EUROGRAPHICS 2011 / M. Chen and O. Deussen
(Guest Editors)

*Volume 30 (2011), Number 2*

# Image and Video Abstraction by Coherence-Enhancing Filtering[*]

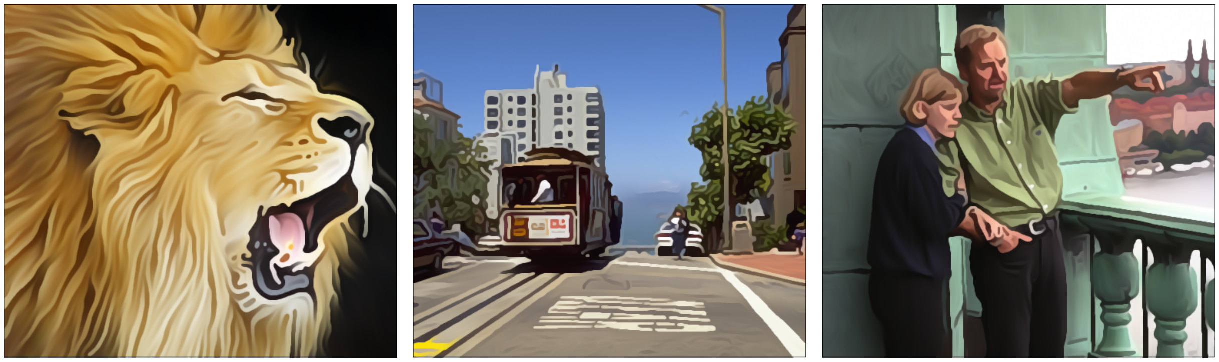
Jan Eric Kyprianidis[1] Henry Kang[2]

[1] Hasso-Plattner-Institut, Germany
[2] University of Missouri, St. Louis, USA

**Abstract**

*In this work, we present a non-photorealistic rendering technique to create stylized abstractions from color images and videos. Our approach is based on adaptive line integral convolution in combination with directional shock filtering. The smoothing process regularizes directional image features while the shock filter provides a sharpening effect. Both operations are guided by a flow field derived from the structure tensor. To obtain a high-quality flow field, we present a novel smoothing scheme for the structure tensor based on Poisson's equation. Our approach effectively regularizes anisotropic image regions while preserving the overall image structure and achieving a consistent level of abstraction. Moreover, it is suitable for per-frame filtering of video and can be efficiently implemented to process content in real-time.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms



## 1. Introduction

Directional features and flow-like structures are considered pleasant, harmonic, or at least interesting by most humans [Wei99]. They are also a highly sought-after property in many of the traditional art forms, such as paintings and illustrations. Enhancing directional coherence in the image helps to clarify region boundaries and features. As exemplified by Expressionism, it also helps to evoke mood or ideas and even elicit emotional response from the viewer [Wik10]. Particular examples include van Gogh and Munch, who have emphasized these features in their paintings. In this work, we present a new image and video abstraction technique that places emphasis on enhancing the directional coherence of features. The most notable related work in this category is image abstraction and stylization based on partial differential equations (PDE), in particular, shape-simplifying image abstraction by Kang and Lee [KL08] and Weickert's coherence-enhancing shock filter [Wei03]. However, such PDE-based techniques may require a large number of iterations and tend to be unstable when used for video processing [Par08].

[*] The definitive version is available at diglib.eg.org and www.blackwell-synergy.com.

