Technical Report

Spatiotemporal Regularity Flow (SPREF): Its Estimation and Applications

Orkun Alatas, Pingkun Yan, Mubarak Shah*

Computer Vision Lab, (http://www.cs.ucf.edu/~vision/)
School of Electrical Engineering & Computer Science,
University of Central Florida, 4000 Central Florida Blvd., Orlando, FL 32816.
*E-mail: shah@cs.ucf.edu

December, 2006

Abstract

Feature selection and extraction is a key operation in video analysis for achieving higher level of abstraction. In this report, we introduce a general framework to extract a new spatiotemporal feature that represents the directions in which a video is regular, i.e. the pixel appearances change the least. Explicit modeling of these directions is very useful and desired by many applications. We propose to model the directions of regular variations with a 3D vector field, which is referred to as *SPatiotemporal REgularity Flow* (SPREF). SPREF vectors are designed to have three cross-sectional parallel components $\mathcal{F}_x$, $\mathcal{F}_y$, and $\mathcal{F}_t$ for convenient use in different applications. They are estimated using all the frames simultaneously by minimizing an energy functional formulated according to its definition. In this report, we first introduce translational SPREF (T-SPREF) and then extend our framework to affine SPREF (A-SPREF). Then we demonstrate the use of SPREF in a few applications, including object removal, video inpainting, and video compression. The promising experimental results prove the usefulness and versatility of this new framework.

1 Introduction

An important task of low level video analysis is to extract useful information from a video sequence. The purpose of the extraction is to convert the raw appearance values into meaningful features in order to achieve higher level of abstraction. The choice of features in this process depends on the nature of the problem at hand. In image and video processing, tasks such as motion analysis, compression, and video inpainting usually require extracting the spatiotemporal features of the data [1–3]. On the other hand, for other problems, such as key frame extraction, scene segmentation, and database queries, even a simple histogram may sufficiently represent the

data. Hence, the complexity of the features may range from simple color histograms, to eigenvalues and eigenvectors, optical flow vectors, wavelet coefficients, and so on, depending on the complexity of the problem.

The regularity direction of a video is an important feature that can be useful in many video processing applications. A video is determined to be regular along the directions, in which pixel intensities change the least [4]. These directions depend on both the type of the motion and the spatial structure of the scene. There is quite much previous work on spatiotemporal analysis of image sequences in video analysis [5–7]. A large body of those works focused on motion analysis in the spatiotemporal space. For instance, Heeger [8] proposed to estimate optical flow by using Gabor filter-based spatiotemporal energy models to deal with the aperture problem [9]. Later, Simoncelli and Adelson [10] revealed the equivalence of the filter-based spatiotemporal energy models and the gradient based methods [11,12] for computing optical flow under some conditions. They also proposed to compute probability distributions of optical flow using spatiotemporal filters. Adelson and Bergen [13] started another research direction by showing that the edges of objects moving in time create 3D surfaces. Many of the studies that followed used this fact, where the edge maps of the images were first computed, contours from them were extracted in each frame, and then the spatiotemporal surfaces that these contours swept were analyzed [14,15]. Allmen and Dyer [16] fitted quadratical patches to every point on the surface, and computed the optical flow through this parameterization. Peng and Medioni [17] took spatiotemporal slices of the data at each edge point, then searched for paths on these slices to find the direction of motion at that point. Baker and Colles [18] developed a *structure from motion* framework using the edges in a stereo image sequence. They fused the epipolar constraints with these points to compute the 3D structure of the scene. However, these point based methods are sensitive to noise and edges.

Due to the problems with edge detection and the increasing complexity of video sequences, the more recent studies started using spatiotemporal tensors for particular applications. Ngo et al. [19] used horizontal (xt) and vertical (yt) slices from the sequences to analyze the camera motion. They used spatial tensors to find the patterns on each slice, and then computed a histogram of these tensors to determine the motion type of the camera, i.e., pan/tilt, zoom, static, etc. Laptev and Lindeberg [20] computed the spatiotemporal tensors of the data to find the interesting points in time, where the spatiotemporal data changes significantly. They used this approach successfully in event detection and classification. Niyogi and Adelson [21] used horizontal spatiotemporal slices in gait detection. They searched for patterns of periodicity in these slices that can only be caused by moving persons. These applications have demonstrated the usefulness of spatiotemporal features in video processing. However, these spatiotemporal features as well as the extraction methods are specific for those applications and hence short of generality. It is difficult to apply them for other applications.

In this report, we propose a systematic approach for finding a new spatiotemporal feature, the local regular directions, along which a spatiotemporal region is regular, i.e. the pixel appearances vary the least. A good representation of the spatiotemporal regularity of a video can be useful in several areas. For example, object removal can be performed by removing the unwanted pixels along the regularity directions. Similarly, the target object can be easily tracked over a video if the regularity directions are known. In video inpainting, where the goal is of filling the spatiotemporal holes in a video without disturbing its temporal regularity, explicit modeling of this regularity allows us to inpaint the videos conveniently. Regularity flow can also be used to

improve the wavelet video compression [3, 22, 23]. If a video is compressed along the directions, in which its entropy is minimum, the rate of compression increases. As we will present later, since the overhead of storing the regularity flow is smaller than the gain in compression, it is quite suitable for this application.

The proposed approach for regularity flow estimation does not rely on edge detection, hence its success does not depend on the presence of strong edges in the scene. Instead it analyzes the whole region, and tries to find the best directions that model the overall regularity of the region. Even when the local gradient of a pixel is not significant, the global analysis of the region assigns a well-defined direction to it. In our work, the directions of regularity are modeled with a 3D vector field, called the *SPatiotemporal REgularity Flow* (SPREF) field. The strength of SPREF lies in treating the data not as a sequence of 2D images but as a 3D volume, and processing all of its information simultaneously. SPREF is designed to have three cross-sectional parallel components in order to handle the regularities that depend on the motion and the scene structure, which provides much flexibility to the applications.

In this report, we first introduce the translational SPREF (T-SPREF) with much simplified computation. T-SPREF gives good estimation results when the directions of regularity of the spatiotemporal region is a function of the flow propagation axis. In other words, when the motion is translational, or when all the edges in the scene extend along the same direction in the absence of motion, T-SPREF performs well. However, the precision of the translational flow model goes down when the directions of regularity depend on multiple axes. This is the case, for example, when the motion is zooming in/out or rotation, where the true flow is not only a function of time, but also a function of spatial location. Similarly, when two edges extend in different directions along the x or y axes, T-SPREF cannot find the correct directions of spatial regularity. In order to deal with such cases, we introduce the affine SPREF (A-SPREF) model. The components of A-SPREF still propagate along one major axis, respectively. However, each component is also a function of the other axes.

The organization of the rest of the report is as follows. We present the framework and mathematical formulation of the SPREF in Section 2. The two types of SPREF, T-SPREF and A-SPREF, are proposed. In Section 3, the relationship between SPREF and optical flow is discussed. Next in Section 4, we demonstrate the applications where SPREF can be used, object removal, video inpainting and video compression. We finally conclude with a discussion of the SPREF framework in Section 5.

2 The Spatiotemporal Regularity Flow (SPREF)

SPREF ($\mathcal{F}$) is a $3D$ vector field that shows the directions, along which an intensity I in a spatiotemporal region Ω is regular, i.e., the pixel intensities in the region change the least. The condition that the intensity should vary regularly in the flow direction can also be perceived as a requirement to follow the directions, in which the sum of directional gradients is minimum. This allows us to write the general flow energy function, for $\mathcal{F}$ as

$$E(\mathcal{F}) = \int_{\Omega} \left| \frac{\partial (I \star H)(x,y,t)}{\partial \mathcal{F}(x,y,t)} \right|^2 dxdydt, \tag{1}$$

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

The particular definition of SPREF $\mathcal{F}$ depends on the flow model that is used. In this section, we will introduce two types of SPREFs based on two different flow models: translational (T-SPREF) and affine (A-SPREF). In the T-SPREF model, we choose one of the main coordinate axes (x, y or t) to be the axis of flow propagation for simplicity. The magnitude of the flow component along the propagation axis is taken as 1. The magnitudes of the remaining components are determined by minimizing the flow energy function (1) according to the flow models, which is only relevant to the propagation axis. Thus, the components of the SPREF along each propagation direction are translational. The A-SPREF can be considered as a general extension of the T-SPREF model, which also propagates along one major axis. However, each component of the A-SPREF is a function of other axes as well. Therefore, the affine motion and/or complex structure can be captured.

2.1 Translational (T-) SPREF

In the T-SPREF model, the flows are approximated by block translations orthogonal to the directions of flow propagation. This results in *planar (cross-sectional) parallelism* in the SPREF, which is defined as all the vectors on a plane being uniform (equal in magnitude and direction). In our framework, a *cross-sectional parallel* flow field consists of the following three components: *xy-parallel* ($\mathcal{F}_t$), *xt-parallel* ($\mathcal{F}_y$), and *yt-parallel* ($\mathcal{F}_x$). In an *xy-parallel* flow, the vectors on the xy plane of the flow field for a particular t are cross-sectional parallel. The planar parallelisms are similarly defined for the xt and *yt-parallelism*, where the flow propagation axes are x and y respectively. Modeling of SPREF using three cross-sectional parallel components is motivated by the requirement of different applications. For example, in video compression, wavelet basis can be warped along the flow directions to exploit the spatiotemporal redundancy in the video. Depending on the video, this parallelism can be *xy-parallel*, *xt-parallel* or *yt-parallel*. Parallelism is required to force the warped wavelet basis to be orthogonal. In addition, having three separate components provides more flexibility to the scheme. The physical meaning of each component can be easily exploited. For example, a moving object can be efficiently removed from a video by using only the *xy-parallel* flow, which describes the motion regularity of the video. Similarly in video inpainting, if the missing part undergoes global motion, the spatiotemporal hole can be completed by using only the *xy-parallel* flow, which can greatly simplify the inpainting process.

