Edge-Preserving Smoothing and Mean-Shift Segmentation of Video Streams

Sylvain Paris

Adobe Systems, Inc.

Abstract. Video streams are ubiquitous in applications such as surveillance, games, and live broadcast. Processing and analyzing these data is challenging because algorithms have to be efficient in order to process the data on the fly. From a theoretical standpoint, video streams have their own specificities – they mix spatial and temporal dimensions, and compared to standard video sequences, half of the information is missing, *i.e.* the future is unknown. The theoretical part of our work is motivated by the ubiquitous use of the Gaussian kernel in tools such as bilateral filtering and mean-shift segmentation. We formally derive its equivalent for video streams as well as a dedicated expression of isotropic diffusion. Building upon this theoretical ground, we adapt a number of classical algorithms to video streams: bilateral filtering, mean-shift segmentation, and anisotropic diffusion.

1 Introduction

This paper proposes a coherent approach to analyzing and filtering video streams. We develop tools that process video frames as soon as they are available and broadcast the result immediately. This scenario encompasses a wealth of practical applications such as surveillance, live preview, interactive simulations, and games. A major constraint imposed by these tasks is the requirement for instant results. For instance, latency cannot be tolerated in games – the displayed video must instantaneously follow the player inputs. This motivates our strict assumption that no future data are known when we process a frame; only the current and past data are available. In this context, we focus on edge-preserving smoothing and image segmentation because they are two techniques at the core of widely used applications such as vignetting correction [1], noise estimation and removal [2, 3, 4], object selection [5, 6], and stylization [7, 8, 9, 10].

Many video-processing tools are inherently off-line because they require the entire video sequence to be provided as input to the algorithm [2, 3, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]. Nonetheless, a few on-line methods exist [9, 10] but they often resort to a frame-by-frame approach. As a consequence, these techniques can be used only on high-quality data with stable filters such as the bilateral filter. Otherwise, temporal incoherence may appear. Intuitively, the frame-by-frame methods under-exploit the available information because they process each frame separately although all the previous frames are known. We address this issue by using past data when processing the current frame.

Our paper is based on a theoretical study that explicitly focuses on data smoothing. Although it is of limited interest when applied to pixel values because it blurs the image, smoothing is the basis of powerful tools. Bilateral filtering has been shown to

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part II, LNCS 5303, pp. 460–473, 2008.

be a Gaussian convolution in a higher-dimensional space [21]. Mean-shift segmentation is known to be driven by a Gaussian kernel [22, 14] while anisotropic diffusion based on partial differential equations (PDE) can be seen as a refinement over isotropic diffusion [23]. These results motivate and structure our work: we first derive smoothing operators for video streams and then build upon them to obtain a set of new tools adapted to video streaming.

Overview and Contributions. First, we review related work and describe how isotropic diffusion and Gaussian convolution can be derived from gradient minimization in image space (Section 2). Then, we formally study video streams and characterize the specificity of the time axis as mixed boundary conditions. We build upon this study to extend two fundamental operators to video streams: isotropic diffusion and Gaussian convolution. We show that a first-order PDE in the temporal domain is the equivalent of the second-order PDE that defines isotropic diffusion in image space. We also demonstrate that an exponential decay over time corresponds to the Gaussian kernel classically used in image space. The theoretical contribution of our paper is the relationship between the temporal PDE and kernel, and their spatial counterpart (Section 3). With this result, we naturally extend a number of algorithms to video streams. We demonstrate anisotropic diffusion, bilateral filtering, and mean-shift segmentation (Section 4). These algorithms run in real time with low memory consumption and achieve temporal coherence on par with off-line methods (Section 5).

2 Background

2.1 Theoretical Scale-Space Studies

A few articles analyze spatio-temporal data from a scale-space standpoint [24, 25, 26, 27]. These papers define multiscale representation of images from various axioms such as non-enhancement of local extrema. Our theoretical results are related and we will discuss the links as they appear. But from an application perspective, these studies are concerned by the intrinsic structure of the data whereas we focus on typical computer vision objectives such as edge-preserving smoothing and image segmentation.

2.2 Video Applications

In this section, we review practical applications and categorize them according to how they deal with the temporal axis t.

Frame-by-Frame Techniques. These methods filter each frame as an isolated image. Formally, they use a Dirac peak centered on the current frame as temporal kernel. If the processing time is shorter than the delay between frames, then real-time video processing can be achieved as demonstrated by Winnemöller *et al.* [9] and Chen *et al.* [10] with bilateral filtering implemented on graphics hardware. The downside is that temporal coherence is not ensured because the filter has no knowledge of the adjacent frames. Only data with limited noise can be processed this way.

Spatio-Temporal Techniques. To ensure temporal coherence, spatio-temporal methods filter along the time axis in addition to the x and y dimensions. *Omniscient* approaches process a given frame assuming past and future data to be known whereas *causal* techniques rely only on past data.

Omniscient methods process a frame using both past and future data. One approach is to handle the t axis as another spatial dimension and process the data as a *video volume* [17] (also known as *video cube* [28]). These techniques involve a 3D Gaussian kernel [12,14] or a PDE [17] dealing similarly with space (xy) and time (t). Another option is to differentiate space and time. This strategy is often used for denoising [2, 3, 20, 16] to favor temporal filtering over spatial smoothing in order to preserve image sharpness. All omniscient techniques are inherently limited to off-line processing since the last frame must be known before the algorithm starts. Moreover, with long sequences, the amount of data to handle can be arbitrarily large, requiring specific memory management techniques such as tiling and out-of-core processing.