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

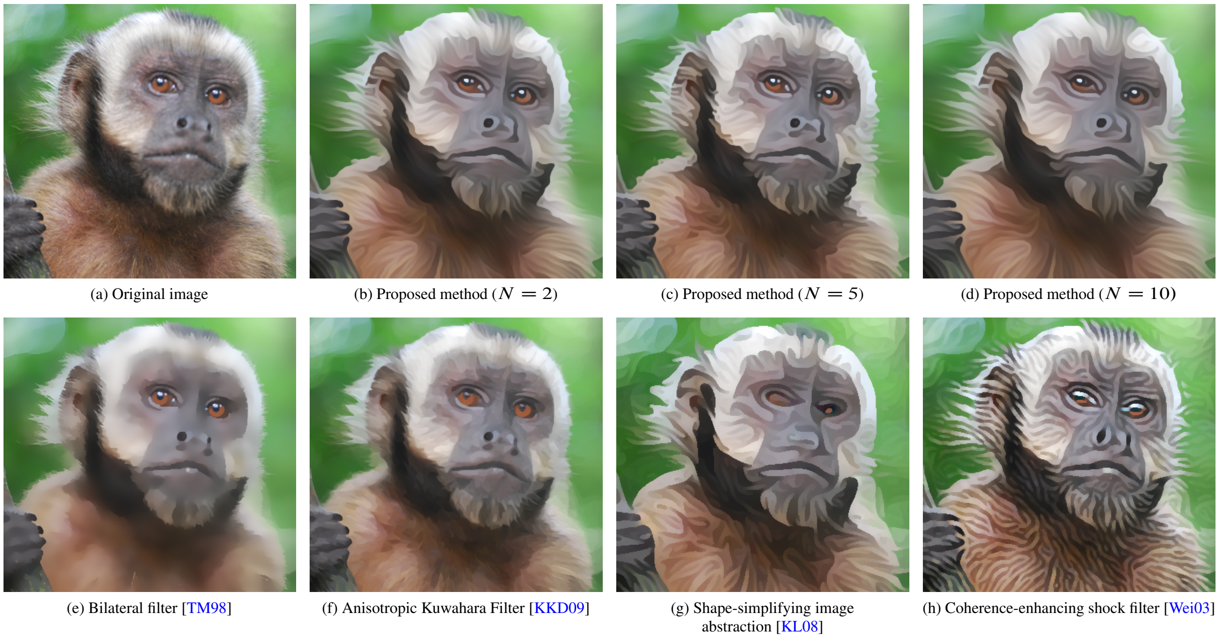

(a) Original image
(b) Proposed method ($N = 2$)
(c) Proposed method ($N = 5$)
(d) Proposed method ($N = 10$)
(e) Bilateral filter [TM98]
(f) Anisotropic Kuwahara Filter [KKD09]
(g) Shape-simplifying image abstraction [KL08]
(h) Coherence-enhancing shock filter [Wei03]

**Figure 1:** *Comparison of our approach with other popular image abstraction techniques. Top row shows results of the algorithm for different number of iterations ($N = 2, 5, 10$).*

We build upon the idea of combining diffusion with shock filtering for image abstraction, but our approach is, in a sense, contrary to that of [KL08], which our technique outperforms in terms of speed, temporal coherence and stability. Instead of simplifying the shape of the image features, we aim to preserve the shape by using a curvature preserving smoothing method that enhances coherence. More specifically, our approach performs smoothing, in the direction of the smallest change, and sharpening, in the orthogonal direction. Instead of modeling this process by a PDE and solving it, we use approximations that operate as local filters on a neighborhood of a pixel. Therefore, good abstraction results are already achieved in a few iterations. This makes it possible to process images and video at real-time rates on a GPU. It also results in a much more stable algorithm that enables temporally-coherent video processing. Compared to the conventional abstraction approaches [WOG06, OBBT07, KKD09], we provide a good balance between the enhancement of directional features and the smoothing of isotropic regions. As shown in Figure 1, our technique preserves and enhances directional features better and creates stronger contrast, which helps to clarify boundaries and features. Furthermore, our approach facilitates easy control over the level of abstractions.

## 2. Related Work

A common approach to image abstraction is segmentation. A classical example is the work by DeCarlo and Santella [DS02], where eye-tracking data is used to guide image abstraction based on mean shift segmentation at different scales.

Another popular approach to image abstraction is the use of edge-preserving smoothing and enhancement filters. Techniques of this type commonly remove detail in low-contrast regions without filtering across discontinuities and, thus, leave the overall structure of the input image unaffected. A well-known example is the bilateral filter [TM98]. Winnemöller et al. [WOG06] combine bilateral filtering with color quantization and difference of Gaussians edges, to create cartoon-style abstractions from images and videos. Kyprianidis and Döllner [KD08] extend this approach and present separable implementations of the bilateral and difference of Gaussians filters that are aligned to the local image structure. Kang et al. [KLC09] present a similar system, where the filter shapes of the bilateral and difference of Gaussians filters are deformed to follow a vector field derived from the salient image features. Since methods based on the bilateral filter preserve high-contrast edges, they generally fail for high-contrast images where either no abstraction is performed or too much detail is removed. This typically results in an inconsistent abstraction.

