SPATIO-TEMPORAL VIDEO INTERPOLATION AND DENOISING USING MOTION-ASSISTED STEERING KERNEL (MASK) REGRESSION

H. Takeda[1]*, P. van Beek*[2]*, P. Milanfar*[1]

[1]Department of Electrical Engineering, University of California at Santa Cruz, USA
[2]Sharp Laboratories of America, Camas, Washington, USA
{htakeda,milanfar}@soe.ucsc.edu, pvanbeek@sharplabs.com

ABSTRACT

In this paper, we extend a (2-D) data-adaptive steering kernel regression framework for image processing to a (3-D) spatio-temporal framework for processing video. In particular, we propose a motion-assisted steering kernel (MASK) suitable for interpolating video data spatially, temporally, or spatio-temporally, and for video noise reduction. We present an algorithm for multi-frame interpolation and reconstruction of video data, and present several simulation results on synthetic and real video data. Comparisons between single-frame and multi-frame kernel regression and with other methods demonstrate the effectiveness of our approach.

Index Terms— video signal processing, interpolation, denoising, adaptive estimation, motion analysis

1. INTRODUCTION

Advances in video display technology have increased the need for high-quality and robust video interpolation and artifact removal methods. For example, LCD flat-panel displays are currently being developed with very high resolution and very high frame rates, namely 4096x2160 pixels at 120 Hz. Such displays may exceed the highest spatial resolution and frame rate of video content commonly available, namely 1920x1080, 60 Hz progressive High Definition (HD) video, in applications such as HD-TV and HD-DVD. In such (and other) applications, the goal for spatial and temporal video interpolation and reconstruction is to enhance the resolution of the input video in a manner that is visually pleasing and artifact-free. Common visual artifacts that may occur in spatial and temporal interpolation are: edge jaggedness, ringing, blurring of edges and texture detail, motion blur and/or judder. In addition, the input video usually contains noise and other artifacts, e.g. due to compression. Due to increasing sizes of modern video displays, as well as incorporation of new display technologies (e.g. higher brightness, wider color gamut), artifacts in the input video and those introduced by scaling are amplified, and are more visible than in the past.

In this paper, we build upon a *kernel regression* framework for interpolating and denoising (2-D) images proposed in [1]. Kernel regression methods are closely related to bilateral filtering and normalized convolution [2]. These methods can achieve accurate and robust image reconstruction results, due to their use of robust error norms and locally adaptive weighting functions [1, 2, 3]. We extend this 2-D framework to a spatio-temporal (3-D) framework for processing video. Specifically, we propose a *motion-assisted steering kernel* (MASK): an approach that utilizes an analysis of the local orientation and local motion to steer spatio-temporal regression kernels. Subsequently, local kernel regression is applied to compute weighted least-squares optimal pixel estimates. Although 2-D kernel regression has been applied to achieve super-resolution reconstruction through fusion of multiple pre-registered frames [1, 2], the proposed method is different in that it does not require explicit motion compensation of the video frames. Instead, we use 3-D weighting kernels that are "warped" according to estimated motion vectors, such that the regression process acts directly upon the video data.

The proposed method is capable of simultaneous spatial interpolation with resolution enhancement, temporal video interpolation and noise reduction. Prior multi-frame resolution-enhanced or super-resolution (SR) reconstruction methods (for overviews see [4] and [5]) often consider only global translational or affine motion; local motion and object occlusions are often not addressed. Many SR methods require explicit motion compensation, which may involve interpolation or rounding of displacements to grid locations. These issues can have a negative impact on accuracy and robustness. Our proposed method is capable of handling local motion, avoids explicit motion compensation, and is more robust. Also, we incorporated temporal video interpolation (frame rate conversion) in the proposed method. Temporal video interpolation (for an overview see [6]) is usually not addressed in prior SR work.

In the next section, we describe spatio-temporal kernel regression and the proposed motion-assisted steering kernel in more detail. In Section 3, we describe an algorithm for interpolating and denoising video data based on the proposed steering kernel. We report on our experiments in Section 4, and conclude in Section 5.