Causal methods are the most related to ours since they filter a given frame using only past data. Kim and Woods [29], and Patti *et al.* [30] use Kalman filtering to aggregate data over time. But, because of the ordering imposed by the Kalman filter, pixels have an asymmetric spatial neighborhood which can incur visible defects. Bennett and McMillan [31] use an exponential decay over time to display motion trails in time-lapse videos. For bilateral filtering, Chen *et al.* [10] also apply an exponential decay to remove artifacts due to their sampling strategy.

A contribution of our paper is to motivate the temporal exponential decay in a general context and to apply it to several applications. In comparison, previous work [31, 10] introduces it as a heuristic for specific cases and limit its use to a single task.

2.3 Background on Image Processing

Smoothing an image I aims for reducing its variations. This can be expressed as reducing the norm of the gradients of I, that is, as minimizing the integral $\int \left(I_x^2 + I_y^2\right)$ where $I_x = \partial I / \partial x$. Isotropic diffusion and Gaussian convolution are two classical and equivalent ways to achieve this minimization.

Isotropic Diffusion. One option to minimize $\int \left(I_x^2 + I_y^2\right)$ is to apply an iterative descent scheme. That is, I evolves such that the gradients are progressively reduced. The "evolution time" is represented by a variable e. Intuitively, each iteration transforms $I(e)$ into $I(e + \delta e)$ which gradients are smaller. This process can be formalized by considering infinitesimal steps δe. The image evolution is then defined by the derivative $I_e = \partial I / \partial e$. Assuming this quantity known, a possible implementation of the filter is iterating $I(e + \delta e) = I(e) + \delta e \times I_e$. The rest of this section explains how to use the Euler-Lagrange formula to define I_e.

For an energy of the form $\int \Phi\left(\alpha, f, f_\alpha\right) \mathrm{d}\alpha$ where α is a scalar, f a function of α, and Φ a function of α, f, and $f_\alpha = \partial f / \partial \alpha$, the following partial derivative equation (PDE) is satisfied at stationary points: $\frac{\partial \Phi}{\partial f} - \frac{\partial}{\partial \alpha}\left(\frac{\partial \Phi}{\partial f_\alpha}\right) = 0$. Using this property, an evolution scheme is defined to progressively transforms f to minimize $\int \Phi$. The variable e denotes the "evolution time". At $e = 0$, f is not transformed, and f evolves as e increases. The

evolution is defined by the derivative of f with respect to e:

$$f_e = \frac{\partial}{\partial \alpha}\left(\frac{\partial \Phi}{\partial f_\alpha}\right) - \frac{\partial \Phi}{\partial f} \tag{1}$$

For multiple independent variables $\alpha_1, \alpha_2...$, the equation is extended by adding corresponding second-order terms.

The above Euler-Lagrange formula is applied with $\Phi\left(..., f_{\alpha_1}, f_{\alpha_2}\right) = f_{\alpha_1}^2 + f_{\alpha_2}^2$ and $f(\alpha_1, \alpha_2) \equiv I(x, y)$. Only the second-order terms are non-zero, leading to isotropic diffusion (*a.k.a.* the heat equation): $I_e = I_{xx} + I_{yy}$.

Gaussian Convolution. Using $G(\sigma) = \left(2\pi\sigma^2\right)^{-1} \exp\left(-(x^2+y^2)/2\sigma^2\right)$ for the Gaussian kernel and $\otimes$ for the convolution operator, $G(\sqrt{2e}) \otimes I$ is known to be the only solution of the heat equation [23, 32]. As a consequence, Gaussian convolution is equivalent to isotropic diffusion and also minimizes the gradient norms.

Summary. The following equation summarizes the relationship between gradient minimization, Gaussian convolution, and the heat equation.

$$\text{minimize } \int \left(I_x^2 + I_y^2\right) \tag{2a}$$

$$G(\sigma) \otimes I \qquad \text{with: } \sigma = \sqrt{2e} \tag{2b}$$

$$I_e = I_{xx} + I_{yy} \tag{2c}$$

Equations 2b and 2c are equivalent and are derived from 2a. In practice, a finite value of σ and e is selected in order not to produce a constant (*i.e.*, infinitely smooth) image, thereby not fully minimizing (2a). One of our contributions is to derive a similar relationship for video streams.

3 Isotropic Diffusion and Gaussian Convolution for Video Streams

Gaussian convolution and isotropic diffusion are classical image-space operators upon which a number of powerful tools are built. For instance, anisotropic diffusion is an extension of isotropic diffusion and the bilateral filter has been shown to be a Gaussian convolution in the intensity-space domain [21]. In this section, we derive equivalent operators for video streams. First, we discuss the specificity of the time axis and then derive new operators that minimize the video-stream gradients.

3.1 Time Axis in Video Streams

We consider a frame of a video stream V at time t_0. All the previous frames have already been processed, that is, we know $V(t)$ for all $t < t_0$. In the video-streaming context, data are displayed or broadcast immediately after being processed and thus cannot be modified. We use the notation $\overline{V}(t)$ for these processed, fixed data at $t < t_0$. Conversely, future frames at $t > t_0$ are unknown and cannot be accessed. These properties make the time dimension fundamentally different from the x and y axes. For instance, when processing a pixel $V(x)$, both $V(x-\epsilon)$ and $V(x+\epsilon)$ can be accessed and modified.