The Kuwahara filter [KHEK76] is another popular edge-preserving filter. It produces clearly noticeable artifacts due to the use of rectangular subregions. In addition, the subregion selection process is unstable if noise is present or if the subregions have the same variance. This results in randomly chosen subregions and corresponding artifacts. Several attempts have been made to address the limitations of the Kuwahara filter. Papari et al. [PPC07] define a new criterion to overcome the limitations of the unstable subregion selection process and replace the rectangular subregions by smooth weighting functions defined over sectors of a disc. The anisotropic

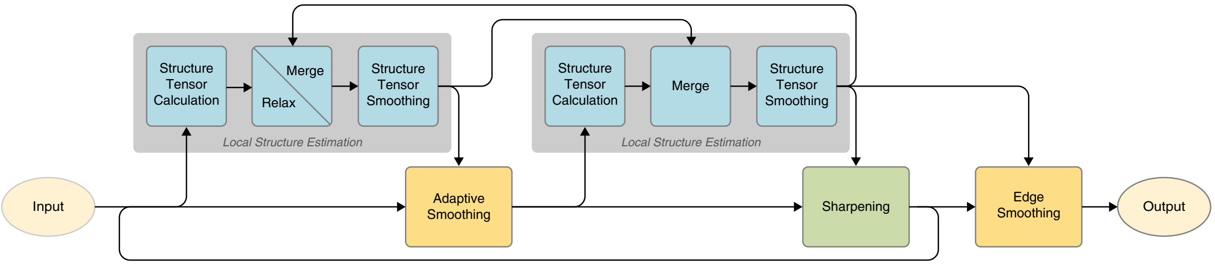

Structure Tensor Calculation
Merge
Relax
Structure Tensor Smoothing
Local Structure Estimation
Structure Tensor Calculation
Merge
Structure Tensor Smoothing
Local Structure Estimation
Input
Adaptive Smoothing
Sharpening
Edge Smoothing
Output

**Figure 2:** *Schematic overview of the proposed method.*

Kuwahara filter [KKD09] builds upon this and replaces the weighting functions, defined over sectors of a disc, by weighting functions defined over sectors of ellipses. By adapting shape, scale and orientation of these ellipses to the local structure of the input, artifacts are avoided and directional image features are better preserved and emphasized. The anisotropic Kuwahara filter is very robust against noise and stable against small changes in the input. This makes it especially useful for video abstraction where per-frame processing achieves excellent results.

A further image abstraction technique has been presented by Kang and Lee [KL08]. Their approach is based on mean curvature flow in conjunction with shock filtering. Methods based on edge-preserving filters, such as the bilateral or the Kuwahara filter, smooth irrelevant color variations while protecting region boundaries, but they do not simplify the shape of those boundaries. In contrast, mean curvature flow simplifies isophote curves and regularizes the geometry. Since mean curvature flow does not properly protect directional image features, Kang and Lee constrain the mean curvature flow. Mean curvature, and its constrained variant, contract isophote curves to points [Gra87]. For this reason, important image features must be protected by a user-defined mask. A further limitation is that the technique is not stable against small changes in the input and, therefore, not suitable for per-frame video processing.

An image abstraction technique, based on image processing in the gradient domain, has been presented by Orzan et al. [OBBT07]. The technique is known not to create temporal coherent output for video [BZCC10]. Bhat et al. [BZCC10] present a robust optimization framework that allows for the specification of constraints for pixel values and pixel gradients. The framework is able to create temporal coherent video output, but optical flow is required and a global optimization problem for the entire video must be solved. Therefore, the technique is not suitable for real-time processing.

## 3. Method

Processing is performed iteratively and stops after a user-defined number of iterations. The number of iterations controls the strength of the abstraction (Figure 1(b)-(d)). A schematic overview of the algorithm is shown in Figure 2. Processing starts with an input image, typically an RGB color image. Next, adaptive smoothing and sharpening are performed for the given number of iterations. Both techniques require information about the local structure, which is obtained by an eigenvalue analysis of the smoothed structure tensor. The local structure is computed, twice, for every iteration. It is computed before the adaptive smoothing and before the sharpening. With every iteration the result becomes closer to a piecewise constant image with large smooth or even flat regions where no distinguished orientation is defined. Having valid orientations defined for these regions is important for the stability of the algorithm. Our solution is to use the structure tensor from the previous calculation in this case. For the first calculation, where no result from a previous computation is available, a relaxation of the structure tensor is performed. Finally, edges are smoothed using the adaptive smoothing with a small standard deviation.