2. SPATIO-TEMPORAL STEERING KERNEL REGRESSION

2.1. Kernel regression

For video processing, we define a spatio-temporal data model as

$$y_i = z(\mathbf{x}_i) + \varepsilon_i, \quad i = 1, \cdots, P, \quad \mathbf{x}_i = [x_{1i}, x_{2i}, t_i]^T, \qquad (1)$$

where y_i is a given sample (pixel) at location $\mathbf{x}_i$, x_{1i} and x_{2i} are the spatial coordinates, t_i is the temporal coordinate, $z(\cdot)$ is the *regression function*, and ε_i is i.i.d zero mean noise. P is the number of samples in a spatio-temporal neighborhood of interest, which spans multiple video frames.

In order to estimate the value of $z(\cdot)$ at point $\mathbf{x}$, given the above data samples y_i, we can rely on a local N^{th} order Taylor expansion about $\mathbf{x}$. We denote the pixel value of interest $z(\mathbf{x})$ by β_0, while $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$, ..., $\boldsymbol{\beta}_N$ denote vectors containing the first-order, second-order, ..., N^{th} order partial derivatives of $z(\cdot)$ at $\mathbf{x}$,

This work was supported in part by Sharp Labs of America.

resulting from the Taylor expansion. For example, $\beta_0 = z(\mathbf{x})$ and $\boldsymbol{\beta}_1 = [z_{x_1}(\mathbf{x}), z_{x_2}(\mathbf{x}), z_t(\mathbf{x})]^T$.

The unknowns, $\{\boldsymbol{\beta}_n\}_{n=0}^N$, can be estimated from $\{y_i\}_{i=1}^P$ using the following weighted least-squares optimization procedure:

$$\min_{\{\boldsymbol{\beta}_n\}_{n=0}^N} \sum_{i=1}^P \big[y_i - \beta_0 - \boldsymbol{\beta}_1^T(\mathbf{x}_i - \mathbf{x}) \\ -\boldsymbol{\beta}_2^T \mathrm{vech}\left\{(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^T\right\} - \cdots\big]^2 \mathsf{K}(\mathbf{x}_i - \mathbf{x}) \quad (2)$$

where N is the regression order and $\mathsf{K}(\cdot)$ is a *kernel function* that weights the influence of each sample. Typically, samples near $\mathbf{x}$ are given higher weights than samples farther away. In *classical* kernel regression, $\mathsf{K}(\cdot)$ is defined as follows:

$$\mathsf{K}(\cdot) \equiv K_{\mathbf{H}}(\mathbf{x}_i - \mathbf{x}) = \frac{1}{\det(\mathbf{H})} K\big(\mathbf{H}^{-1}(\mathbf{x}_i - \mathbf{x})\big),$$

where $\mathbf{H}$ is a smoothing matrix that dictates the shape of the kernel function (see [1]). The prototype kernel function K can be chosen suitably, for example using a Gaussian function.

It can be shown that, regardless of the choice of kernel K and order N, the resulting estimator for $z(\mathbf{x})$ can be written as:

$$\hat{z}(\mathbf{x}) = \widehat{\beta}_0 = \sum_{i=1}^P W_i(\mathbf{x}; K, \mathbf{H}, N)\, y_i, \quad (3)$$

where $W_i(\cdot)$ is called the "equivalent" kernel function for y_i. This expression illustrates the fact that "classical" kernel regression is a local weighted averaging or linear filtering process. This approach yields point-wise estimates of $z(\cdot)$ with minimal assumptions on the signal or noise.

The classical kernel regression framework was extended in [1] by allowing the shape of the kernel to be adapted locally, dependent on the image data. In particular, *steering kernels* were proposed that adapt spatially to local orientation structure, so that their iso-contours are rotated and elongated along edges, and dilated in smooth image areas. This results in strong preservation of details in the final output and greatly improves estimates in the case of denoising and interpolation. Note that these data-adaptive weights result in locally *nonlinear* filtering on the data [1].