Mixed Boundary Conditions. When processing the current data $V(t)$, we treat the past data $\overline{V}(t_0 - \epsilon)$ with $\epsilon > 0$ as hard constraints, that is, we have Dirichlet boundary conditions on the negative side. On the positive side, we cannot compute any derivative since $V(t_0 + \epsilon)$ is unknown. It means that right derivatives will never appear, or equivalently, be set to 0. Thus, we have Neumann boundary conditions on the positive side.

Discrete Modeling. A continuous model in which t is continuous variable and V a smooth variable is not a suitable approach to cope with the streaming setting. By continuity of V, we could write $V(t) = \lim_{\epsilon \to 0} \overline{V}(t - \epsilon)$ that would enforce the current frame to be always equal to the previous frame which is already known. In a continuous setup, we would always process data on the domain boundary where constraints are expressed. Therefore, although continuous modeling can be useful for image-space analysis [23], we prefer a discrete approach that lets us work at a distance from the domain boundaries. Such a choice has also been done in scale-space studies [26, 27].

3.2 Minimizing the Gradients

Gaussian convolution (Eq. 2b) and isotropic diffusion (Eq. 2c) cannot be applied directly to video streams because they do not take the temporal structure of video streams into account. For instance, we cannot use a Gaussian kernel along the time axis because it requires the use of future frames which are not available. Nevertheless, we can still formulate our goal as minimizing variations, *i.e.* gradient norms. We seek a scheme that minimizes $\int \left(V_x^2 + V_y^2 + cV_t^2\right)$ where c is a constant that weights the temporal metric versus the spatial metric. The x and y components are not affected by the video-stream structure, *i.e.* Equation 2 still holds. In the following discussion, we omit these terms for clarity and focus on the temporal term.

As discussed in the previous section, we use a discrete formulation for the time dimension, *i.e.* we minimize $\sum c\,[V(t) - \overline{V}(t-1)]^2$ where $t \in \mathbb{Z}$ is an integer indexing the frames and $V(t) - \overline{V}(t-1)$ is a backward difference, the discrete equivalent to a left derivative with the major difference that it involves the data V and the boundary constraint $\overline{V}$. The forward difference cannot be computed since the frame $t+1$ is unknown at time t. Recall also that at a given time t, the past data $\overline{V}(t-1)$ are fixed and that only the current frame $V(t)$ is processed.

Discrete Euler-Lagrange. With $\alpha \in \mathbb{Z}$ an integer variable, f a function of the variable α, $\Delta_\alpha f(\alpha) = f(\alpha+1) - f(\alpha)$ the discrete derivative of f, and Φ a function of α, f, and $\Delta_\alpha f$, the Euler-Lagrange formula becomes:

$$f_e \;=\; \Delta_\alpha\!\left(\frac{\partial \Phi}{\partial(\Delta_\alpha f)}\right) \;-\; \frac{\partial \Phi}{\partial f} \tag{3}$$

This discrete formula is known to be equivalent to the continuous one (Eq. 1). We refer to the work of Guo *et al.* [33] for a detailed formal proof.

Minimizing Temporal Variations. The temporal term $V(t) - \overline{V}(t-1)$ is not a standard discrete derivative because it involves $\overline{V}$ which is a boundary condition, only $V(t)$ is unknown and can vary. We use the notation $\overline{\Delta}_t V(t-1) = V(t) - \overline{V}(t-1)$. As a consequence, $\sum c\,[\overline{\Delta}_t V(t-1)]^2$ is a zeroth-order term, and the Euler-Lagrange formula (Eq. 3) results in a first-order scheme $V_e = -c\,\overline{\Delta}_t V$.

Diffusion in Video Streams. With the spatial terms and using Δ^2 for the discrete second derivative, we obtain the equivalent of the heat equation (Eq. 2c) for video streams:

$$V_e \;=\; \Delta_x^2 V \;+\; \Delta_y^2 V \;-\; c\,\overline{\Delta}_t V \tag{4}$$

This formula defines isotropic diffusion for video streams. The structure of the t axis induces a major change in the diffusion process with a first-order term instead of the second-order term classically found along spatial dimensions.

Intuition The temporal term acts as an attachment term. It adds to current values a portion of the difference with the previous data. The more we apply the diffusion equation, the more it moves the current data closer the values of the previous frame.

3.3 Integrating the Diffusion Equation

To obtain the equivalent of the Gaussian kernel, we integrate the diffusion equation (4) against the evolution time e. For clarity, we use the notation $V(t,e)$ from now on. For instance, $V(t,0)$ is the input frame at time t.

Spatial Dimensions. The spatial part of Equation 4 is similar to the image-space heat equation (Eq. 2c). As a consequence, the spatial component of the video-stream kernel is a Gaussian $G(\sqrt{2e})$ similarly to the image-space case.