Since the *xy-parallel* flow propagates in the temporal axis, it models the regularity that depends on the motion in a spatiotemporal region Ω. The other two flow types, *xt-parallel* and *yt-parallel*, can model the temporal regularity to some extent but they can also model the spatial regularity of Ω when there is no motion. If the motion in Ω is global, its directions of regularity can be modeled by a single *xy-parallel* SPREF. However, if there are different layers of motion, then Ω needs to be segmented into smaller spatiotemporal regions (Ω_i) until each region's regularity can be captured by a single SPREF. A similar segmentation is necessary for the static scenes, where the scene regularity is too complex to be captured by a single xt or *yt-parallel* SPREF. To do this, the video is first divided into *group(s) of frames* (gof). Then we partition each gof into smaller *subgroups of frames* ($subgof$) using an oct-tree. This segmentation allows us to analyze the regularity of the gof at multiple locations and various sizes. This step may or may not be followed by merging the $subgof$s depending on the application. The quality of each SPREF is determined by a metric, specific for the goal of the application.

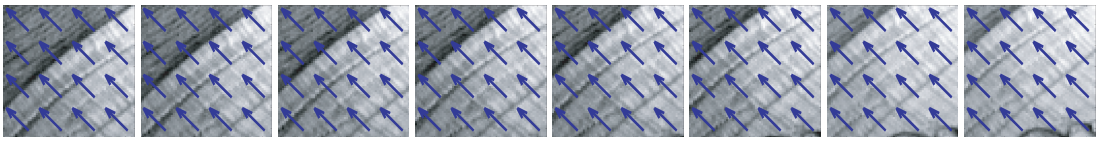
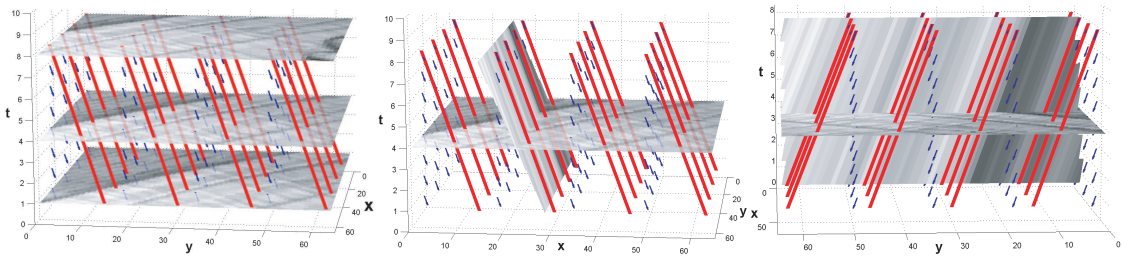


Figure 1: A synthetic sequence, where the global translational motion (u,v) is (-2,-2). The *xy-parallel* SPREF directions (blue arrows) are projected on each frame.



(a) (b) (c)

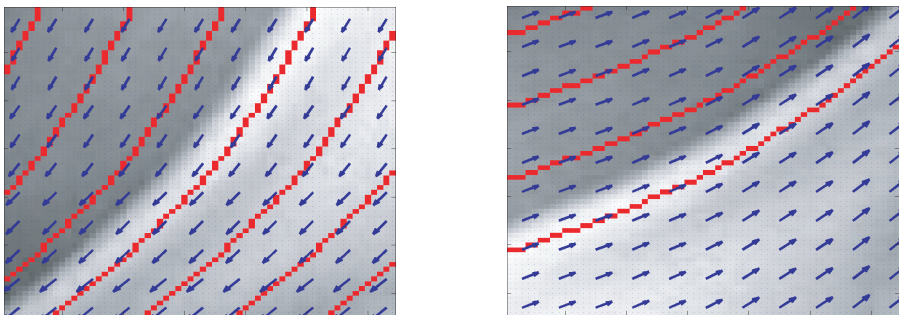
Figure 2: (a) The 1st, 4th, and 8th frames of the sequence in Fig. 1 are shown in 3D. SPREF is subsampled and its directions are shown with blue arrows. Along the SPREF curves, shown in red, the appearance varies the least. (b) Side view of the cross-section of the sequence along the flow curves. Notice that the flow curves extend along the direction of motion. (c) The front view of the cross-sectional surface. Notice that the pixel appearances on the surface change the least along the SPREF curves. Since the SPREF estimate is precise for this sequence, the pixel values along the flow curves are actually constant.

All the three components of SPREFs can be formulated by discretizing the continuous flow energy function (1), and tailoring it according to how $\mathcal{F}$ is defined. If the flow is *xy-parallel*, then $\mathcal{F}$ is defined as $\mathcal{F}_t = (c_1'(t), c_2'(t), 1)$, which results in

$$E(\mathcal{F}_t) = \sum_{\Omega} \left| \left(I \star \frac{\partial H}{\partial x} \right) c_1'(t) + \left(I \star \frac{\partial H}{\partial y} \right) c_2'(t) + I \star \frac{\partial H}{\partial t} \right|^2 . \tag{2}$$

Notice that the formulation of $\mathcal{F}_t$ implies that the x and y components of the flow, $(c_1'(t), c_2'(t))$, are functions of time only, which are constant for all the pixels in a given frame, i.e., xy cross-section of Ω. Fig. 1 shows a synthetic clip where the motion in all frames is a translation by 2 pixels in both up and left directions. Fig. 2(a) shows the the 1st, 4th, and 8th frames, and the estimated 3D SPREF field (blue arrows). The red curves are the curves of regular appearance, which will be described later. Fig. 2(b) and (c) show a cross-section of the data along the curve directions. On this cross-sectional surface, the pixel intensities vary the least along the flow directions (in fact, they are constant in this example since the SPREF estimate is precise).

If the flow is *yt-parallel*, then $\mathcal{F}$ is formulated as $\mathcal{F}_x = (1, c_2'(x), c_3'(x))$. As the definition of *yt-parallelism* implies, the flow vector, $(c_2'(x), c_3'(x))$, is the same on a yt cross section at a given



(a) (b)

Figure 3: The first frames of two sequences with no motion, whose SPREFs are xt (a) and yt (b) parallel. Since the sample sequences are static, the temporal component of the flow is zero, and SPREF shows the spatial directions of regularity of the frames (blue arrows) . The flow curves, shown in red, follow these directions. Notice that, due to cross-sectional parallelism, the flow directions of xt (yt) parallel SPREF are uniform for a given column (row) in all frames.

x. This definition results in the following flow energy function:

$$E(\mathcal{F}_x) = \sum_{\Omega} \left| I \star \frac{\partial H}{\partial x} + \left(I \star \frac{\partial H}{\partial y} \right) c_2'(x) + \left(I \star \frac{\partial H}{\partial t} \right) c_3'(x) \right|^2. \tag{3}$$

Fig. 3(b) shows $\mathcal{F}_x$ for a static sequence, where the flow directions (red arrows) capture the directions of regularity of the frames, and the curves show the paths of regular variation.

For the *xt-parallel* flow, $\mathcal{F}$ is specified as $\mathcal{F}_y = (c_1'(y), 1, c_3'(y))$, which means that the flow directions in the xt cross section of the SPREF, $(c_1'(y), c_3'(y))$, are uniform. The expansion of general equation (1) with this definition results in the following:

$$E(\mathcal{F}_y) = \sum_{\Omega} \left| \left(I \star \frac{\partial H}{\partial x} \right) c_1'(y) + I \star \frac{\partial H}{\partial y} + \left(I \star \frac{\partial H}{\partial t} \right) c_3'(y) \right|^2. \tag{4}$$

Fig. 3(a) shows another static sequence, where the directions of regularity of the scene are captured by $\mathcal{F}_y$. Similar arguments for the flow directions and the curves apply here.

The equations (2), (3) and (4) can directly be solved for the flow directions, $c_m'[u]$, $(m = 1, 2, 3)$, using spatio-temporal derivatives: $I \star \frac{\partial H}{\partial x}$, $I \star \frac{\partial H}{\partial y}$, $I \star \frac{\partial H}{\partial t}$. However, we approximate the flow directions with splines and solve these equations for the spline parameters. Since spline can be used to represent the flow directions in several frames, fewer parameters are needed with splines. Moreover, since splines are smooth, we obtain a better approximation of flow directions, which is not sensitive to noisy flow direction in a particular frame. We use the translated box spline functions of the first degree, $b(u)$, for the approximation of the flow directions, which result in:

$$c_m'(u) = \sum_n \alpha_n^m b(2^{-l}u - n) = \sum_n \alpha_n^m b_n^l(u), \tag{5}$$

where α_n ($n = 1, \ldots, 2^l$) is the nth spline coefficient, u is the index of the SPREF component ($u = x$, y or t), $l = 1, \ldots, k$ is a scale factor, and 2^k is the width of Ω in the axis of parallelism. The spline function used in our experiments is defined as

$$b(z) = \begin{cases} 1 - |z|, & \text{if } |z| < 1, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

The coefficients α_n can be solved by quadratic minimization of the energy functions (2), (3) or (4), the choice depending on the parallelism type. The open form of the solution is given in Appendix A.

Knowing only the directions of the regularity of Ω does not provide sufficient information most of the time. In many applications we need to know the actual 3D curves, on which Ω is regular. A *SPREF curve*, $c[u]$, is an integral curve, whose tangents are parallel to the SPREF directions. The coordinates of a SPREF curve in the discrete domain are computed according to the equation,

$$c(u) = \sum_{z=1}^{u} c'(z), \quad u \in \{x, y, t\}. \tag{7}$$

The orthogonality of the flow directions in our SPREF model results in one-to-one mapping of all points in the spatiotemporal region along the flow curves. Moreover, due to this attribute if a spatiotemporal region has a mixture of points with weak and strong gradients, then the directions of regularity of the weak-gradient points can still be defined based on the ones with strong gradients. The SPREF curves for different types are shown in red in Figs. 2 and 3. The curve for an *xy-parallel* SPREF is the set of points $(x + c_1(t), y + c_2(t), t)$ for a constant (x, y) and a varying t. For an *xt-parallel* SPREF, the curve coordinates are $(x + c_1(y), y, t + c_3(y))$ for constant (x, t) and varying y. Finally, for a *yt-parallel* SPREF, the set of curve points is $(x, y + c_2(x), t + c_3(x))$ for constant (y, t) and varying x. All curve coordinates are defined only inside the support of Ω.