### 3.1. Local Structure Estimation

The structure tensor is also known as the second moment matrix and it is a popular tool in image processing and computer vision. A systematic treatment can be found in [BBL*06].

#### 3.1.1. Structure Tensor Calculation

The calculation of the structure tensor requires approximations of the partial derivatives. Popular choices for calculating these approximations are Gaussian derivatives and the Sobel filter. Motivated by the results in [WS02], we use the rotational symmetric derivative filter proposed in [JSK99]

$$D_x = \frac{1}{2}\begin{pmatrix} p_1 & 0 & -p_1 \\ 1-2p_1 & 0 & 2p_1-1 \\ p_1 & 0 & -p_1 \end{pmatrix}, \qquad D_y = D_x^T,$$

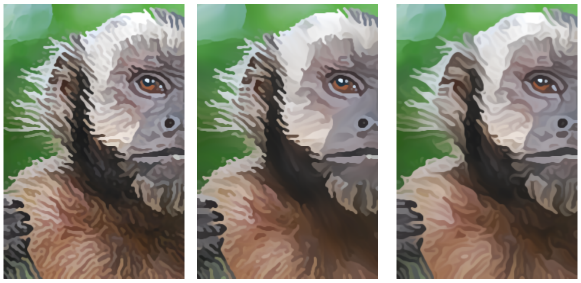
with $p_1 = 0.183$. Let $\Omega \subset \mathbb{Z}^2$ and let $f : \Omega \longrightarrow \mathbb{R}^3$ denote the input image. Then, approximations of the partial derivatives in $x$- and $y$-direction are given by convolution with the corresponding filter stencils:

$$f_x = D_x \star f \qquad f_y = D_y \star f$$

The structure tensor is now defined by

$$S = \begin{pmatrix} \langle f_x, f_x \rangle & \langle f_x, f_y \rangle \\ \langle f_x, f_y \rangle & \langle f_y, f_y \rangle \end{pmatrix} =: \begin{pmatrix} E & F \\ F & G \end{pmatrix},$$

where $\langle , \rangle$ denotes the scalar product. This is a positive semidefinite symmetric matrix and the induced quadratic



(a) No smoothing (b) Relaxation ($\tau_r = 0.002$) (c) Gaussian smoothing ($\sigma_d = 1$)

**Figure 3:** *Comparison of different smoothing approaches for the structure tensor* ($N = 10$).

form measures the squared rate of change of $f$. By calculating the major and minor eigenvalues

$$\lambda_{1,2} = \frac{E + G \pm \sqrt{(E-G)^2 + 4F^2}}{2}$$

we can find the directions of maximum and minimum change. Those are given by the corresponding eigenvectors:

$$\eta = \begin{pmatrix} F \\ \lambda_1 - E \end{pmatrix} \quad \text{and} \quad \xi = \begin{pmatrix} \lambda_1 - E \\ -F \end{pmatrix}$$

In the following, we will assume the eigenvectors $\eta$ and $\xi$ to be normalized. The major eigenvector $\eta$ is equivalent to the classical multi-channel image gradient [DZ86].

### 3.1.2. Structure Tensor Relaxation

In low-contrast regions, the signal-to-noise ratio is high and the gradient information is not reliable. Accordingly, the estimated orientation is almost random. Therefore, we replace structure tensors with low gradient magnitude using basic inpainting. Let

$$\partial\Omega = \left\{ x \in \Omega \;\middle|\; \sqrt{E(x)^2 + G(x)^2 + 2F(x)^2} > \tau_r \right\}$$

be the set of points whose structure tensors have a gradient magnitude greater than a threshold $\tau_r$. We are looking for a smooth function that interpolates the values of the boundary $\partial\Omega$. Such a function is given by the membrane that minimizes:

$$\underset{s}{\operatorname{argmin}} \int_\Omega |\nabla s|^2 \quad \text{with} \quad s|_{\partial\Omega} = S|_{\partial\Omega}$$

This problem is known to be equivalent to solving Laplace's equation with a corresponding Dirichlet boundary condition. A solution can be found, iteratively, by a series of Jacobi relaxation steps. Let $s^0_{i,j}$ denote the structure tensor at $(i,j)$, then a relaxation step is given by:

$$s^{k+1}_{i,j} = \begin{cases} s^k_{i,j} & \text{if } (i,j) \in \partial\Omega \\ \frac{s^k_{i+1,j} + s^k_{i-1,j} + s^k_{i,j+1} + s^k_{i,j-1}}{4} & \text{otherwise} \end{cases}$$

