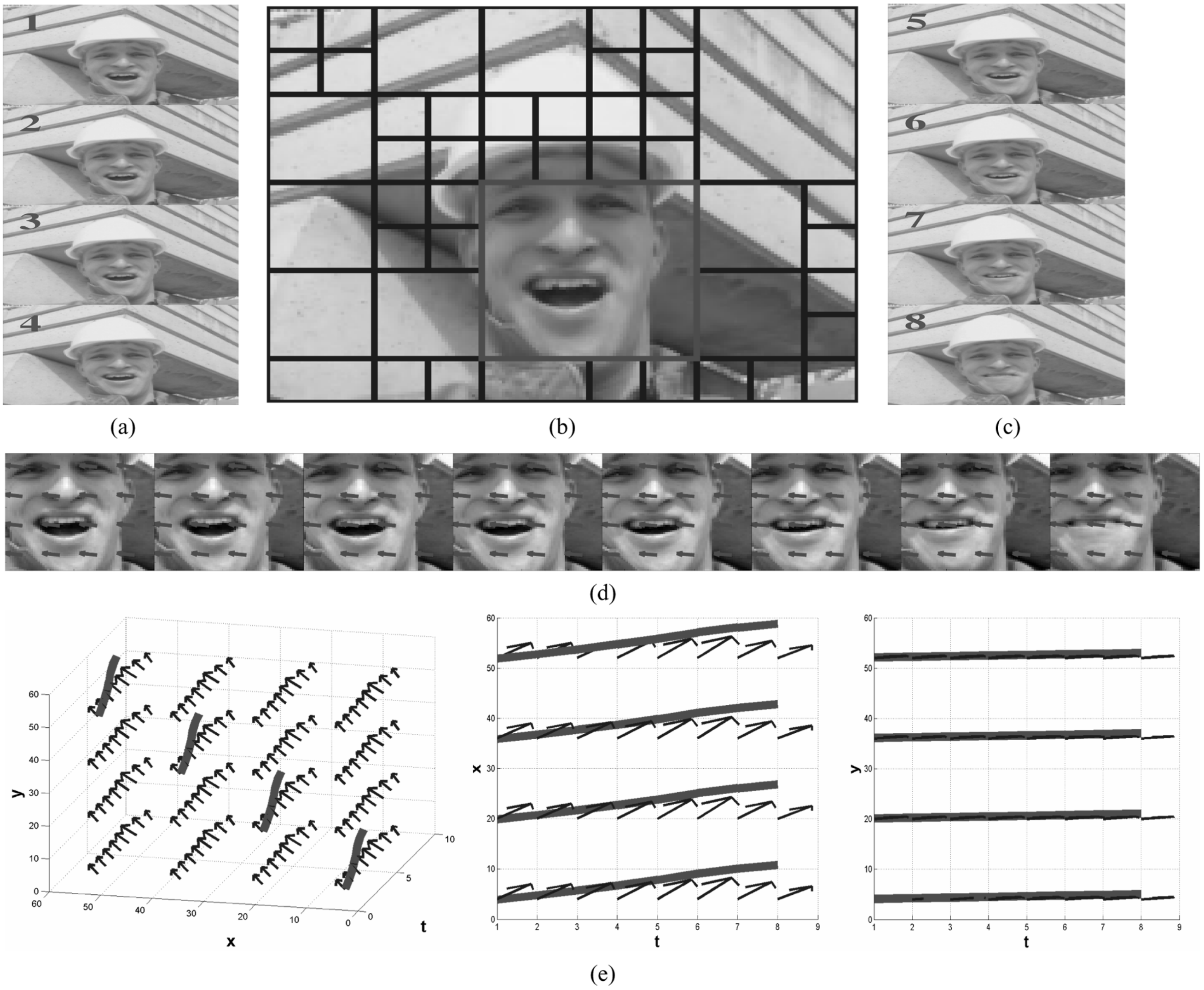
where $D_{i,q}$ is a child of the node D_i. The split-merge algorithm is applied until the top of the tree is reached, which concludes the optimal segmentation of the *gof* in terms of the bit rate and the reconstruction error. The 3-D SPIHT encoding allows the user to fix the bit rate to a certain number, so in our experiments, we dropped the λ term since R_i was the same for all children of a node. However, when other encoding techniques are to be used, this term should be incorporated. The basis for the whole *gof* is called the *block orthonormal bandelet basis*, and it consists of the union of the bases of the *subgofs* in the final segmentation, on their own supports.