2.2 Affine (A-) SPREF

The T-SPREF gives good results when the directions of regularity of the spatiotemporal region is a function of the flow propagation axis. In other words, when the motion is translational, or in the absence of motion when all the edges in the scene extend along the same direction, T-SPREF performs well. However, the precision of the translational flow model goes down when the directions of regularity depend on multiple axes. This is the case, for example, when the motion is zooming in/out or rotation, where the true flow is not only a function of time, but also a function of spatial location. Similarly, in the absence of motion, when two edges extend in different directions along the x or y axes, T-SPREF cannot find the correct directions of regularity. For such cases, we change the flow model from translational to affine, where the flow still propagates along one major axis, however, it is a function of all the axes.

Since the flow vector field $\mathcal{F}$ is defined according to the affine model, the general flow energy function (1) is expanded accordingly. When the propagation axis is t, $\mathcal{F}$ is defined as $\mathcal{F}_t = (c'_1(x, y, t), c'_2(x, y, t), 1)$, and formulated by

$$E(\mathcal{F}) = \sum_{\Omega} \left| \left(F \star \frac{\partial H}{\partial x} \right) c'_1(x, y, t) + \left(F \star \frac{\partial H}{\partial y} \right) c'_2(x, y, t) + F \star \frac{\partial H}{\partial t} \right|^2, \tag{8}$$

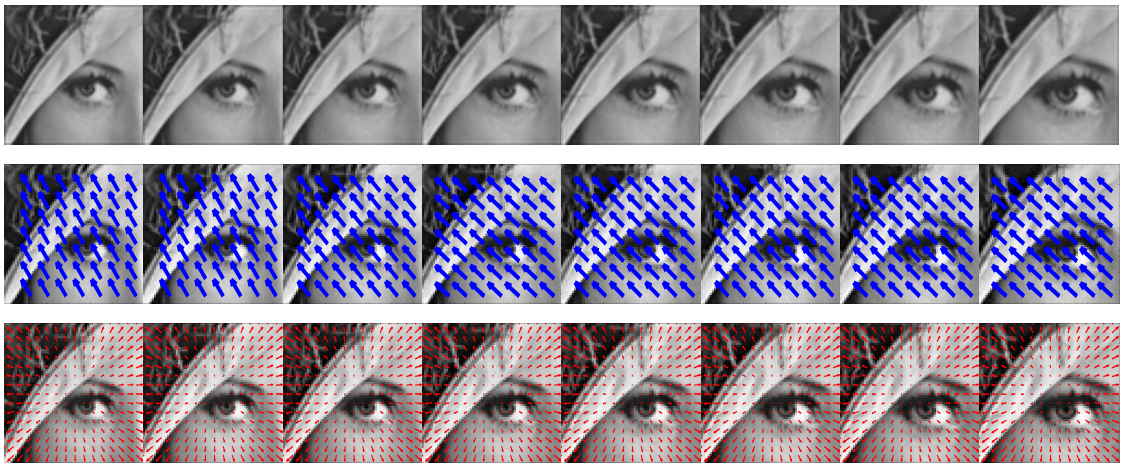


Figure 4: (a) A synthetic sequence from the Lena image where Lena's eye is zoomed in successive frames. (b) T-SPREF approximation to the underlying directions of regularity, shown with blue flow vectors superimposed on the images. Since zooming in is approximated by translations, the approximation is not successful. (c) A-SPREF approximation of the directions of regularity. The flow vectors are clearly more precise than T-SPREF.

where

$$\begin{bmatrix} c_1'(x,y,t) \\ c_2'(x,y,t) \end{bmatrix} = \begin{bmatrix} a_{11}(t) & a_{12}(t) & a_{13}(t) \\ a_{21}(t) & a_{22}(t) & a_{23}(t) \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{9}$$

Just like in T-SPREF, the flow parameters $a_{ij}(t)$ can be obtained by directly solving the flow energy function (8). However, since we want to achieve a global solution that uses all the information in the spatiotemporal region Ω, we approximate these parameters by splines. Hence, $a_{ij}(t)$ is expanded as:

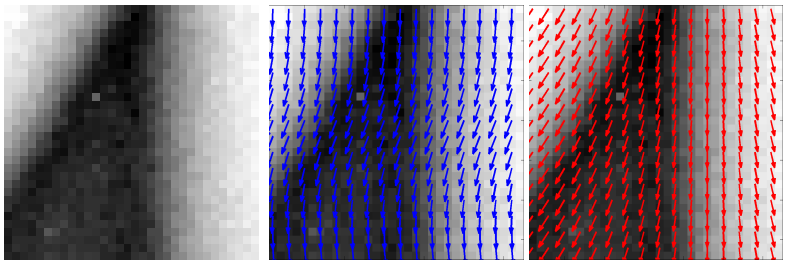
$$a_{ij}(t) = \sum_n \alpha_n^{ij} b(2^{-l}t - n) = \sum_n \alpha_n^{ij} b_n^l(t). \tag{10}$$

The solution of (8) for the spline parameters is given in Appendix B.

Fig. 4 (a) shows a synthetic sequence generated from the Lena image, where her eye is zoomed in successive frames. The directions of regularity obtained by an *xy-parallel* T-SPREF is shown in Fig. 4(b), where it can clearly be seen that the translational approximation cannot estimate the underlying motion. On the other hand, since A-SPREF can handle this type of motion, the estimated flow vectors in Fig. 4(c) reveal the true directions of regularity for the image sequence.

When the flow propagation axis is x or y, the flow components $\mathcal{F}_x$ and $\mathcal{F}_y$ are computed, respectively. For the former case, we have $\mathcal{F}_x = (1, c_2'(x,y,t), c_3'(x,y,t))$ and

$$\begin{bmatrix} c_2'(x,y,t) \\ c_3'(x,y,t) \end{bmatrix} = \begin{bmatrix} a_{21}(x) & a_{22}(x) & a_{23}(x) \\ a_{31}(x) & a_{32}(x) & a_{33}(x) \end{bmatrix} \begin{bmatrix} t \\ y \\ 1 \end{bmatrix}. \tag{11}$$



(a)

(b)

Figure 5: (a) (Left) The first frame of a static sequence, sampled from another location of the Lena image. Notice that there are two different edge orientations (vertical and oblique) in the sample. (Middle) The *xt-parallel* T-SPREF approximation of the directions of regularity superimposed on the first frame of the sequence. Notice that the oblique orientation dominates the final estimate, resulting in a poor estimate for the vertical orientation. (Right) The A-SPREF approximation of the directions of regularity, where the flow propagates along y axis. With the affine model, the vectors estimate both orientations correctly. (b) (Left) An eagle image, from which we used a small patch (bounded with a red box) to create another static sequence. Notice that the orientation of each feather is different. (Middle) The *yt-parallel* T-SPREF approximation of the directions of regularity superimposed on the first frame. The translational model chooses to fit all the directions to the middle feather, since its orientation is halfway between the top and the bottom feathers. (Right) The A-SPREF approximation of the directions of regularity, where the flow propagates along x axis. Here it can be clearly seen that the vectors have been adjusted to fit all orientations separately.

Similarly, we have $\mathcal{F}_y = (c_1'(x,y,t), 1, c_3'(x,y,t))$ and the parameters are

$$\begin{bmatrix} c_1'(x,y,t) \\ c_3'(x,y,t) \end{bmatrix} = \begin{bmatrix} a_{11}(y) & a_{12}(y) & a_{13}(y) \\ a_{31}(y) & a_{32}(y) & a_{33}(y) \end{bmatrix} \begin{bmatrix} x \\ t \\ 1 \end{bmatrix}. \tag{12}$$

Fig. 5(a) shows the first frame of a static sequence from the Lena image. In this sequence there are two different edge orientations. With the flow propagating along the y axis, *xt-parallel* T-SPREF tries to find the optimal orientation that can minimize the flow energy (middle image). A-SPREF on the other hand can handle multiple orientations (right image). Hence, its estimation is more precise than that of the T-SPREF. In Fig. 5(b), the first image shows an eagle, where we sampled

a patch (marked with a red bounding box) to create another static sequence. With the flow propagation axis chosen as x, the middle image shows the *yt-parallel* T-SPREF approximation of the directions of regularity. Notice that there are three different orientations of the feathers in the patch. T-SPREF tries to find the optimal direction that can approximate all the three simultaneously. A-SPREF, on the other hand, can approximate the three directions separately. Therefore, the affine model can estimate the local geometry of the spatiotemporal region much better, as shown in the last image in this figure.