Since the computation involves a convex combination, the result is a positive semidefinite matrix and, thus, is well defined. Typically, a large number of iterations are required. In our implementation, we use a fast multigrid approach [BHM00], but performing the relaxation for every computation of the structure tensor is, nevertheless, expensive. Therefore, we perform the relaxation only for the first computation of the structure tensor. All subsequent computations use the structure tensor of the previous computation for points in $\partial\Omega$.

### 3.1.3. Structure Tensor Smoothing

The vector fields, that are defined by the eigenvectors of the structure tensor, are not smooth and not suitable for guiding the adaptive smoothing and sharpening. This is demonstrated in Figure 3(a). Furthermore, the smoothing achieved by the relaxation described above is not sufficient to create a smooth look (Figure 3(b)). A possible approach would be to smooth the image prior to derivative computation, thereby using Gaussian derivatives. A better approach is to apply smoothing directly to the structure tensor. Smoothing the structure tensor is more stable, since it corresponds to solving a weighted least square problem instead of taking a weighted average. Figure 3(c) shows the effect of smoothing the structure tensor with a Gaussian filter. Best results are achieved by combining relaxation with Gaussian smoothing as can be seen in Figure 1(d).

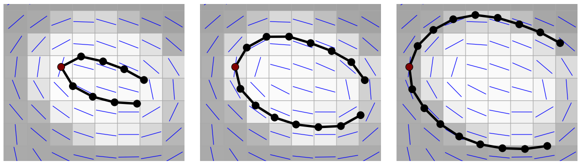
## 3.2. Adaptive Smoothing

To simplify the image, we perform line integral convolution [CL93] along the stream lines defined by the minor eigenvector field of the smoothed structure tensor. In contrast to the isophote curves of the image, the stream lines defined by the smoothed structure tensor are much smoother. The effect of this is that smoothing along the stream lines also results in a regularization of the geometry of the isophote curves. To achieve high-quality results, we perform the stream line integration using the second-order Runge-Kutta method. For pedagogical reasons, the simpler first-order Euler integration is also discussed.

### 3.2.1. Integration Scale

Controlling the amount of smoothing is important, since too much smoothing may destroy important image features. To adaptively control the amount of smoothing we use the following popular anisotropy measure [FG87], which is based on the eigenvalues of the structure tensor:

$$A = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} = \frac{\sqrt{(E-G)^2 + 4F^2}}{E + G}$$



(a) High curvature (b) Medium curvature (c) Low curvature

**Figure 4:** *Stream lines for different local curvature.*

The anisotropy $A$ ranges from 0 to 1, where 0 corresponds to isotropic and 1 corresponds to entirely anisotropic regions. The filter weights along the stream line are defined by a one-dimensional Gaussian function $G_{\tilde{\sigma}_s}$ with standard deviation

$$\tilde{\sigma}_s = \tfrac{1}{4}\sigma_s\,(1 + A)^2\,.$$

The global parameter $\sigma_s$ allows us to fine tune the smoothing. For all our examples, we use $\sigma_s = 6$. The adapted parameter $\tilde{\sigma}_s$ will lie in the range $\sigma_s/4 \leq \tilde{\sigma}_s \leq \sigma_s$. In regions of high curvature, the anisotropy will be low and, therefore, $\sigma_s$ small. In regions of low curvature, $\sigma_s$ will be large and result in an enhancement and regularization of directional image features (Figure 4).

### 3.2.2. Euler Integration

In order to perform line integral convolution, the stream line through the point to smooth must be found. A straightforward approach is to use the Euler integration method (Figure 5). Special attention needs to be paid to the sign of the minor eigenvector. The structure tensor defines only orientation, but no particular direction. This is due to the quadratic nature of the structure tensor. Therefore, the sign of the minor eigenvector is chosen, depending upon the direction of the previous step:

$$t_k^{\pm}(x) = \begin{cases} \pm\xi(x) & \text{if } k = 0 \\ \operatorname{sign}\langle t_{k-1}^{\pm}, \xi(x)\rangle \cdot \xi(x) & \text{otherwise} \end{cases}$$