Temporal Dimension. We focus on the temporal part: $V_e = -c\,[V(t,e) - \overline{V}(t-1,e)]$ (Eq. 4). It is a first-order differential equation with V as unknown since $\overline{V}(t-1,e)$ is constant. By imposing the value at $e=0$, the solution is:

$$V(t,e) = \overline{V}(t-1,e) + \exp(-c\,e)[V(t,0) - \overline{V}(t-1,e)] \tag{5}$$

With $q = 1-\exp(-c\,e)$, we rewrite Equation 5 to express the output at t as a function of the input at t and the output at $t-1$:

$$V(t,e) \;=\; (1-q)\,V(t,0) \;+\; q\,\overline{V}(t-1,e) \tag{6}$$

We recursively apply Equation 6 at time $t-s$ for $s>0$ to remove the dependency on the past output frames. Since $q<1$ when $e>0$, we have $\lim_{s\to+\infty} q^s = 0$ and:

$$V(t,e) \;=\; (1-q)\,\sum_{s\geq 0} q^s\,V(t-s,0) \tag{7}$$

Equation 7 corresponds to a convolution between the input frames $V(\cdot,0)$ and a truncated exponential decay D:

$$V(t,e) \;=\; \big[D \;\otimes\; V(\cdot,0)\big](t) \tag{8a}$$

$$\text{with}\quad D : t \mapsto k\,\exp(-t/\lambda)\,H(t) \tag{8b}$$

where $k = 1-q = \exp(-c\,e)$ is a normalization constant, H the Heaviside function ($H(t) = 1$ if $t \geq 0$ and 0 otherwise), and λ obtained by comparing two successive frames: $\lambda = -1/\log\big(1-\exp(-c\,e)\big)$.

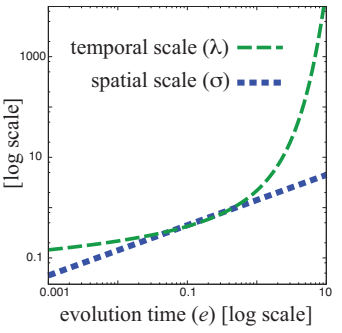


Fig. 1. Spatial and temporal scales as functions of the evolution time e for $c = 1$.

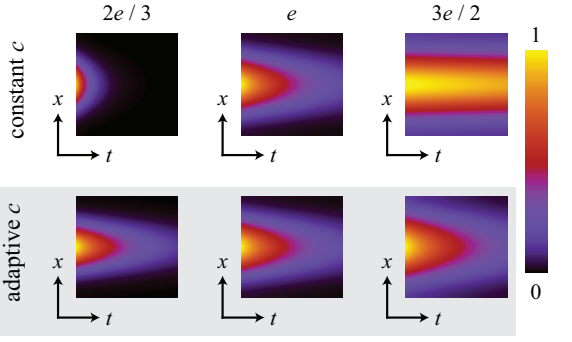


Fig. 2. Variation of the kernel shape with the evolution time e. Without adjustment, the temporal scaling is more important than the spatial one (top row). We can enforce a uniform scaling by adapting the metric parameter c (bottom row).

Scale. The σ and λ parameters respectively control the spatial and temporal scales of the kernel. Their behavior with the respect to the evolution time e is different as illustrated in Figure 1. When using the diffusion operator (Eq. 4), varying e does not uniformly scale the spatio-temporal kernel: the t dimension is more altered than the xy axes (Fig. 2-top). Nonetheless, the metric constant c can be adjusted to obtain a uniform scaling. For instance, one can define the scale σ and the aspect ratio $\gamma = \sigma/\lambda$. From Equation 2b, we get $e = \sigma^2/2$, and from the definition of λ: $c = -2\log(1 - \exp(\sigma/\gamma))/\sigma^2$. Using these settings, one can uniformly scale the kernel (Fig. 2-bottom).

Practical Recursive Algorithm. The spatio-temporal convolution can be made efficient since it is separable. Several techniques such as Deriche's recursive scheme [34] exist for the spatial Gaussian part. For the temporal part, we use the recursive formula (Eq. 6). It performs an exact computation based only on the current and last frames. This aspect makes our approach practical since only the current frame $V(t)$ and the previous one $\overline{V}(t-1)$ are needed at a given time. As a consequence, our algorithm has low memory requirements and the data always fit in memory independently of the video length. Figure 3 provides the pseudo-code for the convolution. In comparison, omni-

FASTSPATIOTEMPORALCONVOLUTION
· input: $V(t,0)$, $\overline{V}(t-1,e)$, e, c
· internal variables: q, $\tilde{V}$
· output: $V(t,e)$

1. Decay computation: $q = 1 - \exp(-ce)$
2. Space convolution: $\tilde{V} = G(\sqrt{2e}) \otimes V(t,0)$
3. Time recursion: $V(t,e) = (1-q)\tilde{V} + q\overline{V}(t-1,e)$

Fig. 3. Pseudo-code of the spatio-temporal convolution

scient methods are either limited to short sequences on the order of a few seconds or resort to using complex memory management strategies such as tiling. The differential formulation (Eq. 4) enjoys the same low-memory benefits.