2.2. Adaptive spatio-temporal steering kernels

A *3-D steering kernel* $\mathsf{K}_{\mathrm{3DSK}} \equiv K_{\mathbf{H}_i^{\mathrm{3D}}}(\mathbf{x}_i - \mathbf{x})$ is a direct extension of the 2-D steering kernel defined in [1]. The 3×3 data-dependent steering matrix $\mathbf{H}_i^{\mathrm{3D}}$ can be defined as

$$\mathbf{H}_i^{\mathrm{3D}} = h\big(\mathbf{C}_i^{\mathrm{3D}}\big)^{-\frac{1}{2}} \quad (4)$$

where h is a global smoothing parameter and $\mathbf{C}_i^{\mathrm{3D}}$ is a covariance matrix based on the sample variations in a local (3-D) neighborhood around sample $\mathbf{x}_i$. We can construct the matrix $\mathbf{C}_i^{\mathrm{3D}}$ parametrically as $\mathbf{C}_i^{\mathrm{3D}} = \gamma_i \mathbf{R}_i \boldsymbol{\Lambda}_i \mathbf{R}_i^T$, where $\mathbf{R}_i$ is a 3-D rotation matrix, $\boldsymbol{\Lambda}_i$ is a 3-D elongation matrix, and γ_i is a scaling parameter. We have found that such an approach performs reasonably well for spatial upscaling of video. However, this 3-D kernel does not consider the specific spatio-temporal characteristics of video data. In particular, problems may occur in the presence of large object displacements (fast motion). This may result in either shrinking of the kernel in the temporal direction, or spatial blurring (as the kernel weights spread across unrelated data samples), both undesirable effects.

To ameliorate the above problems, we introduce a data-adaptive kernel called *motion-assisted steering kernel* (MASK), specifically for video data. A good choice for steering spatio-temporally is to consider local motion or optical flow vectors caused by object motion in the scene, in conjunction with spatial steering along local edges and isophotes. Spatial steering should consider the locally dominant orientation of the pixel data and should allow elongation of the kernel in this direction, as well as spatial scaling. Spatio-temporal steering should allow alignment of the kernel weights with the local optical flow or motion trajectory, as well as overall temporal scaling. Hence, we construct our spatio-temporal kernel as a product of a spatial- and motion-steering kernel, and a kernel that acts temporally:

$$\mathsf{K}_{\mathrm{MASK}} \equiv \frac{1}{\det(\mathbf{H}_i^{\mathrm{s}})} K\big((\mathbf{H}_i^{\mathrm{s}})^{-1}\mathbf{H}_i^{\mathrm{m}}(\mathbf{x}_i - \mathbf{x})\big)\, K_{h_i^{\mathrm{t}}}(t_i - t), \quad (5)$$

where $\mathbf{H}_i^{\mathrm{s}}$ is a 3×3 *spatial steering* matrix, $\mathbf{H}_i^{\mathrm{m}}$ is a 3×3 *motion steering* matrix, and h_i^{t} is a *temporal steering* parameter. These data-dependent kernel components determine the steering action at sample $\mathbf{x}_i$, as illustrated in Fig. 1, and are described next.

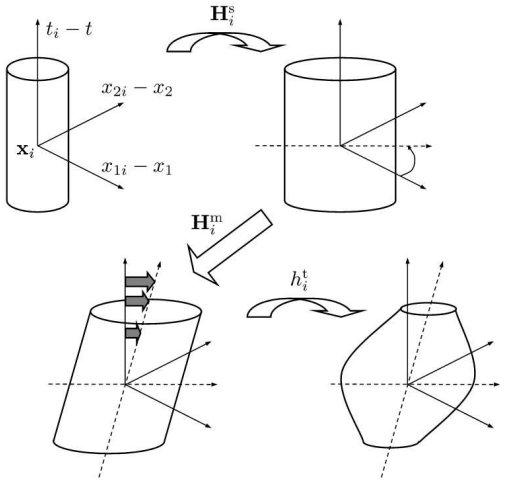


Fig. 1. Illustration of MASK (iso-surfaces) and steering action.

Following [1], the spatial steering matrix $\mathbf{H}_i^{\mathrm{s}}$ is defined by:

$$\mathbf{H}_i^{\mathrm{s}} = h^{\mathrm{s}} \begin{bmatrix} \mathbf{C}_i^{\mathrm{s}} & \\ & 1 \end{bmatrix}^{-\frac{1}{2}}, \quad (6)$$

where h^{s} is a global spatial smoothing parameter, and $\mathbf{C}_i^{\mathrm{s}}$ is a 2×2 covariance matrix capturing the sample variations in a local spatial neighborhood around $\mathbf{x}_i$. $\mathbf{C}_i^{\mathrm{s}}$ is constructed in a parametric manner, based on: an orientation angle θ_i that determines rotation of the kernel in the spatial (x_1, x_2) plane; an elongation parameter σ_i that determines spatial elongation of the kernel along its spatial axes; a scaling parameter γ_i that determines local spatial scaling.