Let $x_0^{\pm} = x_0$ denote the current point, then for $k \geq 0$ a step of the Euler method is given by:

$$x_{k+1}^{\pm} = x_k^{\pm} + t_k^{\pm}(x_k)$$

If $l$ denotes the cut-off of the discretized Gaussian, for example $l = \lceil 2\tilde{\sigma}_s \rceil$, then line integral convolution is defined as:

$$\frac{1}{\kappa}\left[ G_{\tilde{\sigma}_s}(0)\, f(x_0) + \sum_{k=1}^{l} G_{\tilde{\sigma}_s}(k)\left[ f(x_k^+) + f(x_k^-) \right] \right]$$

Here, $\kappa = \sum_{k=-l}^{l} G_{\tilde{\sigma}_s}(k)$ denotes the corresponding normalization factor and $f$ denotes the image.

### 3.2.3. Second-Order Integration

Instead of using the Euler integration method, we perform line integral convolution using the more precise second-order Runge-Kutta method. A step of the second-order Runge-Kutta method is given by:

$$x_{k+1}^{\pm} = x_k^{\pm} + t_k^{\pm}\Big( x_k^{\pm} + \tfrac{1}{2}\, t_k^{\pm}(x_k) \Big)$$

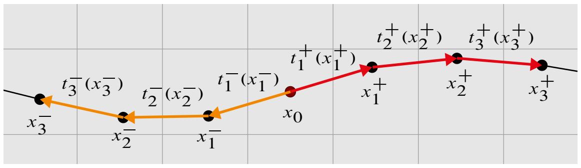
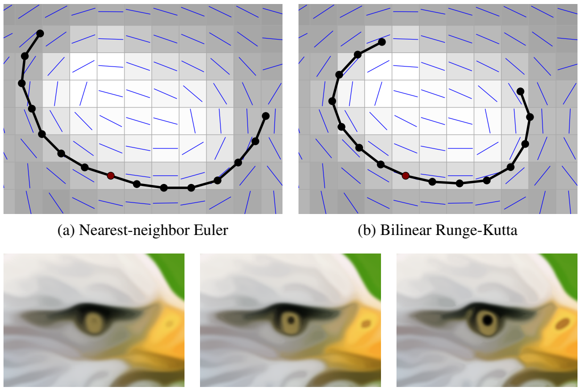


**Figure 5:** *Integration of the minor eigenvector field using Euler integration.*


(a) Nearest-neighbor Euler

(b) Bilinear Runge-Kutta

(c) Nearest-neighbor Euler

(d) Nearest-neighbor Runge-Kutta

(e) Bilinear Runge-Kutta

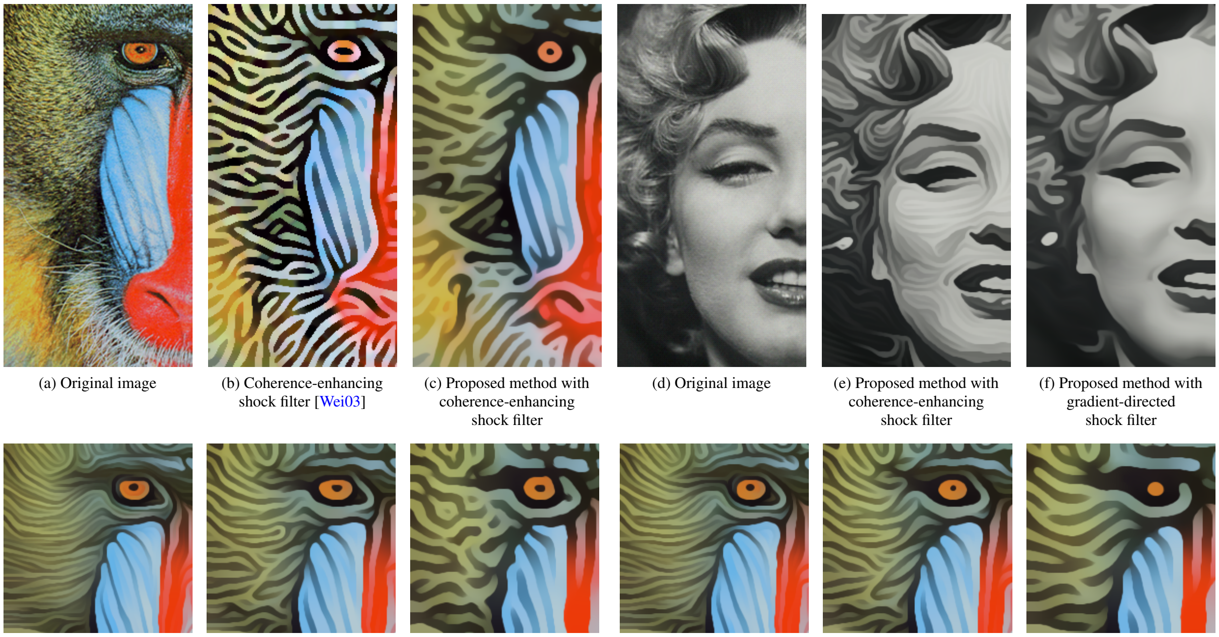
**Figure 6:** *Comparison of Euler and second-order Runge-Kutta structure tensor integration.*