V. Results

In this section, we show the results of the bandelet video compression on some standard video sequences, i.e., Akiyo, Alex, Foreman, and Mobile. All sequences are at QCIF resolution except for Alex, whose resolution is CIF. In sequences with low motion content (Alex and Akiyo), our results demonstrate the success of *SPREF* in improving the non motion compensated wavelet compression. In other ones where the motion is dominant, we compare the performance of *SPREF*-based compression with a motion-compensated wavelet compression technique, namely the *LIMAT* framework of Secker and Taubman in [15].

In our experiments, we used Daubechies 7–9 filters, and decomposed the data in two levels using the lifting scheme for both bandelets and wavelets. In the bandelet decomposition, the smallest *subgof* in the oct tree is $16 \times 16 \times 8$ $(x \times y \times t)$. The motion parameters are quantized with a step size of 1/8. When the frame size is not an integer multiple of the spatial size of our largest *subgof* (64×64), as in QCIF (176×144) frames, the oct-tree segmentation uses noncuboid *subgof*s near the image boundaries.

The directions of regularity in a video are usually not uniform. Our algorithm handles this nonuniformity by segmenting the *gof* into smaller regions. Fig. 3 shows the results of the *SPREF*-



(a) (b) (c)

(d)

(e)

Fig. 7. *SPREF*-based compression of frames 26–33 of the Foreman sequence at 500 kbps. (a), (c) Original frames; (b) optimal segmentation at this bit rate; (d) *SPREF* of the marked region superimposed on foreman's face; (e) 3-D view of the *SPREF* from oblique, top, and side views. Notice that the horizontal translational motion of the foreman's head is captured as seen in (d). Moreover, the top and side views shown in (e) also tell us that the motion is dominantly horizontal. Hence, decomposing the sequence along this direction is a better choice.

based compression of the frames 98–105 of the Alex sequence at 1000 kbps. Fig. 3(a) and (c) shows the original frames, and Fig. 3(b) shows the final segmentation at this bit rate. Notice that segmentation occurs when the spatial or temporal regularity cannot be modeled by a single *SPREF*, such as at the boundary of Alex's head and the background, and/or the right and left ends of the frame where the frame is padded with gray pixels. The *SPREF* of the region bounded by red rectangle in Fig. 3(c) is analyzed further in Fig. 3(d) and (e). In Fig. 3(d), the *SPREF* directions are superimposed on the *subgof*. From the motion of Alex's chin and her left ear, it can be seen that the x and y components of *SPREF* capture the motion of her head well. In Fig. 3(e), the *SPREF* is shown from various angles so that the motion components can be seen independently.

Fig. 4(a) shows a comparison of the *SPREF*-based compression and wavelet video compression at various bit-rates. The improvement as a result of the directional decomposition and bandeletization in *SPREF*-based compression can be clearly observed in this plot.

Fig. 5 shows the frames 11–18 of the Akiyo sequence compressed at 480 Kbps with our method. Following the format in Fig. 3, Fig. 5(a) and (c) shows the original frames, and Fig. 5(b) is the optimal segmentation at this bit rate. Notice the segmentation around Akiyo's head, which is moving in front of a static background. The background is also segmented in regions, where the spatial directions of regularity cannot be modeled by a single $x-t$ or $y-t$ *parallel SPREF*, such as the corners of the screens behind Akiyo. A more detailed analysis of the region marked with a red rectangle, and its *SPREF* is presented in Fig. 5(d) and (e). The motion of Akiyo's head in Fig. 5(d) can be observed from her brows, which can also be verified by the superimposed *SPREF* directions. In Fig. 5(e), we see this *SPREF* from oblique, top, and side views. Fig. 4(b) shows the bit rate versus PSNR plot of this sequence for both *SPREF*-based compression and standard wavelet video compression.