3.4 Summary and Discussion

We have shown that gradients of video streams are minimized by a PDE that extends the classical second-order heat equation with a first-order temporal term. After integration, this PDE is equivalent to a spatio-temporal kernel made of a spatial Gaussian and a temporal exponential decay. The difference between the spatial dimensions and the time axis stem from the previously processed frames and the missing future data which impose boundary constraints onto the smoothing process. The following relationship and Figure 4 summarize this result.

$$\text{minimize} \sum \left[(\Delta_x V)^2 + (\Delta_y V)^2 + (\overline{\Delta}_t V)^2 \right] \tag{9a}$$

$$G(\sigma) D(\lambda) \otimes V \tag{9b}$$

$$V_e = \Delta_x^2 V + \Delta_y^2 V - c\overline{\Delta}_t V \tag{9c}$$

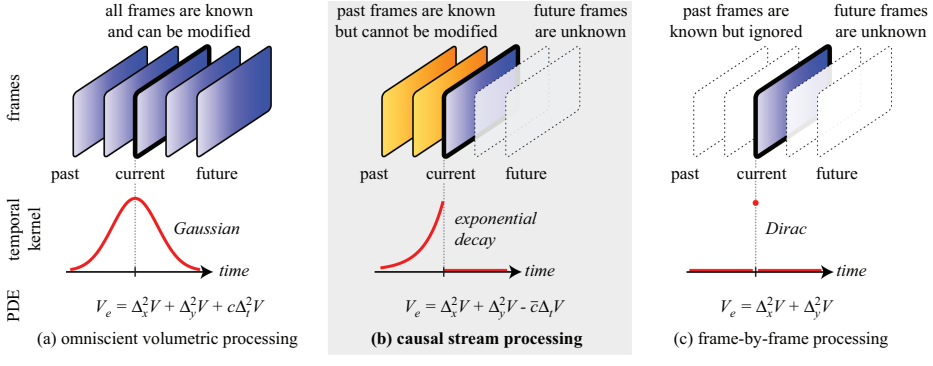


Fig. 4. Our approach (b) uses past data to ensure temporal coherence unlike frame-by-frame techniques (c) that rely only the quality of the input and the stability of the filter to achieve coherence. Compared to omniscient techniques (a), we do not assume that the future frames are known, thereby enabling on-line processing.

Link to Existing Work. The spatio-temporal PDE and kernel has been described by Lindeberg using scale-space axioms [26, 27]. Our contribution is to derive them from a simple smoothing problem, *i.e.* gradient minimization. The other major aspect of our work is to extend this result to edge-preserving smoothing and image segmentation as described in the following section.

4 Applications

Using the previously defined PDE and kernel directly on the pixel values is of limited interested because it blurs the video content. For instance, a naive approach would be

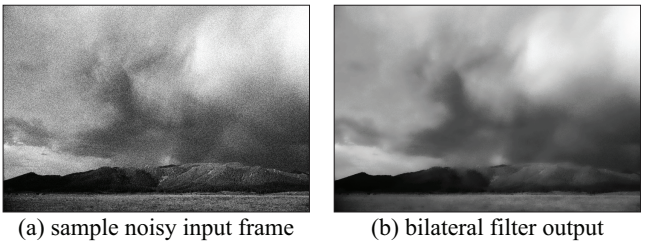
to apply temporal smoothing as a post-process to an edge-preserving filter to enforce temporal coherence. This produces blurry results as shown in the companion video. Nonetheless, smoothing can be used to build several useful tools which demonstrate the value of our theoretical study. Videos are provided in the supplemental material.

4.1 Bilateral Filtering

The bilateral filter [35, 36, 37] is an edge-preserving filter that has proven to be an effective tool in computational photography [38]. For an image I, the output at a pixel $\mathbf{p}$ is the normalized average of the adjacent pixels $\mathbf{q}$ weighted by two Gaussian functions, $G(\sigma_\mathrm{s}, \cdot)$ and $G(\sigma_\mathrm{i}, \cdot)$, accounting for the spatial and intensity distances respectively:

$$\frac{1}{W} \sum_{\mathbf{q}} G\big(\sigma_\mathrm{s}, \|\mathbf{p} - \mathbf{q}\|\big)\, G(\sigma_\mathrm{i}, |I(\mathbf{p}) - I(\mathbf{q})|)\, I(\mathbf{q})$$

Our adaption to video streams is based on the result by Paris and Durand who expressed this filter as a Gaussian convolution in the space-intensity domain [21]. We build upon the result of the previous section and adapt the bilateral filter to video streams by convolving with our spatio-temporal kernel instead of the original Gaussian. Figure 5 shows a sample result produced by our algorithm.



(a) sample noisy input frame (b) bilateral filter output

Fig. 5. Bilateral filter result. The companion video demonstrates the stability of our approach in presence of noise.

4.2 Mean-Shift Segmentation

Mean shift is a method to segment images made popular by Comaniciu and Meer [22]. Each pixel is associated to a feature point and the produced segmentation can be seen as the modes of the density of feature points estimated with a Gaussian kernel [22, 14]. We extend this approach to video streams by estimating the feature-point density with our causal spatio-temporal kernel. With this approach, modes span space and time and temporal coherence is naturally achieved (Fig. 6).