After finding the flow directions, the next step in A-SPREF is computation of the *flow curves*. Different from T-SPREF, here the curves are computed by propagating the affine parameters along the propagation axis. Since the flow model is not translational, summing up the parameters is not an option. Let's assume that the axis of propagation is t. The flow $(c_1'(x,y,t), c_2'(x,y,t), 1)$ only maps the pixels in frames t to $t+1$. To compute the flow curves, however, one needs to map the pixels in one frame to all the others, with a new set of parameters. Given two sets of affine parameters estimated according to (10): $A_{t\to t+1} = \{a_1, a_2, \ldots, a_6\}_{t\to t+1}$ and $A_{t+1\to t+2} = \{b_1, b_2, \ldots, b_6\}_{t+1\to t+2}$, the *propagate operation* $\mathcal{G}$ produces the new parameter set $\widehat{A}_{t\to t+2} = \{\widehat{a}_1, \widehat{a}_2, \ldots, \widehat{a}_6\}_{t\to t+2} = \mathcal{G}(A_{t\to t+1}, A_{t+1\to t+2})$ as follows:

$$\begin{aligned}
&\widehat{a}_1 = a_1 + b_1 * (1 + a_1) + b_2 * a_4, \quad \widehat{a}_2 = a_2 + b_2 * (1 + a_5) + b_1 * a_2,\\
&\widehat{a}_3 = b_3 + a_3 * (1 + b_1) + b_2 * a_6, \quad \widehat{a}_4 = a_4 + b_4 * (1 + a_1) + b_5 * a_4,\\
&\widehat{a}_5 = a_5 + b_5 * (1 + a_5) + b_4 * a_2, \quad \widehat{a}_6 = b_6 + a_6 * (1 + b_5) + b_4 * a_3.
\end{aligned} \tag{13}$$

With the propagate operation $\mathcal{G}$ in mind, the new parameters that will be used to compute the flow curves are written as follows:

$$\widehat{A}_{0\to t+1} = \mathcal{G}\left(\widehat{A}_{0\to t}, A_{t\to t+1}\right), \tag{14}$$

where $\widehat{A}_{0\to 1} = A_{0\to 1}$.

After the new set of parameters are computed, the flow curve coordinates that they imply are stored in coordinate grids, which are given as $(\widehat{a}_{11}(t)x + \widehat{a}_{12}(t)y + \widehat{a}_{13}(t),\ \widehat{a}_{21}(t)x + \widehat{a}_{22}(t)y + \widehat{a}_{23}(t))$ for the SPREF propagating along axis t. When the flow propagation axis is x or y, the same algorithm can be applied after doing the necessary change of variables. In fact, Fig. 6 shows these types of curves on static sequences that we showed in Fig. 5. The images on the left of Fig. 6 show the T-SPREF curves, while the ones on the right side show the A-SPREF curves. Notice that these images are challenging for T-SPREF to approximate, however A-SPREF can nicely produce curves that follow all the edges in the scene.

2.3 Modeling Nonuniform Regularities

Extending the SPREF framework to model the whole video is a must in many video applications. SPREF is designed to compute the local directions of regularity of a spatiotemporal region, and the whole video can be considered as a local region only when it undergoes global motion, or when there is no motion and the spatial structure of the scene is simple. However, usually this is not the case; the videos often are mixtures of regions with both local and global motion. Also, the scenes are usually highly textured, which hurts the *xt* and *yt-parallel* SPREF approximations. In such

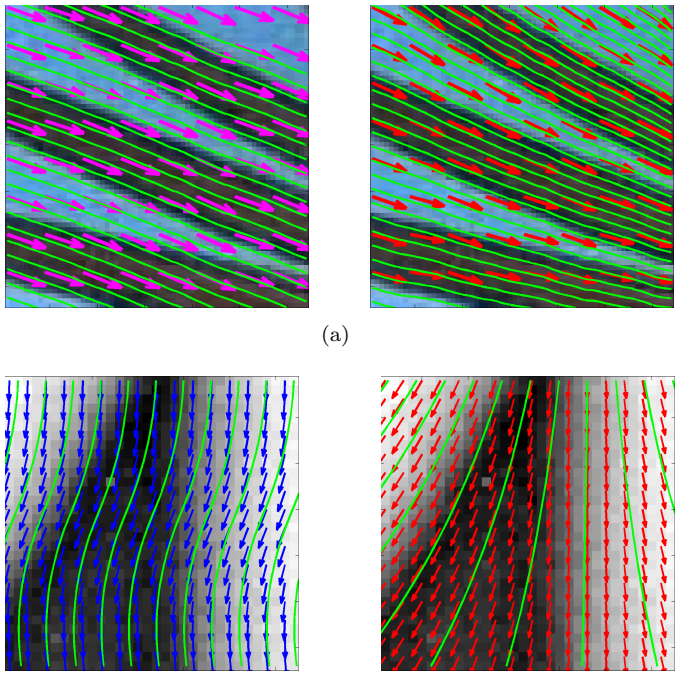


Figure 6: (a) (Left) The T-SPREF curves superimposed on the first frame of the clip in Fig. 5(a). Notice that the curve can neither follow the oblique edge, nor the vertical one. The solution is an approximate curve that extends between the two edges. (Right) The A-SPREF curve for the same image. This time the orientation of both edges are estimated correctly and the curves extend along both. (b) (Left) The T-SPREF curves for the second clip in Fig. 5(b). Notice that the curve extends nicely along the middle feather, however this curve clearly is not the best for the top and the bottom ones. (Right) The A-SPREF curves for the same image. This time the curves extend along all edges, which is most visible when one looks at the top feather.

cases, the video needs to be segmented into smaller spatiotemporal regions until the regularity of each region is uniform. Then the SPREF can be computed. To do this, the video is first divided into *group(s) of frames* (gof) and then each gof is partitioned into smaller *subgroups of frames* ($subgof$) using an oct-tree, as shown in Fig. 7 This segmentation allows us to analyze the regularity of the gof at multiple locations and various sizes. This step may or may not be followed by merging the $subgof$s depending on the application. The quality of each SPREF is determined by a metric, specific for the goal of the application. For instance, the metric can be the flow error for inpainting applications, or the total bit cost for compression applications.

With the defined metric, a gof can be divided into $subgof$s to minimize the metric, such that the directions of regularity of each $subgof$ is estimated as closely as possible by its corresponding SPREF. The challenge here is to find the optimal segmentation of the gof's support, V, into V_is, ($V = \bigcup_i V_i$), so that the overall cost is minimized. A general metric is used here to illustrate the

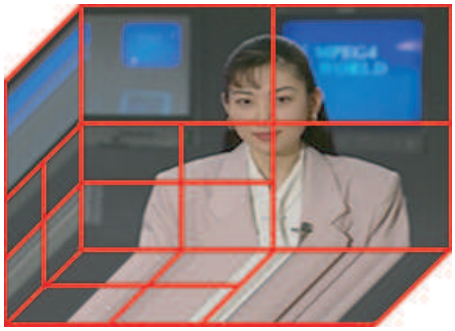


Figure 7: A sample oct-tree segmentation of a group of frames (gof). The segments can be divided recursively to achieve the minimum overall cost.

segmentation process, which is defined as

$$D + \lambda N = \sum_i D_i + \lambda N_i, \tag{15}$$

where

$$D_i = \sum_x \sum_y \sum_t (I_{i,original}(x,y,t) - I_{i,reconstructed}(x,y,t))^2, \tag{16}$$

is the sum of squared reconstruction error of I_i. N_i denotes the number of the SPREF parameters, and λ is a Lagrange multiplier. The objective here is to find a segmentation to minimize the cost in (15) by dividing a gof into minimal number of $subgof$s where the regularity can be represented using uniform SPREF.

In order to achieve this segmentation, we initially partition the gof into rectangular prisms (cuboids) using an *oct-tree* data structure, as shown in Fig. 7. The width of each dimension of a cuboid is 2^{k_j}, where $j \in \{1,2,3\}$ denotes the particular dimension and k denotes the level. The number of SPREF coefficients (N_i) of a $subgof$ also indicates whether a certain type of spatiotemporal regularity flow exists or not. $N_i = 0$ means that there is no flow defined for the $subgof$. This can be the case, where the sequence is isotropic, i.e. all the pixels have the same value, or the pixel intensities in the frames are totally random. The minimization of the total cost in Eqn. (15) starts with computing the cost, $(D_i + \lambda N_i)$, for each cuboid in the oct-tree, which can only be minimum for one of the four flow hypotheses, including the no-flow case. When computing the cost of the SPREFs for a certain parallelism, the optimal scale parameter l $(1 \leqslant l \leq k)$ in Eqn. (5) is found by trying all possible values of l, and selecting the one that results in the smallest cost.

The optimal segmentation of V is found by using 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 cost of the parent, otherwise they stay split. This merging constraint can be formulated as:

$$D_i + \lambda N_i < \sum_{q=1}^{8} D_{i,q} + \lambda N_{i,q}. \tag{17}$$

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 number of SPREF coefficients and the reconstruction error.

3 SPREF vs Optical Flow

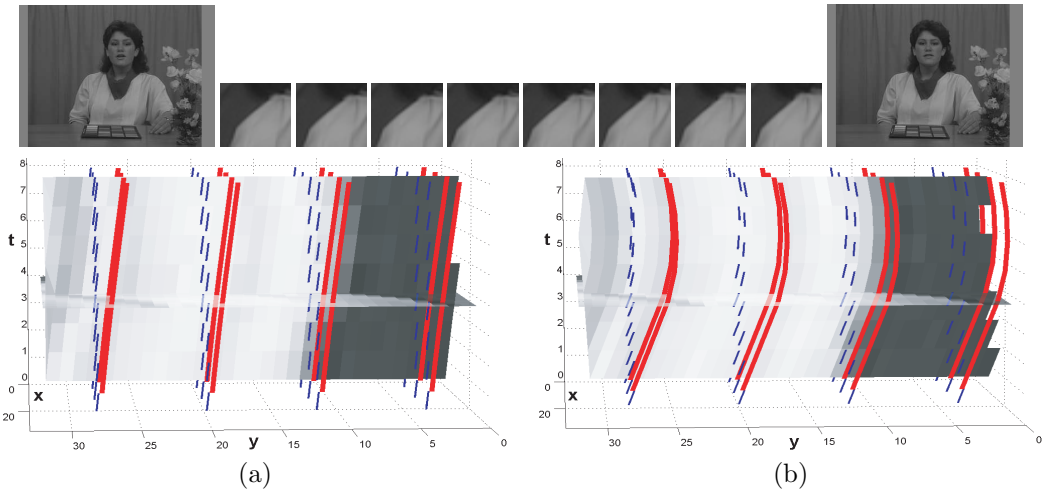
The fundamental difference between SPREF and optical flow is that the SPREF captures both the spatial and temporal regularity information simultaneously, while optical flow only cares about motion information in the temporal direction. When motion exists in the video, the SPREF along the t direction, i.e. *xy-parallel* SPREF, is similar to the optical flow. However, it carries similar but not necessarily the same information as the optical flow. The true optical flow $(u(x,y), v(x,y))$ yields the directions of highest regularity between two frames as a function of the pixels' spatial locations. Its estimation requires minimizing the *brightness constancy equation*,

$$f_x u + f_y v + ft = 0. \tag{18}$$

Hence, it is not well-defined if computed over regions where the spatiotemporal gradients are insignificant. The *xy-parallel* SPREF equation (2) also has the same structure as the brightness constancy equation (18), however, our spline-based formulation allows us to minimize it over multiple frames not pairwise but simultaneously. Splines were previously used in motion estimation by Szeliski and Coughlan [24], where they *spatially* fitted a $2D$ spline to the optical flow. They also claimed that this approach could be extended to multiple frames if the motion is linear. The *xy-parallel* SPREF differs from this work in that it fits a spline to the motion temporally in the whole volume, which makes it a more powerful over multiple frames.