In Figure 6, a comparison with the Euler integration method is shown. Since we use a large standard deviation, the stream lines used for convolution are very long. Therefore, accuracy of the integration method is very important. The Euler integration method smoothes pixels lying on adjacent isophote curves of the image (Figure 6(a)). This corresponds to smoothing in the direction of the major eigenvector. By contrast, the second-order Runge-Kutta method traces the stream line at a higher accuracy and does not blur across edges (Figure 6(b)).

The second-order Runge-Kutta method requires values of the minor eigenvector for arbitrary positions. One option is to calculate these in one pass and then use nearest-neighbor sampling in a second pass. Bilinear interpolation of the eigenvectors is complicated, since opposite vectors may cancel out. A better approach is to sample the structure tensor directly using bilinear interpolation. While this will produce a continuous vector field, it is more expensive since the minor eigenvector has to be computed for every sample. As can be seen in Figure 6(e), second-order Runge-Kutta integration with bilinear sampling of the structure tensor gives the best results and is used in our approach.

### 3.3. Sharpening

The adaptive smoothing is very aggressive. As can be seen in Figure 8, the overall shape of the image features is well preserved, but transitions between color regions are smooth. In order to obtain sharp transitions at edges, we perform deblurring by shock filtering [OR90].



(a) Original image
(b) Coherence-enhancing shock filter [Wei03]
(c) Proposed method with coherence-enhancing shock filter
(d) Original image
(e) Proposed method with coherence-enhancing shock filter
(f) Proposed method with gradient-directed shock filter

(g) $\sigma_i = 0, \sigma_g = 1$
(h) $\sigma_i = 0, \sigma_g = 2$
(i) $\sigma_i = 0, \sigma_g = 4$
(j) $\sigma_i = 1, \sigma_g = 1$
(k) $\sigma_i = 2, \sigma_g = 1$
(l) $\sigma_i = 4, \sigma_g = 1$

**Figure 7:** *Top row shows comparison of coherence-enhancing shock filter (*20 *iterations,* $\rho = 5$*,* $\sigma = 3$*,* $\tau = 0.4$*) and proposed method (*$N = 10$*). Bottom row shows additional combinations of isotropic and gradient directed smoothing parameters for the proposed method.*

### 3.3.1. PDE-based Shock Filter

Let $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ denote a grayscale image, then the classical shock filter evolution equation is given by

$$u_t = -\operatorname{sign}(\Delta u)|\nabla u| \ ,$$

with initial condition $u(x,0) = f(x)$. In the influence zone of a maximum, the Laplacian $\Delta u$ is negative and, therefore, a dilation is performed. In the influence zone of a minimum, the Laplacian $\Delta u$ is positive, which results in an erosion. This sharpens the edges at the zero-crossings of $\Delta u$. The Laplacian is very sensitive to noise and small detail. Alvarez and Mazorra [AM94], therefore, replace the Laplacian with a Laplacian of Gaussian (LoG).