4.3 PDE-Based Anisotropic Diffusion

A large body of work exists on smoothing images with anisotropic diffusion using PDEs [23]. From a high-level standpoint, one can interpret these filters as variants of

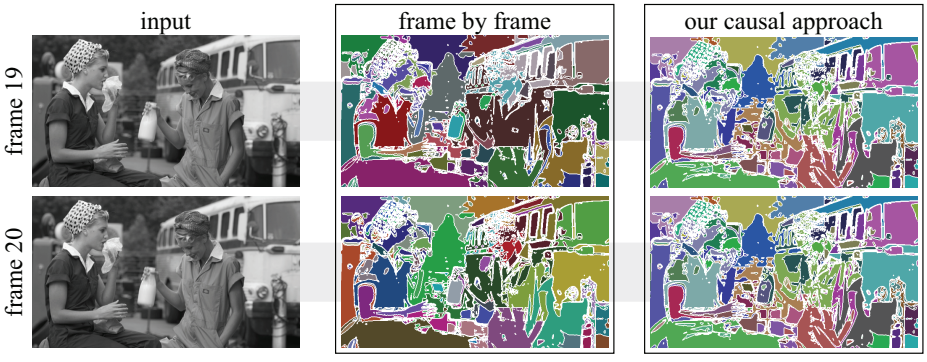
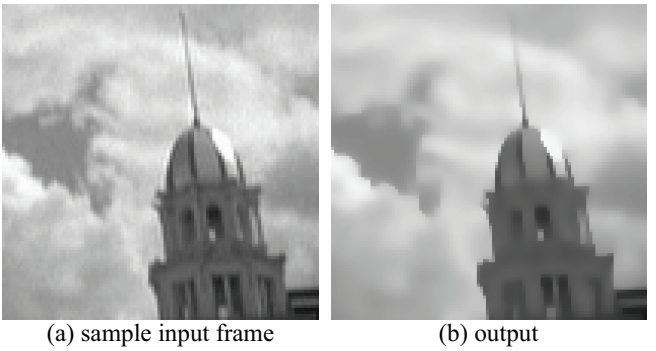


Fig. 6. Mean-shift segmentation. The clusters are color-coded. The frame-by-frame approach produces new clusters at each frame (indicated by new colors). A heuristic to "link" clusters across frames would not be satisfying since boundaries are not coherent. In comparison, our algorithm achieves a coherent segmentation. This is better seen in the companion video.

the heat equation (Eq. 2c) that steer the diffusion to avoid blurring the main image contours. We demonstrate our approach on the Perona-Malik filter [39]. In image space, this filter is similar to the heat equation with the gradients weighted by a stopping function $g(\cdot)$: $I_e = \mathrm{div}(g(\|\nabla I\|)\nabla I)$ where *div* is the divergence $\Delta_x + \Delta_y$. Based on our result (Eq. 9c), we propose the following equivalent for video streams:

$$V_e \;=\; \mathrm{div}(g(\|\nabla V\|)\nabla V) \;-\; c\, g(\|\nabla V\|)\overline{\Delta}_t V$$

The divergence term is equivalent to the image-space case. The last term is specific to video streaming. A sample output is shown in Figure 7.



(a) sample input frame (b) output

Fig. 7. Close-up on anisotropic diffusion results. We applied a strong smoothing effect to make it more visible. The full sequence is provided in supplemental material.

4.4 Possible Extensions

In the following sections, we discuss uses of our approach that requires studies beyond the scope of this paper. We plan to pursue these directions as future work.

Poisson Reconstruction. This technique is commonly used to reconstruct in the least-square sense an image from an approximated gradient field (u, v, w) [40, 41, 42]. For video streams, the least-square problem is: $\left[(\Delta_x V - u)^2 + (\Delta_y V - v)^2 + (\overline{\Delta}_t V - w)^2\right]$, which can be solved similarly to Equation 9a with the discrete Euler-Lagrange formula: $V_e = \Delta_x^2 V - \Delta_x u + \Delta_y^2 V - \Delta_y v - \overline{\Delta}_t V + w$. The first four terms correspond to the standard Poisson equation while the last two are specific to video streaming.

We experimented with this equation on panorama stitching [42], pasting [41], and Retinex [40]. At best, we found only minor improvements compared to frame by frame, while the lack of motion compensation induces defects at moving boundaries with Retinex. Unfortunately, optical-flow techniques have limitations that make them unsuitable for video streaming as discussed in the following paragraph.

Optical Flow. In general, we could benefit from optical flow to account for the scene motion. However, high-accuracy algorithms are computationally expensive and fast methods such as the one used in MediaPlayer have limited precision [43]. Thus, the current state of the art in motion estimation does not allow for stream processing yet. However, we believe that it is a promising research direction for the future. Real-time and accurate flow computation would have many applications. In our case, it would allow for steering the temporal component $\overline{\Delta}_t V$ according to the optical flow, thereby yielding results at boundaries as accurate as the ones in uniform areas.

5 Results

Setup. We have implemented the described applications in C++, compiled them with GCC 4.0.1 with optimization turned on, and run the tests on an Intel Xeon 3GHz. The Gaussian part of the kernel is implemented using a downsampled 5-tap approximation. The bilateral filter and the mean-shift segmentation are set with $\sigma_\mathrm{s} = 32$ and $\sigma_\mathrm{i} = 2.5$ considering that the intensity range spans $[0..100]$. Anisotropic diffusion uses axis-projected gradients as suggested by Perona and Malik [39]. Frame rates reported in Table 1 are averaged over 100 frames, not including disk operations. Memory consumption excludes system libraries. Videos are provided in the supplemental material.