SPREF assigns all pixels a certain direction of regularity even when only some of them have significant spatiotemporal gradients. The optical flow, on the other hand, usually lacks orthogonality. Hence, it cannot be used to construct flow curves that are guaranteed to go through all pixels. Even though some optical flow estimation method can be extended to using multiple frames, the estimated optical flow remains to be pairwise. That is, no matter how many frames are used, they only estimate the optical flow between neighboring frames. On the other hand, the SPREF curve gives the global regularity direction of the whole volume.

For the *xy-parallel* T-SPREF, we formulate u and v, i.e. c_1' and c_2', as functions of time only, which models the motion as a block in each frame, and results in an orthogonal flow field. If the motion in a spatiotemporal region is globally translational, then *xy-parallel* SPREF converges to the optical flow. Otherwise, it tries to estimate the underlying motion as close as possible. If the estimation is poor, then data can be segmented as explained in the previous section, and a closer estimate of motion can be achieved. A comparison of the two models is given in Fig. 8. The top row shows a mini clip sampled from the Alex sequence. The directions of regularity for this clip are computed by block motion and SPREF, separately. The bottom row shows the flow curves computed using (a) block motion, and (b) SPREF. The spatiotemporal data is sliced along the flow curves as in Fig. 2.(c). Notice that the pixels along the SPREF curves are smoother due to incorporation of all frames in the solution.



(a) (b)

Figure 8: (Top) A mini clip from the Alex sequence. (Bottom) The directions of regularity obtained by (a) block motion and (b) *xy-parallel* SPREF. Notice that when the sequence is sliced along the directions of regularity, the resulting surface is smoother along the flow curves when SPREF is used.

4 Applications

In this section, the applications of SPREF in object removal, video inpainting, and video compression are demonstrated. The results obtained using SPREF are also compared with other state-of-the-art approaches for performance evaluation. Note that SPREF is a general framework. Here we just demonstrate a few example applications. The use of SPREF is certainly not limited to the applications shown here.

4.1 Object Removal

Object removal is to remove a target object from the video [25]. In many video applications, this is one of the key techniques for video processing. For example, some common digital special effects are composition of graphic objects, removal of unnecessary objects, and insertion of virtual objects in a video sequence. Unfortunately, manual selection of the object from each frame is normally required to remove the object. The procedure is labor intensive and therefore time consuming. However, by using the SPREF, the amount of manual work can be significantly decreased.

Our SPREF based object removal algorithm is presented as Algorithm 1. The basic idea is to remove the moving object by following the *xy-parallel* SPREF curve. In our approach, the manual selection is only required for the first and the last frames of the GOF. The object in the other frames can then be removed automatically by using the *xy-parallel* SPREF.

Fig. 9 shows an example of object removal using the SPREF based method. The objective there is to remove the airplane from the video. SPREF was computed from the original video

Figure 9: An example of object removal. The first row shows the original video frames containing an airplane. In the first and the last frames, the airplane is manually removed. It is then automatically removed from other frames according to the SPREF computed from the original sequence.

Algorithm 1 Object Removal

Input: A group of frames (GOF)
Compute the *xy-Parallel* SPREF $\mathcal{F}_t$.
Compute the SPREF curves.
Remove the target object from the first and the last frames of the GOF.
for Pixels in the other frames **do**
  if The first and last pixels along the curves have been removed **then**
    Remove the pixel.
  end if
end for

sequence. The airplane in the 1st and the 8th frames is erased manually. As shown in Fig. 9, the airplane in the other 6 frames is then removed successfully, although the background is also moving. Compared to the way of manual selection in each frame, the amount of manual work has been reduced by 75%.

4.2 Video Inpainting

When an object is removed from a video, it leaves a spatiotemporal hole behind. Video inpainting is filling this hole naturally, while preserving the video's temporal regularity. In previous studies, this regularity has been preserved *implicitly* by various techniques, such as motion layer mosaicing [25, 26], and hole filling with small spatiotemporal patches that do not disturb the temporal regularity [27–29]. The patches in the latter methods need to be searched in the huge spatiotemporal space of the video. However, these techniques can be avoided by using SPREF since the regularity of a spatiotemporal region is modeled *explicitly*. In this section, we first explain how to inpaint a group of frames gof, when the motion of the pixels surrounding the spatiotemporal hole can be modeled by a single SPREF. Next, we extend this solution to the cases, where the hole may lie on the motion boundaries of the frames. We present an algorithm based on the segmentation of the video for this purpose.

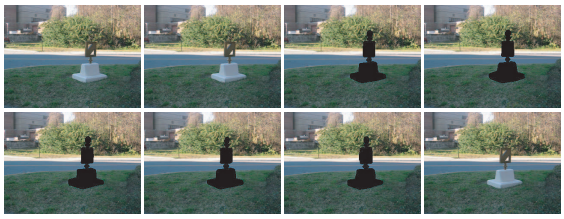


Figure 10: The statue is artificially removed from the mid-frames, then inpainted back by our algorithm, as shown in Fig. 12.

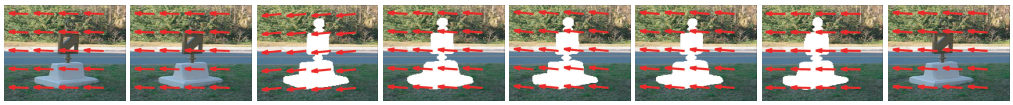


Figure 11: The frames from the selected subsequence used for computing the flow. The SPREF directions are superimposed on their respective frames.

4.2.1 Inpainting with Single SPREF

When a group of frames that undergo global motion contain a spatiotemporal hole, we can assume that the directions of regularity of the hole is the same as that of the pixels surrounding it. Fig. 10 shows a clip from the Statue sequence, where the statue is removed from frames in the middle. Since the video undergoes a global translational motion, the hole can be inpainted using only the *xy-parallel* T-SPREF component. Although the sequence is damaged by the removal of the statue, its SPREF can still be computed from the undamaged pixels. In order to do this, a *subgroup of frames* (*subgof*) that fully contains this spatiotemporal hole is automatically selected, and then SPREF energy function (2) is solved to find the *xy-parallel* flow directions. These directions, along which the *subgof* varies the least, are converted to a set of coordinates by computing the *SPREF curves*.

Fig. 11 shows the selected *subgof*, and its SPREF vectors superimposed on each frame. Note that the flow directions are parallel in each frame, due to the constraint of *xy-parallelism*. The flow directions reveal certain coordinates, represented by *SPREF curves*, along which the *subgof* varies the least. These curves constitute the most important part of our video inpainting algorithm. Since the pixel appearances on a flow curve vary smoothly, we can safely assume that the appearances of the damaged pixels can not deviate too much from those of the undamaged ones. Hence, we fit a spline to the known pixel intensities on the flow curve, and interpolate the missing appearance values from this spline. Note that we allow no damaged pixels at the *subgof* boundaries to guarantee avoiding any extrapolations that may result in extremely high or low intensity values in this case.

Fig. 12 shows the results of our inpainting algorithm, where we inpaint the Statue back in the intermediate frames successfully, and show the original ones for comparison. The best number of spline parameters, l in flow approximation equation (5), is selected as the one that minimizes the error energy function (2).

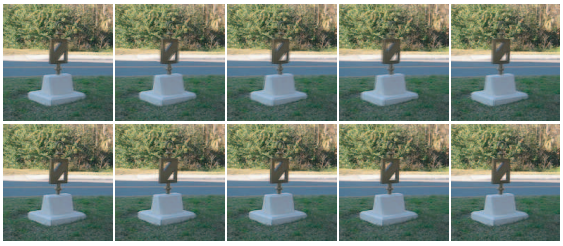


Figure 12: The results of the SPREF-based video inpainting: (1st Row) The inpainted frames. (2nd Row) The original frames.

Figure 13: First row: clip of the original walking human sequence. Second row: sequence after removing the sign board.

4.2.2 Moving Object Inpainting

Inpainting using a single SPREF in previous section works well as long as the motion of the group of frames (gof) are well-approximated. However, this may not always be the case. In many natural video sequences, we need to inpaint the hole left by removing a static object, which partly occludes another moving object. In such cases, the single SPREF inpainting algorithm does not work and a more sophisticated method is needed.