In Fig. 6, we show various outputs from the compression of the frames 99–106 of the Mobile sequence at 355kbps. The segmentation here is mainly determined by the motion of the calendar, the train and the background. Fig. 6(a) and (c) shows the original frames, and Fig. 6(b) shows the optimal segmentation at this bit rate. Notice that the motion boundaries are segmented into smaller regions. The *SPREF* of the marked region in Fig. 6(b) is analyzed further in Fig. 6(d) and (e). The motion in this region is purely horizontal, as can be seen from the superimposed directions in Fig. 6(d), and the top and side views of the *SPREF* in Fig. 6(e). Fig. 4(c) shows a comparison of compression between *SPREF*-based compression and the *LIMAT* framework at multiple bit rates. The motion in this sequence consists of many components such as global zooming out, rotating ball. In addition, the nonrigid motions of the objects, such as the swinging toy, can be modeled well with *SPREF* only if the group of frames is broken into much smaller spatiotemporal regions. The mesh model used in *LIMAT*, however, can model some of these nontranslational motion types better than *SPREF*. Hence, *LIMAT* performs marginally better than *SPREF* in this particular example.

We conclude this section with an example from the Foreman sequence. In Fig. 7, we see the frames 26–33 of the Foreman sequence compressed at 500kbps. In this clip, both the camera and the foreman's head are moving. However, since their motions are different, the regions that contain both are segmented into smaller regions. Following the same format in the previous figures, Fig. 7(a) and (c) shows the original frames, Fig. 7(b) shows the segmentation of the clip, and finally Fig. 7(d) and (e) shows the *SPREF* of the region marked with a red rectangle in Fig. 7(b). Notice that the global translational motion of the foreman's head in horizontal direction is captured by *SPREF*. This can be seen more clearly in Fig. 7(e), where the x and y components of the *SPREF* are shown in top and side views, respectively. Fig. 4(d) shows a comparison of the performance of *SPREF*-based compression with the *LIMAT* framework. The global motion in the Foreman sequence is mostly translational, which can be modeled very well by *SPREF* with a low motion overhead due to the spline representation. Moreover, bandeletization further takes advantage of this well-captured regularity. Hence, the improvement in PSNR in this sequence is significant.

VI. Conclusion

In this paper, we presented a new framework for video coding, using a special class of wavelets, the spatiotemporal bandelets. We approached the problem as that of finding the spatiotemporal directions, along which a group of frames is regular. Since the entropy of regular data is low, its decomposition yields higher compression rates. Previously, the studies on motion compensated wavelets have tried to improve the compression rates by only accounting for the temporal directions of regularities, and ignored the spatial regularities of the frames. In contrast, we presented a novel representation that models the directions of regularity that are determined by both *motion* and *spatial* structure of the scene. This distinguishes our work from the *MCWC* because *SPREF* allows us to exploit not only the temporal regularity but also the spatial regularity. We compared our work with standard wavelet video compression and *MCWC* where appropriate. We showed that as long as *SPREF* can model the motion well, it outperforms these methods. Our results imply that there is room for future work in *SPREF* such as modifying the model to handle nontranslational motions more accurately. We also want to explore the possibilities of computing good optical flow using this framework. We believe that the analysis of *SPREF* at different parallelisms holds the key to computing a good optical flow.

References

[1] [Online]. Available: http: //www.chiariglione.org/mpeg/standards/mpeg-1/mpeg-1.htm.

[2] [Online]. Available: http: //www.chiariglione.org/mpeg/standards/mpeg-2/mpeg-2.htm.

[3] [Online]. Available: http: //www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm.

[4] T. Acharya and P. S. Tsai, *JPEG2000 Standard for Image Compression*. New York: Wiley-Interscience, 2004.

[5] K. Ramchandran and M. Vetterli, "Wavelets, subband coding, and best bases," *Proc. IEEE*, vol. 84, no. 5, pp. 541–560, Apr. 1996.

[6] [Online]. Available: http: //www.jpeg.org/jpeg2000.

[7] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic.

[8] E. L. Pennec and S. Mallat, "Sparse geometric image representation with bandelets," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, Apr. 2005.