Performance. Our approach yields memory efficient algorithms that achieves real-time or near real-time performance on gray-level videos at DVD resolution while maintaining temporal coherence (Table 1). There is no limit on the video length, the reported

Table 1. Frame rate and memory consumption. Input stream is 640×360 in gray levels.

	frame-by-frame	our approach
bilateral filter	29.1Hz (3.8MB)	28.4Hz (3.9MB)
mean shift	59.5Hz (1.1MB)	52.9Hz (1.3MB)
Perona-Malik	14.6Hz (8.8MB)	11.7Hz (9.7MB)

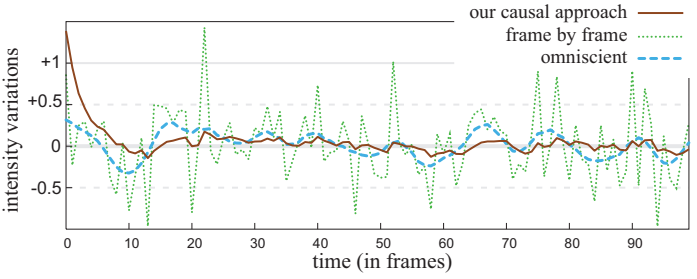


Fig. 8. Temporal stability of the Perona-Malik filter. Our approach exhibits a transient start but rapidly achieves more stable results than frame-by-frame and even omniscient methods because it relies on the past frames which have already been smoothed. We added noise up to $\pm 5\%$ of the intensity range. All filters have been applied with $e = 2$.

performance and memory usage can be sustained for an arbitrary duration. As future work, we plan to leverage graphics hardware to process color data.

Stability. Temporal stability is a key issue, especially in uniform areas where incoherence yields undesirable flickering. We evaluated the stability of our approach compared to frame-by-frame and omniscient methods using the Perona-Malik filter. We processed 100 uniform frames with uncorrelated noise added and kept the evolution time e fixed to ensure a fair comparison. Figure 8 shows the intensity plot of the center pixel. As expected, frame-by-frame results lack coherence. Our causal approach exhibits a transient start because there is no past data available. Then, it becomes slightly more stable than the omniscient output. This property stems from our use as hard constraints of the past data which are already smoothed whereas the omniscient process exploits more data but smoothes them at the same time, *i.e.* in the first iterations, past and future frames are still noisy and not as useful as the smoothed past frame in our case.

Discussion. Compared to omniscient algorithms which are inherently limited to off-line processing, our approach enables on-line tasks. Nonetheless, if on-line processing is not required, omniscient approaches offer more flexibility such as multi-pass analysis [12] since they work in a less constrained scenario. Compared to frame-by-frame results, the computation and memory overhead is limited (Table 1). For bilateral filtering and anisotropic diffusion, our tests showed that our approach yields more stable results on noisy images. Chen *et al.* [10] used this property to further speed up their hardware implementation of the bilateral filter. Our contribution is to formally motivate the technique that Chen described as a heuristic. For mean-shift segmentation, unlike other approaches, our method can coherently process arbitrary long videos.

6 Conclusions

We described a formal approach to processing video streams. We did not assume any knowledge about the future data and relied on the past frames to ensure temporal coherence. This translates into boundary constraints on the time axis which we used to

derive smoothing operators dedicated to video streaming. Based on this result, we revisited bilateral filtering, anisotropic diffusion, and mean-shift segmentation, and demonstrated that they can be applied on video streams in real time. Our tests show that our causal approach is suitable for applications such as coherent denoising for surveillance and real-time segmentation for on-line analysis. In particular, it should be preferred to frame-by-frame methods for noisy data and segmentation tasks.

Acknowledgement. The author thanks his Adobe and MIT colleagues for their feedback.

References

1. Zheng, Y., Lin, S., Kang, S.B.: Single-image vignetting correction. In: Proc. of the conf. on Computer Vision and Pattern Recognition, vol. 1, pp. 461–468. IEEE, Los Alamitos (2006)
2. Bennett, E.P., McMillan, L.: Video enhancement using per-pixel virtual exposures. ACM Transactions on Graphics 24, 845–852 (2005); Proc. of the ACM SIGGRAPH conf.
3. Bennett, E.P., Mason, J.L., McMillan, L.: Multispectral bilateral video fusion. IEEE Transactions on Image Processing 16, 1185–1194 (2007)
4. Liu, C., Freeman, W.T., Szeliski, R., Kang, S.: Noise estimation from a single image. In: Proc. of the Computer Vision and Pattern Recognition Conf. IEEE, Los Alamitos (2006)
5. Li, Y., Sun, J., Shum, H.Y.: Video object cut and paste. ACM Transactions on Graphics 24, 595–600 (2005); Proc. of the ACM SIGGRAPH conf.
6. Wang, J., Bhat, P., Colburn, R.A., Agrawala, M., Cohen, M.F.: Video cutout. ACM Transactions on Graphics 24 (2005); Proc. of the ACM SIGGRAPH conf.
7. Bae, S., Paris, S., Durand, F.: Two-scale tone management for photographic look. ACM Transactions on Graphics 25, 637–645 (2006); Proc. of the ACM SIGGRAPH conf.
8. DeCarlo, D., Santella, A.: Stylization and abstraction of photographs. In: Proc. of the ACM SIGGRAPH conf. (2002)
9. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-time video abstraction. ACM Transactions on Graphics 25, 1221–1226 (2006); Proc. of the ACM SIGGRAPH conf.
10. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. ACM Transactions on Graphics 26 (2007); Proc. of the ACM SIGGRAPH conf.
11. Wang, J., Xu, Y., Shum, H.Y., Cohen, M.F.: Video tooning. ACM Transactions on Graphics 23, 294–302 (2004); Proc. of the ACM SIGGRAPH conf.
12. Wang, J., Thiesson, B., Xu, Y., Cohen, M.F.: Image and video segmentation by anisotropic mean shift. In: Proc. of the European Conf. on Computer Vision (2004)
13. Bousseau, A., Neyret, F., Thollot, J., Salesin, D.: Video watercolorization using bidirectional texture advection. ACM Transactions on Graphics 26 (2007); Proc. of the ACM SIGGRAPH conf.
14. Paris, S., Durand, F.: A topological approach to hierarchical segmentation using mean shift. In: Proc. of the IEEE conf. on Computer Vision and Pattern Recognition (2007)
15. Buades, A., Coll, B., Morel, J.M.: A non local algorithm for image denoising. In: Proc. of the conf. on Computer Vision and Pattern Recognition (2005)
16. Chen, J., Tang, C.K.: Spatio-temporal Markov random field for video denoising. In: Proc. of the IEEE conf. on Computer Vision and Pattern Recognition (2007)
17. Drori, I., Leyvand, T., Fleishman, S., Cohen-Or, D., Yeshurun, H.: Video operations in the gradient domain. Technical report, Tel-Aviv University (2004)
18. Chuang, Y.Y., Agarwala, A., Curless, B., Salesin, D., Szeliski, R.: Video matting of complex scenes. ACM Transactions on Graphics 21 (2002); Proc. of the ACM SIGGRAPH conf.