The first row of Fig. 13 shows a clip from the walking human sequence in which a man is partly occluded by a sign board. In the second row, the sign board is removed and we need to inpaint the hole marked by red. In this example, the spatiotemporal hole is on the boundary of the background and the walking human. The background is static while the man is moving forward. Since there exist multiple directions of regularities due to multiple layers of motion, a single SPREF is not able to handle it. The solution lies in segmenting the gof, which is described in Section 2.3. The segmentation creates many $subgof$s so that each $subgof$ contains a unique SPREF. When computing the SPREF for the $subgof$s, three cases are considered. In the first case where there is no hole region in the current subgof, the SPREF can be computed directly. In the second case where part of the subgof is hole region, the SPREF is computed according to the non-hole part. In the last case where the subgof is totally composed by hole region, the SPREF is estimated by interpolating SPREFs of neighboring subgofs.

Then we perform the inpainting along the flow curves with interpolation and/or extrapolation. During inpainting, there are two questions to be answered. First, since only one of the three

SPREF components will be used for inpainting, which one should be used? Second, as mentioned earlier, the same region of a spatiotemporal hole may be contained by more than one $subgof$s (oct-tree nodes). As only one of these nested $subgof$s can result in the visually most plausible inpainting, which one should be selected? The answers to the questions lie in the quality of the inpainted video. The one which obtains the best video inpainting result will be selected. The inpainting results depends on the *goodness* of the SPREF, which is measured by the amount of flow errors in energy functions (2)–(4) of T-SPREF and similarly for A-SPREF. Thus, for each $subgof$ segment, the SPREF component with minimum cost is assigned to it. Then the inpainting is done by using the segment with the minimum flow error from those containing the region of interest. The overall video inpainting algorithm is summarized as follows.

Algorithm 2 Video Inpainting

Divide the gof into smaller $subgof$s according to the segmentation algorithm in Section 2.3.
Compute the SPREF of each $subgof$.
for each leaf node (smallest $subgof$s) in the oct-tree **do**
  Find all its parents up to the root node.
  Compute the flow error of the leaf node using their SPREFs and the leaf node's own SPREF.
  Assign the SPREF component with the minimum flow error to the segment.
  Find the node whose SPREF results in the minimum flow error.
  Fill the hole in the leaf node using that node's interpolation results.
end for

The spatiotemporal hole in the walking human sequence is then inpainted using the above method. For comparison, we implemented the video inpainting method proposed by Wexler et al. [29] and applied to the same video sequence. The inpainting results are shown in Fig. 14. The method in [29] repairs the video by searching the missing part blindly in global spatiotemporal space, which is very time consuming. With a typical hole size, repairing 10 frames normally takes about 2 hours in a PC with 2GHz CPU, while our method requires only 30 minutes. Furthermore, since each pixel is processed independently and the color value associated with the pixel is obtained by computing the weighting average of the matched patches, the inpainting result sometimes may not be meaningful, although the color of the recovered pixels are consistent with those of their surroundings (see figures (d)-(f) in the third row of Fig. 14). The authors proposed to incorporate some video tracking techniques to increase the weights of the "correct" patches. However, the video tracking itself is a complex problem and the general solution does not exist. On the contrary, our video inpainting algorithm is based on the computed SPREFs, which provide the regularity directions. Therefore, once the SPREF is available, the missing parts of the video can be easily recovered by interpolation. In addition, the basic unit of inpainting is $subgof$ with unique SPREF, which facilitates the integrity of the video (see figures (a)-(c) in the third row of Fig. 14). Since the walking direction of the man in the sequence is not parallel to the camera image plane, which causes complicated motion beyond the scope of the T-SPREF, the A-SPREF is used in this experiment.

(a) (b) (c) (d) (e) (f)

Figure 14: Results of walking human sequence inpainting. First row: inpainting results of our algorithm based on the A-SPREF. Second row: inpainting results of our implementation of [29]. Third row: Zoomed view of the inpainted human in the sequence, (a)-(c) the results of our algorithm and (d)-(f) the results of the method in [29].

4.3 Video Compression

According to the information theory, lower entropy results in higher compression ratio. Thus, if a spatiotemporal region Ω is filtered along the directions of regularity, where entropy is lower, better compression can be obtained. Since SPREF indicates the directions of regularity, it is a very suitable tool to increase the efficiency of the compression. Moreover, its compactness due to the spline representation has a low compression overhead.

In video compression, wavelet based coding techniques have been widely used for their excellent performance. In standard wavelet video coding, the wavelet coefficients are obtained by consecutive filtering and downsampling of a *gof* Ω along the horizontal (x), vertical (y), and temporal (t) dimensions. The filters used are a pair of high-pass ψ and low-pass ϕ filters [30]. A 3D wavelet basis can be constructed by filtering all dimensions with all possible combinations of these filters. The compression is performed by thresholding, followed by the quantization of the wavelet coefficients, so that only the significantly large coefficients have non-zero quantized values. In recent studies, the motion-related regularity of the video has been exploited using *motion-compensated* (MC) wavelets [31–34]. However, since the motion models used in these studies, i.e., block motion [35]/cube motion [36], deformable meshes [37], and dense flow fields [38], are computed for pairs of frames, the model parameters contain temporal redundancy. Moreover, none of the studies in MC wavelets exploit the spatial regularity of the scene, which is another strength of SPREF.

SPREF based video compression can be possible by warping the 3D wavelet basis along the

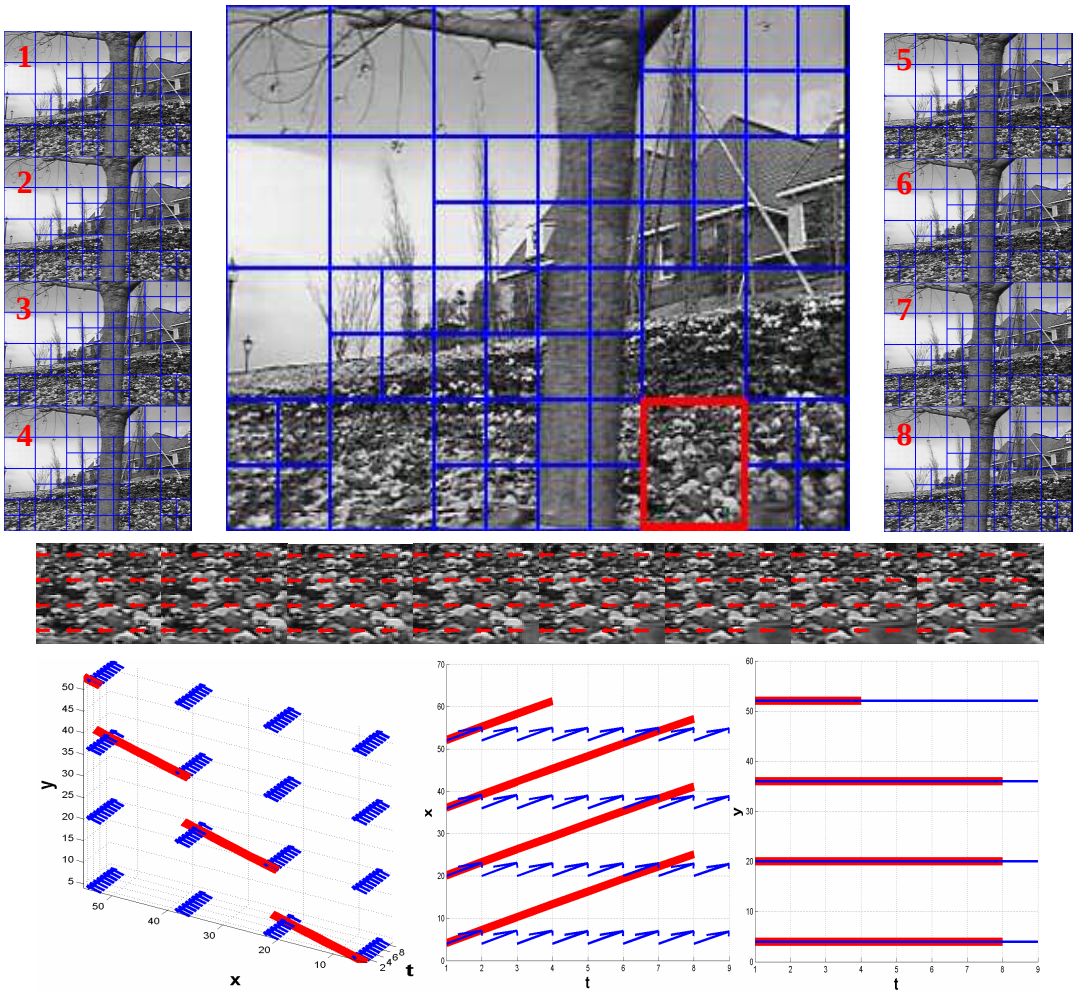


Figure 15: The results for a gof of the Flower Garden sequence ($\Delta = 20$). (1^{st} Row) (Left and Right) The original frames with the optimal segmentation. (Middle) The 3^{rd} frame zoomed-in for details. Since the tree moves very fast, as seen in the original frames, it sweeps a wide region in time, resulting in the segmentation of the $subgof$s in these regions. (2^{nd} Row) SPREF of the $subgof$ marked with red, superimposed on the sub-frames. (3^{rd} Row) The same flow from oblique, top and side views.

flow directions. For this purpose, a warping operator, $\mathcal{W}$, is defined so that the filtering can be performed on the flow curves. Alternatively, if we warp region Ω instead of the basis itself, and define the standard wavelet basis in the warped domain, then transforming this basis back to the original domain produces the *orthogonal warped wavelet basis*. After this step, the compression can be improved further by converting the warped wavelet basis into a *bandelet* basis, which was introduced by Pennec and Mallat [22]. This step involves taking more advantage of the regularity

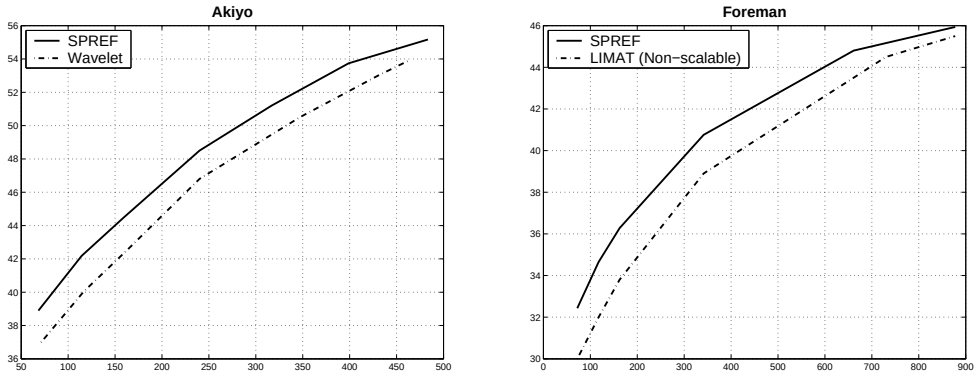


Figure 16: The bit-rate vs PSNR plots of Akiyo and Foreman. Both SPREF-based compression and LIMAT framework are shown in the results.

of the data by replacing the low-pass filters, $\{\phi(t)\}_j$, with high-pass filters, $\{\psi(t)\}_r$, at lower resolutions ($r > j$). This change of basis results in less number of significant coefficients, and it can be performed for other components of SPREFs similarly.