[9] G. Karlsson and M. Vetterli, "Three-dimensional subband coding of video," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Apr. 1988, pp. 1100–1103.

[10] D. Marpe and H. L. Cycon, "Very low bit-rate video coding using wavelet-based techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 2, pp. 85–94, Feb. 1999.

[11] J. Ohm, "Temporal domain sub band video coding with motion compensation," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Mar. 1992, vol. 3, pp. 229–232.

[12] D. Taubman and A. Zakhor, "Multirate 3d subband coding of video," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 572–588, Sep. 1995.

[13] S. Han and C. Podilchuk, "Video compression with dense motion fields," *IEEE Trans. Image Process.*, vol. 10, no. 11, pp. 1605–1612, Nov. 2001.

[14] J. Konrad and E. Dubois, "Bayesian estimation of vector fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 910–927, Sep. 1992.

[15] A. Secker and D. Taubman, "Highly scalable video compression with scalable motion coding," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1029–1041, Aug. 2004.

[16] J. E. Fowler and Y. Wang, "3d video coding using redundant wavelet multihypothesis and motion-compensated temporal filtering," presented at the IEEE Int. Conf. Image Processing, 2003.

[17] S. Cui, Y. Wang, and J. E. Fowler, "Multihypothesis motion compensation in the redundant wavelet domain," in *Proc. IEEE Int. Conf. Image Processing*, 2003, vol. 2, pp. 53–56.

[18] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, pp. 245–267, 1996.

[19] B. Kim, Z. Xiong, and W. W. Pearlman, "Low bit-rate scalable video coding with 3-d set partitioning in hierarchical trees (3-d spiht)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 1374–1387, Dec. 2000.

Orkun Alatas received the B.Sc. degree in electrical and electronics engineering from the Middle East Technical University, Ankara, Turkey, in 2000, and the M.S. degree in computer science from the University of Central Florida (UCF), Orlando, in 2003. He was pursuing the Ph.D. degree at the Computer Vision Laboratory, UCF.

His research topics included video compression, object modeling, and video in-painting.

Mr. Alatas passed away in September 2005.

Omar Javed (M'06) received the Ph.D. degree in computer science from the University of Central Florida, Orlando, in 2005.

He is a Research Scientist at the Center for Video Understanding, ObjectVideo, Reston, VA. His research interests include wide-area surveillance, tracking using a forest of sensors, video compression, multimedia content extraction, and semi-supervised classification.

Dr. Javed received the Hillman Fellowship for excellence in research in a Ph.D. program in 2001.

Mubarak Shah (F'03) is the Agere Chair Professor of Computer Science and the founding Director of the Computer Vision Laboratory, University of Central Florida (UCF), Orlando. He is a researcher in computer vision. He has published two books, ten book chapters, 55 papers in top journals, and 130 papers in refereed international conferences. He has worked in several areas, including activity and gesture recognition, violence detection, event ontology, object tracking (fixed camera, moving camera, multiple overlapping, and nonoverlapping cameras), video segmentation, story and scene segmentation, view morphing, ATR, wide-baseline matching, and video registration.

Dr. Shah was an IEEE Distinguished Visitor speaker for 1997 to 2000 and is often invited to present seminars, tutorials, and invited talks all over the world. He received the Harris Corporation Engineering Achievement Award in 1999; the IEEE Outstanding Engineering Educator Award in 1997; TOKTEN awards from the United Nations Development Program in 1995, 1997, and 2000; the Teaching Incentive award in 1995 and 2003; the Research Incentive Award in 2003; the Millionaires' Club award in 2005; the PEGASUS Professor award in 2006; an honorable mention for the ICCV 2005 "Where Am I?" Challenge Problem; and he was nominated for the best paper award in the ACM Multimedia 2005 Conference. He is an Editor of international book series on "Video Computing," the Editor-in-Chief of the *Machine Vision and Applications* journal, Area Editor of the *Wiley Encyclopedia of Computer Science and Engineering*, and an Associate Editor for the *Pattern Recognition* journal. He was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, and a Guest Editor of the Special Issue on Video Computing of the *International Journal of Computer Vision*.