The motion steering matrix $\mathbf{H}_i^{\mathrm{m}}$ is constructed on the basis of a local estimate of the motion (or optical flow vector) $\mathbf{m}_i = [m_{1i}, m_{2i}]^T$ at $\mathbf{x}_i$. Namely, we warp the kernel along the local motion trajectory using the following shearing transformation:

$$\begin{cases} (x_{1i} - x_1) \leftarrow (x_{1i} - x_1) - m_{1i} \cdot (t_i - t) \\ (x_{2i} - x_2) \leftarrow (x_{2i} - x_2) - m_{2i} \cdot (t_i - t) \end{cases},$$

see Fig. 1. Hence,

$$\mathbf{H}_i^{\mathrm{m}} = \begin{bmatrix} 1 & 0 & -m_{1i} \\ 0 & 1 & -m_{2i} \\ 0 & 0 & 0 \end{bmatrix}. \quad (7)$$

Assuming a spatial prototype kernel was used with elliptical footprint, this results in a spatio-temporal kernel with the shape of a tube or cylinder with elliptical cross-sections at any time instance t. Also, the center point of each such cross-section moves along the motion path.

The final component of Eq. (5) is a temporal kernel that provides temporal penalization. A natural approach is to give higher weight to samples in frames closer to t. The relative temporal extent of the kernel is controlled by the temporal scaling parameter h_i^{t}, which can be adapted based on a measure of the reliability of the local motion vector estimate. Note that the proposed spatio-temporal regression approach implicitly assumes that the local motion field is smooth within the (3-D) region of support, i.e. across several frames used in the estimation. The above temporal penalization mechanism can be used to mitigate the effects of this assumption.

3. VIDEO PROCESSING BASED ON MOTION-ASSISTED SPATIO-TEMPORAL KERNEL

A video interpolation and denoising algorithm based on motion-assisted spatio-temporal steering kernel regression is illustrated in Fig. 2. The algorithm estimates spatial and motion steering parameters using gradient-based techniques. Hence, we first compute initial estimates of the spatial and temporal derivatives $\hat{z}_{x_1}(\cdot), \hat{z}_{x_2}(\cdot), \hat{z}_t(\cdot)$, e.g. based on classic kernel regression. Let $\hat{\mathbf{z}}_{x_1}$, $\hat{\mathbf{z}}_{x_2}$ and $\hat{\mathbf{z}}_t$ denote vectors containing (in lexicographical order) derivative estimates from the pixels in a local analysis window w_i around $\mathbf{x}_i$, i.e. $\hat{z}_{x_1}(\mathbf{x}_j), \hat{z}_{x_2}(\mathbf{x}_j), \hat{z}_t(\mathbf{x}_j), \mathbf{x}_j \in w_i$. A robust estimate of the spatial orientation (θ_i), elongation (σ_i) and scaling (γ_i) parameters at $\mathbf{x}_i$ can be obtained by applying singular value decomposition (SVD) to the matrix $\mathbf{G}_i = [\hat{\mathbf{z}}_{x_1}, \hat{\mathbf{z}}_{x_2}]$ containing spatial gradient data. We refer to [1] for further details. A motion vector ($\mathbf{m}_i$) at $\mathbf{x}_i$ is estimated using the well-known Lucas and Kanade method, i.e. by solving $\mathbf{G}_i\mathbf{m}_i + \hat{\mathbf{z}}_t = \underline{\mathbf{0}}$ in the least-squares sense [7].