The cost of compression for a given SPREF is approximated by $D + \lambda N$ as we defined earlier. For a given SPREF component, say *xy-parallel* T-SPREF, the energy function (2) can have as many solutions as the spline parameters. The one that minimizes the compression cost is chosen as the correct solution. The choice of λ as a function of the quantization step size Δ can be computed by minimizing the compression cost with respect to Δ. This minimization results in the definition of λ as $\lambda = 3\Delta^2/4\gamma_0$.

The efficiency of the compression depends on the closeness of the approximation of the regularity. Hence, we propose a split/merge algorithm, based on the segmentation described in Section 2.3, for better approximation. This algorithm eventually splits the regions with different regularity characteristics, and merges the similar ones in order to reduce the overhead of storing the SPREF parameters. Fig. 15 shows the results of our segmentation algorithm on a *gof* from the *Flower* sequence. After segmentation is performed, the steps of our algorithm go as follows.

Algorithm 3 Video Compression

Segment region Ω into smaller subregions Ω_i.
for subregion Ω_i **do**
  Compress it using *xy*, *xt* and *yt-parallel* SPREFs separately.
  Compute the compression costs.
  Select the flow direction with lowest cost for compression.
end for
Run a bottom-up split-merge algorithm on the oct-tree.

We show the results of our SPREF based bandelet video compression scheme on some standard video sequences, i.e., Akiyo, Foreman and Mobile. All sequences are at QCIF resolution. We also compare the results of our algorithm with those of the Lifting-based invertible motion adaptive

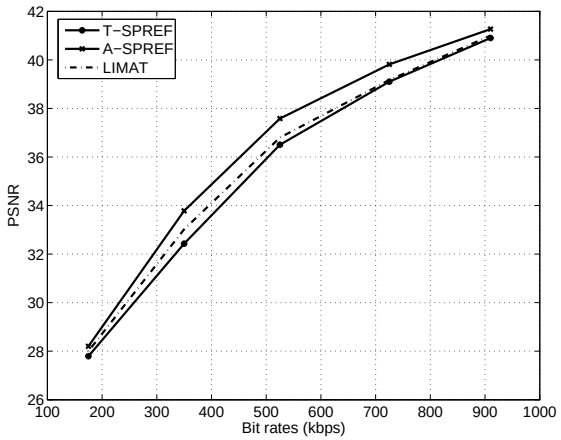


Figure 17: The PSNR plot of the Mobile sequence.

transform (LIMAT) framework of Secker and Taubman in [33], a motion-compensated wavelet video compression technique, for performance evaluation. 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 size of the smallest *subgof* in the oct-tree is $16 \times 16 \times 8$. 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 non-cuboid *subgof*s near the image boundaries.

We give a comparison of the SPREF-based compression and wavelet video compression at various bit-rates in Fig. 16. The improvement as a result of the directional decomposition and bandeletization in SPREF-based compression can be clearly observed in these plots. In these experiments, T-SPREF is used since the motion in these sequence is basically translational.

Fig. 17 shows the compression results using LIMAT and our algorithm based on T-SPREF and A-SPREF. It is noted that the A-SPREF based algorithm performs best, while LIMAT outperforms T-SPREF in this case. The reason is that the motion in the Mobile sequence consists of many non-rigid components such as global zooming out, the swinging toy, and the rotating ball. As we have discussed in Section 2.2, the T-SPREF is not able to approximate non-rigid motion of the objects well. The inaccurate estimation of the directions of regularity degrades the compression performance. The mesh model used in LIMAT can model some of these non-translational motion types better than SPREF. Hence, LIMAT performs marginally better than SPREF in this particular example. However, the non-rigid motion can be well approximated by A-SPREF and the video compression algorithm based on A-SPREF consequently performs much better than the other two.

5 CONCLUSION

We presented a new general framework called SPREF that shows the local directions, along which a spatiotemporal region changes the least. SPREF is a 3D vector field that approximates these

directions with splines. In terms of an image sequence, using splines allows us to incorporate all frames in the solution, which results in a better estimation. The directions of regularity depend on the motion content and the spatial structure of the region. All these cases are handled by three components of SPREF. If the motion in the spatiotemporal region has multiple layers, or if the structure of a static scene is complex, single SPREF is unable to model the directions of regularity. In such cases, the region can be recursively partitioned into smaller subregions using an oct-tree, which have less spatial complexity or less number of motion layers. Then SPREF can be separately computed for each subregion, and these regions can be merged later, or used as it is, depending on the application.

We have shown successful use of SPREF in three popular applications: object removal, video inpainting, and wavelet based video compression. In the first application, a moving object with dynamic background is erased from the video by following the SPREF curve. The amount of manual work has been greatly decreased. In the second application, we inpaint the missing pixels by interpolating them from the observed ones along SPREF curves. The advantages of our method are demonstrated by comparing the results to those of state-of-the-art method. In video compression, we exploit the fact that the data on SPREF curves has smaller entropy. So, we warp the 3D wavelet basis along these curves, and decompose the videos along the directions, which entropy is smaller. Our compression results indicate a significant improvement over the traditional wavelet video compression.

A Solving the T-SPREF Equation

In this section we will show how to solve the energy function (2) by quadratic minimization over the spline parameters, $A = [\alpha_1, \ldots, \alpha_{l_1}, \beta_1, \ldots, \beta_{l_2}]$, where α_i $(i = 1, \ldots, l_1)$ and β_i $(i = 1, \ldots, l_2)$ are the spline parameters for $c_1'(t)$ and $c_2'(t)$, respectively. Let $f_u(x, y, t)$, $f_v(x, y, t)$ and $f_\tau(x, y, t)$ denote the spatiotemporal gradients at the point $(x, y, t) \in \Omega$ and $B_t^l = [b_1^l(t) \ldots b_l^l(t)]^T$ represent the values of the spline points $i = 1, \ldots, l$ at the time t. When being minimized and rearranged based on these variables, Eqn. (2) takes the following form:

$$(A)_{[1\times(l_1+l_2)]} \begin{bmatrix} (S_1)_{[l_1\times l_1]} & (S_3)_{[l_1\times l_2]} \\ (S_2)_{[l_2\times l_1]} & (S_4)_{[l_2\times l_2]} \end{bmatrix} + \left[(S_5)_{[1\times l_1]}\ (S_6)_{[1\times l_2]}\right] = 0, \tag{19}$$

where the column vectors of matrix S_i $(i = 1, \ldots, 6)$, are defined as follows

$$\begin{aligned} S_1(j) &= \sum_t b_j^{l_1}(t) \left(\sum_{x,y} f_u^2\right) B_t^{l_1}, & S_2(j) &= \sum_t b_j^{l_1}(t) \left(\sum_{x,y} f_u f_v\right) B_t^{l_2}, \\ S_3(j) &= \sum_t b_j^{l_2}(t) \left(\sum_{x,y} f_u f_v\right) B_t^{l_1}, & S_4(j) &= \sum_t b_j^{l_2}(t) \left(\sum_{x,y} f_v^2\right) B_t^{l_2}, \\ S_5(j) &= \sum_t b_j^{l_1}(t) \left(\sum_{x,y} f_u f_\tau\right), & S_6(j) &= \sum_t b_j^{l_2}(t) \left(\sum_{x,y} f_v f_\tau\right). \end{aligned} \tag{20}$$

The spline parameters A can then be easily computed by solving Eqn. (19).

By doing a simple necessary change of variables in these equations, we can have the functions for computing *yt-parallel* and *xt-parallel* SPREFs. Or a simpler method is rotating the whole

region so that the desired propagation axis is aligned with t, and the directly applying the above equations for computation.