19. DeMenthon, D.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. In: Proc. of the Statistical Methods in Video Processing Workshop (2002)
20. Zitnick, C.L., Jojic, N., Kang, S.B.: Consistent segmentation for optical flow estimation. In: Proc. of the International Conf. on Computer Vision (2005)
21. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. In: Proc. of the European Conf. on Computer Vision (2006)
22. Comaniciu, D., Meer, P.: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis Machine Intelligence 24, 603–619 (2002)
23. Aubert, G., Kornprobst, P.: Mathematical problems in image processing: Partial Differential Equations and the Calculus of Variations. Applied Mathematical Sciences, vol. 147. Springer, Heidelberg (2002)
24. Koenderink, J.J.: Scale-time. Biological Cybernetics 58 (1988)
25. ter Haar Romeny, B.M., Florack, L.M.J., Nielsen, M.: Scale-time kernels and models. In: Proc. of the conf. on Scale-Space and Morphology in Computer Vision (2001)
26. Lindeberg, T., Fagerström, D.: Scale-space with causal time direction. In: Proc. of European Conf. on Computer Vision (1996)
27. Lindeberg, T.: Linear spatio-temproal scale-space. In: Proc. of the International Conf. on Scale-Space Teory in Computer Vision (1997)
28. Klein, A., Sloan, P.P., Finkelstein, A., Cohen, M.F.: Stylized video cubes. In: Proc. of the ACM SIGGRAPH Symposium on Computer Animation (2002)
29. Kim, J., Woods, J.W.: Spatiotemporal adaptive 3-D Kalman filter for video. IEEE Transactions on Image Processing 6 (1997)
30. Patti, A.J., Tekalp, A.M., Sezan, M.I.: A new motion-compensated reduced-order model Kalman filter for space-varying restoration of progressive and interlaced video. IEEE Transactions on Image Processing 7 (1998)
31. Bennett, E.P., McMillan, L.: Computational time-lapse video. ACM Transactions on Graphics 26 (2007); Proc. of the ACM SIGGRAPH conf.
32. Koenderink, J.J.: The structure of images. Biological Cybernetics 50 (1984)
33. Guo, H.Y., Li, Y.Q., Wu, K.: Difference discrete variational principle, Euler-Lagrange cohomology and symplectic, multisymplectic structures. ArXiv Math. Physics e-prints (2001)
34. Deriche, R.: Recursively implementating the Gaussian and its derivatives. Technical Report RR-1893, INRIA (1993)
35. Aurich, V., Weule, J.: Non-linear gaussian filters performing edge preserving diffusion. In: Proc. of the DAGM Symposium (1995)
36. Smith, S.M., Brady, J.M.: SUSAN – a new approach to low level image processing. International Journal of Computer Vision 23, 45–78 (1997)
37. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. of the International Conf. on Computer Vision, pp. 839–846. IEEE, Los Alamitos (1998)
38. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: A gentle introduction to bilateral filtering and its applications. In: Course at the ACM SIGGRAPH conf. (2007)
39. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions Pattern Analysis Machine Intelligence 12, 629–639 (1990)
40. Horn, B.K.P.: Determining lightness from an image. Computer Graphics and Image Processing 3 (1974)
41. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on Graphics 22 (2003); Proc. of the ACM SIGGRAPH conf.
42. Levin, A., Zomet, A., Peleg, S., Weiss, Y.: Seamless image stitching in the gradient domain. In: Proc. of the European Conf. on Computer Vision (2006)
43. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M., Szeliski, R.: A database and evaluation methodology for optical flow. In: Proc. of the International Conf. on Computer Vision (2007)