Local steering parameters are estimated at each pixel location $\mathbf{x}_i$ in the region of support for the final regression step. Given the steering information, MASK regression is applied on the input video data (y_i) to perform actual interpolation (upscaling) and/or denoising, generating the estimated pixel values $\widehat{\beta}_0 = \hat{z}(\mathbf{x})$. Upscaling can be spatial, temporal or both. The kernel regression stage involves determining the equivalent steering kernel function weights $W_i(\mathbf{x}; \mathbf{H}_i^{\mathrm{s}}, \mathbf{H}_i^{\mathrm{m}}, h_i^{\mathrm{t}}, K, N)$ given the steering parameters, and finally local spatio-temporal regression as follows:

$$\hat{z}(\mathbf{x}) = \sum_{i=1}^{P} W_i(\mathbf{x}; \mathbf{H}_i^{\mathrm{s}}, \mathbf{H}_i^{\mathrm{m}}, h_i^{\mathrm{t}}, K, N)\, y_i. \tag{8}$$

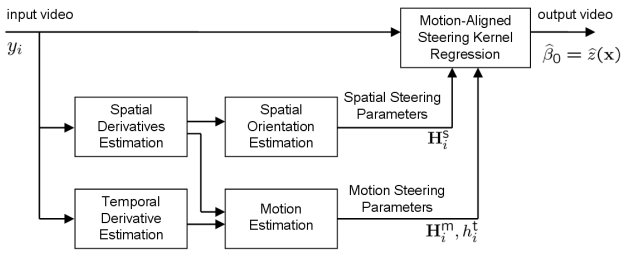


Fig. 2. Illustration of video processing based on motion-assisted spatio-temporal steering kernel (MASK) regression.

4. EXPERIMENTS

We first performed a controlled simulation with a synthetically generated video sequence. Each frame in this sequence was generated by cropping, lowpass filtering and downsampling by a factor of 8 (in each direction) an area from the high-resolution "Cafe" image. The area taken from the original image was shifted from frame to frame, resulting in $[0.5, 0.5]^T$ pixel per frame translation in the downsampled frames. The downsampled frames were upscaled spatially from 256x256 to 512x512 pixels using the proposed algorithm as well as comparison methods. We repeated this simulation with different lowpass filters from a family of maximally flat FIR filters, such that the test sequences contained decreasing amounts of aliasing. To compute PSNR, a reference video sequence was generated using the first filter from the same family and downsampling by 4 (instead of 8). The proposed multi-frame kernel regression method (MASK) was compared to frame-by-frame (2-D) steering kernel regression (SKR as in [1]), and frame-by-frame (2-D) upscaling based on the Total Variation (TV) minimization method of [8] (with optimized regularization parameter). MASK was provided with the (known) motion, since in this experiment we wished to confirm its performance in the absence of motion estimation errors. Note that we used 2^{nd}-order regression in all our experiments ($N = 2$). The spatial smoothing parameter was $h^s = 0.75$ for both SKR and MASK. For MASK, we kept the temporal scaling parameter h_i^{t} constant, corresponding to a fixed temporal support of 5 frames. Both the PSNR results, in Table 1, and the visual results, show a performance gain for MASK over single-frame (2-D) methods, and confirm it is capable of resolution synthesis (reconstruction). MASK can take advantage of the aliasing present in the input. This aliasing can result in severe artifacts for single-frame methods. Fig. 3 shows example frames.

Table 1. Average PSNR (in dB) for Cafe video sequences generated with different max-flat filters with decreasing lowpass band width (max-flat filter design parameters in the top row).

Algorithm	41/05	41/01	81/01	81/00
TV (2-D)	20.33	20.30	19.66	18.37
SKR (2-D)	20.66	20.89	20.05	18.54
MASK (3-D)	24.00	22.13	20.62	18.74

(a) SKR (b) MASK

Fig. 3. Cafe (41/01) results (frame 15, 256x256 cropped area).

Our second experiment used a portion of the "Foreman" sequence at QCIF (176x144) resolution, after applying MPEG-2 compression at 3 different bit rates. We used MASK to simultaneously perform: spatial interpolation by 2, i.e. upscaling each frame to CIF (352x288) resolution; temporal interpolation by 2, i.e. doubling

the frame rate; and removal of compression artifacts (blocking and mosquito noise). We compared MASK (with 5 frames temporal support) to several frame-by-frame methods, namely bicubic interpolation, TV minimization, and 2-D SKR. In MASK, we estimated motion as described in Section 3. We used $h^s = 2.0, 1.75$ and 1.50 for SKR and MASK. In the PSNR results in Table 2 we only compared the original frames - not intermediate ones. Table 2 shows about 1 dB gain was achieved by MASK over 2-D SKR, and demonstrates its applicability in sequences with complex local motion. Example frames are shown in Fig. 4. Bicubic interpolation simply amplifies the compression artifacts. The TV-based method and SKR provide better artifact suppression. However, MASK provides even better artifact removal, while also providing excellent motion smoothness (removal of judder) due to temporal interpolation.