B Solving the A-SPREF Equation

The A-SPREF parameters are solved in a fashion similar to T-SPREF, by minimizing the same flow energy function (2) by quadratic minimization over the spline parameters. One can fit a spline to all affine parameters separately. However, since this will increase the search space, we assign one spline per flow component. In case of temporal propagation axis of size N, this means that for the flow $c_i'(x,y,t)$, $a_{ij}(t) = \sum_n \alpha_n^{ij} b(2^{-l}t - n)$ for $j = 1,2,3$. The spline parameters to find for a temporal flow propagation axis are:

$$A = [\alpha_1^{11}...\alpha_{l_1}^{11}, \alpha_1^{12}...\alpha_{l_1}^{12}, \alpha_1^{13}...\alpha_{l_1}^{13}, \alpha_1^{21}...\alpha_{l_2}^{21}, \alpha_1^{22}...\alpha_{l_2}^{22}, \alpha_1^{23}...\alpha_{l_2}^{23}],$$

where l_1 and l_2 are the number of control points for each spline. When minimized and rearranged according to these variables, Eqn. (8) takes the same form as Eqn. (19). However, extension of the terms :

$$(A)_{[l_1+l_2]} * ((B)_{3[l_1+l_2]\times 6N} * (M)_{6N\times 3[l_1+l_2]}) + (N)_{1\times 3[l_1+l_2]} = 0. \tag{21}$$

In this equation, B is the matrix of all splines. Before defining this matrix, we will define a matrix C that stores all possible values of a spline b with l_i control points: $C_{l_i}[t,n] = b(2^{-l_i}t - n)$. With this definition, next we define B:

$$B = \begin{bmatrix} C_{l_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & C_{l_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{l_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{l_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{l_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{l_2} \end{bmatrix}.$$

We will build the matrix M, using these $(1 \times N)$small matrices S_1 to S_6, defined as:

$$\begin{aligned}
S_1[n] &= \sum_{x,y,t=n} [\, x^2f_x^2 \;\; xyf_x^2 \;\; xf_x^2 \;\; x^2f_xf_y \;\; xyf_xf_y \;\; xf_xf_y \,]^T, \\
S_2[n] &= \sum_{x,y,t=n} [\, xyf_x^2 \;\; yyf_x^2 \;\; yf_x^2 \;\; xyf_xf_y \;\; yyf_xf_y \;\; yf_xf_y \,]^T, \\
S_3[n] &= \sum_{x,y,t=n} [\, xf_x^2 \;\; yf_x^2 \;\; f_x^2 \;\; xf_xf_y \;\; yf_xf_y \;\; f_xf_y \,]^T, \\
S_4[n] &= \sum_{x,y,t=n} [\, x^2f_xf_y \;\; xyf_xf_y \;\; xf_xf_y \;\; x^2f_y^2 \;\; xyf_y^2 \;\; xf_y^2 \,]^T, \\
S_5[n] &= \sum_{x,y,t=n} [\, xyf_xf_y \;\; yyf_xf_y \;\; yf_xf_y \;\; xyf_y^2 \;\; yyf_y^2 \;\; yf_y^2 \,]^T, \\
S_6[n] &= \sum_{x,y,t=n} [\, xf_xf_y \;\; yf_xf_y \;\; f_xf_y \;\; xf_y^2 \;\; yf_y^2 \;\; f_y^2 \,]^T.
\end{aligned}$$

With these definitions, $\mathbf{M}$ is defined as follows:

$$\begin{aligned}
M[\,t,\ N+t,\ 2N+t,\ 3N+t,\ 4N+t,\ 5N+t\,,k] &= C_{l_1}[k,t] * S_1(t),\\
M[\,t,\ N+t,\ 2N+t,\ 3N+t,\ 4N+t,\ 5N+t\,,k+l_1] &= C_{l_1}[k,t] * S_2(t),\\
M[\,t,\ N+t,\ 2N+t,\ 3N+t,\ 4N+t,\ 5N+t\,,k+2l_1] &= C_{l_1}[k,t] * S_3(t),\\
M[\,t,\ N+t,\ 2N+t,\ 3N+t,\ 4N+t,\ 5N+t\,,k+2l_1 &= C_{l_2}[k,t] * S_4(t),\\
M[\,t,\ N+t,\ 2N+t,\ 3N+t,\ 4N+t,\ 5N+t\,,k+3l_1+l_2] &= C_{l_2}[k,t] * S_5(t),\\
M[\,t,\ N+t,\ 2N+t,\ 3N+t,\ 4N+t,\ 5N+t\,,k+3l_1+2l_2] &= C_{l_2}[k,t] * S_6(t).
\end{aligned}$$

The final term in (21) is N, and it is defined as follows:

$$N[\,k,k+l_1,k+2l_1,k+2l_1,k+3l_1+l_2,k+3l_1+2l_2\,] =$$
$$\sum_{n=1}^{N}\left[C_{l_1}[k,n]\Big(\sum_{x,y,t=n}[xf_xf_t\ \ yf_xf_t\ \ f_xf_t]\Big), C_{l_2}[k,n]\Big(\sum_{x,y,t=n}[xf_yf_t\ \ yf_yf_t\ \ f_yf_t]\Big)\right].$$

The solution for A-SPREFs with different propagation axes can be obtained by either a change of variables in these equations, or by rotating the spatiotemporal region such that the desired axis of propagation is aligned with t.

References

[1] Z.-N. Li, X. Zhong, and M. S. Drew, "Spatialtemporal joint probability images for video segmentation," *Pattern Recognition*, vol. 35, pp. 1847–1867, 2002.

[2] J. Y. A. Wang and E. H. Adelson, "Spatio-temporal segmentation of video data," in *Proc. SPIE: Image and Video Processing II*, vol. 2182, 1994, pp. 120–131.

[3] O. Alatas, O. Javed, and M. Shah, "Video compression using structural flow," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2005, pp. 245–248.

[4] H.-H. Nagel and A. Gehrke, "Spatiotemporally adaptive estimation and segmentation of OF-fields," in *Proc. European Conf. Computer Vision*, H. Burkhardt and B. Neumann, Eds., vol. 2, 1998, pp. 86–102.

[5] J. G. Choi, S.-W. Lee, and S.-D. Kim, "Spatio-temporal video segmentation using a joint similarity measure," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 279–286, 1997.

[6] J. Fan, J. Yu, G. Fujita, T. Onoye, L. Wu, and I. Shirakawa, "Spatiotemporal segmentation for compact video representation," *Signal Processing: Image Communication*, vol. 16, no. 6, pp. 553–566, 2001.

[7] H. Greenspan, J. Goldberger, and A. Mayer, "A probabilistic framework for spatio-temporal video representation & indexing." in *Proc. European Conf. Computer Vision*, vol. 4, 2002, pp. 461–475.

[8] D. J. Heeger, "Optical flow using spatiotemporal filters," *Int. J. Computer Vision*, pp. 279–302, 1988.

[9] T. Y. Tian and M. Shah, "Recovering 3D motion of multiple objects using adaptive hough transform." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 10, pp. 1178–1183, 1997.

[10] E. Simoncelli and E. Adelson, "Computing optical flow distributions using spatio-temporal filters," MIT Media Lab Vision and Modeling, Tech. Rep. 165, 1991.

[11] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[12] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Seventh Int. Joint Conf. Artificial Intelligence*, Vancouver, Canada, 1981, pp. 674–679.

[13] E. H. Adelson and J. Bergen, "Spatiotemporal energy models for the perception of motion," in *J. Opt. Soc. Amer.*, vol. 2, February 1985, pp. 284–299.

[14] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *Int. J. Computer Vision*, vol. 2, no. 3, pp. 283–310, 1989.

[15] W. T. Freeman and E. Adelson, "The design and use of steerable filters," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 13, no. 9, pp. 891–906, Sept. 1991.

[16] M. Allmen and C. R. Dyer, "Computing spatiotemporal surface flow," in *Proc. IEEE Int. Conf. Computer Vision*, 1990, pp. 47–50.

[17] S. L. Peng and G. Medioni, "Interpretation of image sequences by spatio-temporal analysis," in *Proc. Workshop on Visual Motion*, 1989, pp. 344–351.

[18] H. H. Baker and R. C. Bolles, "Generalizing epipolar-plane image analysis on the spatiotemporal surface," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1988, pp. 2–9.

[19] C.-W. Ngo, T.-C. Pong, and H.-J. Zhang, "Motion analysis and segmentation through spatio-temporal slices processing," *IEEE Trans. Image Processing*, vol. 12, pp. 341–355, Mar. 2003.

[20] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proc. IEEE Int. Conf. Computer Vision*, 2003, pp. 432–439.

[21] S. A. Niyogi and E. H. Adelson, "Analyzing and recognizing walking figures in XYT," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1994, pp. 469–474.

[22] E. L. Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Trans. Image Processing*, vol. 14, pp. 423–438, Apr. 2005.

[23] O. Alatas, O. Javed, and M. Shah, "Video compression using spatiotemporal regularity flow," *IEEE Trans. Image Processing*, vol. 15, no. 12, pp. 3812–3823, Dec. 2006.

[24] R. Szeliski and J. Coughlan, "Spline-based image registration," *Int. J. Computer Vision*, vol. 22, pp. 199–218, 1999.

[25] Y. Zhang, J. Xiao, and M. Shah, "Motion layer based object removal in videos," in *Proc. IEEE WACV'05*, 2005, pp. 516–521.

[26] J. Jia, T. Wu, Y. Tai, and C. Tang, "Video repairing: Inference of foreground and background under severe occlusion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. 364–371.

[27] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 335–362.

[28] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, 2005, pp. 69–72.

[29] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, 2004, pp. 120–127.

[30] S. G. Mallat, *A Wavelet tour of signal processing.* Academic Press, 1997.

[31] J.-R. Ohm, "Temporal domain sub band video coding with motion compensation," in *ICASSP'92*, vol. 3, 1992, pp. 229–232.

[32] J. Konrad and E. Dubois, "Bayesian estimation of vector fields," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 14, pp. 910–927, September 2002.

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

[34] J. E. Fowler and Y. Wang, "3D video coding using redundant wavelet multihypothesis and motion-compensated temporal filtering," in *Proc. IEEE Int. Conf. Image Processing*, 2003, pp. 755–758.

[35] 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. 1, pp. 85–94, 1999.

[36] A. A. Kassim, P. Yan, W. S. Lee, and K. Sengupta, "Motion compensated lossy-to-lossless compression of 4D medical images using integer wavelet transforms," *IEEE Trans. Inform. Technol. Biomed.*, vol. 9, no. 1, pp. 132–138, Mar. 2005.

[37] D. Taubman and A. Zakhor, "Multirate 3D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, pp. 572–588, Sept. 1994.

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