Table 2. Average PSNR (in dB) for Foreman video sequences compressed at different bit rates.

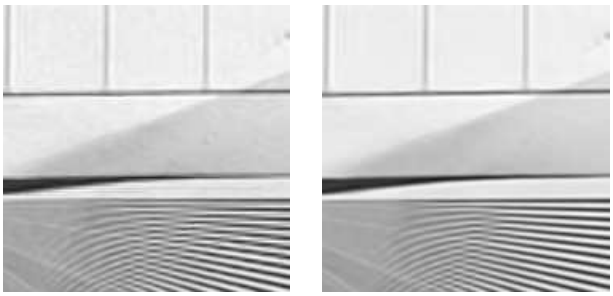
Algorithm	200 kbps	400 kbps	600 kbps
Bicubic (2-D)	25.29	25.70	25.84
TV (2-D)	27.63	28.36	28.61
SKR (2-D)	27.40	28.07	28.24
MASK (3-D)	28.27	29.16	29.44

(a) Bicubic (b) TV

(c) SKR (d) MASK

Fig. 4. Foreman results (200 kbps, frame 4, 256x256 cropped area).

In our third experiment, we performed spatial and temporal interpolation (by a factor 2) on cropped frames from the "Spin Calendar" sequence using MASK. This sequence contains rotational motion as well as real noise. Again, we applied motion estimation as in Section 3; also $N = 2$, $h^s = 1.0$, and temporal support is 5 frames for MASK. We visually compared the resulting video sequence with the result of spatial interpolation using 2-D SKR and frame repetition. Example frames are shown in Fig. 5. When seen as a video sequence, the results show that MASK provides better spatial reconstruction of the features in the image, as well as improved noise reduction. Also, the motion is very smooth due to temporal interpolation, with intermediate frames that appear to be artifact-free. Video files with these and our other results can be downloaded for viewing from http://www.soe.ucsc.edu/∼htakeda/MASK.



(a) SKR (b) MASK

Fig. 5. Spin Calendar results (frame 8, 180x180 cropped area).

5. CONCLUSION

In this paper, we introduced a spatio-temporal (3-D) steering kernel regression framework for video processing. We proposed a motion-assisted steering kernel (MASK) as well as a complete algorithm for multi-frame video interpolation and denoising. We presented several simulation results on synthetic and real video data, showing that MASK achieves significant visual and numerical improvement over 2-D methods. Our ongoing work includes: development of a multi-scale spatio-temporal regression algorithm; further comparison of MASK to existing multi-frame super-resolution reconstruction methods; and application of MASK to color video data. Future work also includes improving the computational efficiency.

6. REFERENCES

[1] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, February 2007.

[2] T. Q. Pham et al., "Robust fusion of irregularly sampled data using adaptive normalized convolution," *EURASIP J. on Appl. Sign. Proc.*, vol. 2006, Article ID 83268, pp. 1–12, 2006.

[3] H. Takeda, S. Farsiu, and P. Milanfar, "Robust kernel regression for restoration and reconstruction of images from sparse noisy data," in *Int. Conf. on Image Processing (ICIP 2006)*, Atlanta, GA, October 2006.

[4] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, May 2003.

[5] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *Int. Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.

[6] G. de Haan, "Video scanning and format conversion and motion estimation," in *The Digital Signal Processing Handbook*, V. K. Madisetti and D. B. Williams, Eds. CRC Press, 1998.

[7] C. Stiller and J. Konrad, "Estimating motion in image sequences - a tutorial on modeling and computation on 2-D motion," *IEEE Signal Processing Magazine*, vol. 16, no. 4, July 1999.

[8] T. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 231–241, February 2001.