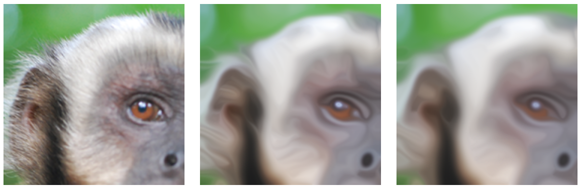
A further modification is the coherence-enhancing shock filter by Weickert [Wei03]. The sign of the Laplacian is replaced by the sign of the second derivative in the direction of the major eigenvector of the smoothed structure tensor:

$$u_t = -\operatorname{sign}(v_{\eta\eta})|\nabla u|$$

Here, $v = G_\sigma \star u$ denotes a smoothed version of the evolving image and $\eta$ is the major eigenvector derived from the structure tensor. In Figure 7(b) the coherence-enhancing shock filter applied to the popular mandrill image is shown. Clearly noticeable is the typical high-contrast of the output. Figure 7(c) shows the output of the proposed method, where the gradient-directed shock filter has been replaced by the coherence-enhancing shock filter. The color contrast is much more balanced and color regions have a consistent diffuse color variation. Weickert's shock filter achieves excellent results in combination with the proposed adaptive smoothing, but one limitation is its performance. The filter is typically implemented using the explicit Osher-Sethian [OS88] upwind scheme. In order to guarantee stability, the time step size has to be chosen $\leq 0.5$ and, therefore, multiple iterations have to be performed. We use a time step size of 0.4 and three iterations. For each iteration, first and second order derivatives, and the smoothed structure tensor, have to be calculated. Another limitation is shown in Figure 7(e). Weickert's shock filter will introduce shocks in almost smooth regions, resulting in maze-like artifacts.

### 3.3.2. Gradient-directed Shock Filter

As shown by Guichard and Morel [GM03], a shock filter can be approximated by either a minimum or a maximum filter that is chosen depending upon the sign of an associated edge detector. Inspired by the flow-based difference of Gaussians



(a) Original image
(b) 5 iterations
(c) 10 iterations

**Figure 8:** *Several iterations of the flow-guided smoothing without shock filtering.*

filter [KLC07], and its separable implementation [KD08], we calculate the sign using a one-dimensional (scale-normalized) Laplacian of Gaussian (LoG)

$$\sigma_g^2 G''_{\sigma_g}(t) = \sigma_g^2 \frac{\mathrm{d}^2 G_{\sigma_g}(t)}{\mathrm{d}t^2} = \frac{x^2 - \sigma_g^2}{\sqrt{2\pi}\sigma_g^3} \exp\left(-\frac{t^2}{2\sigma_g^2}\right)$$

in the direction of the minor eigenvector. The parameter $\sigma_g$ controls the size of the LoG, that is the resulting line thickness. To provide additional artistic control, we introduce a second smoothing parameter $\sigma_i$ that performs isotropic smoothing. Let $L$ be the input image converted to grayscale and let $v = G_{\sigma_i} \star L$, then

$$z(x_0) = \sigma_g^2 \int G''_{\sigma_g}(t)\, v\big(x_0 + t\,\eta(x_0)\big)\,\mathrm{d}t$$

defines the convolution with a one-dimensional LoG in the direction of the major eigenvector. In our implementation, we use a constant step size that has a unit size, along either the $x$ or the $y$ axis, to evaluate the integral. Sampling is performed using bilinear interpolation. This effectively results in linear interpolation of two neighboring pixels, due to the unit step size [KD08], and allows efficient implementation on GPUs using texturing hardware. Let $x_0$ denote the current point, then the gradient-directed shock filter is defined as:

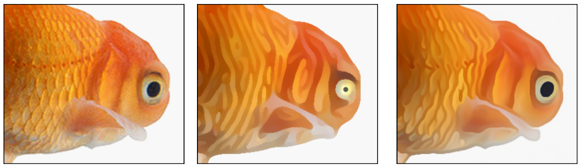
$$\begin{cases} \min_{x \in \Lambda_r(x_0)} f(x) & \text{if } z(x_0) > +\tau_s \\ \max_{x \in \Lambda_r(x_0)} f(x) & \text{if } z(x_0) < -\tau_s \\ f(x_0) & \text{otherwise} \end{cases}$$

Here, $f$ denotes the input image. Determination of the minimum and maximum is performed based on the corresponding gray values. The filter neighborhood $\Lambda_r$ with radius $r$ is defined by

$$\Lambda_r(x_0) = \left\{\, x \in \Omega \;\middle|\; \exists \lambda \in [-r, r] : x = x_0 + \lambda\,\eta(x_0) \,\right\}$$

and corresponds to all pixels, with a distance less than $r$ that intersect the line defined by the major eigenvector. For the examples, we typically use $r = 2$. The parameter $\tau_s$ controls the sensitivity to noise and is typically set to $\tau_s \in [0, 0.01]$. Since a scale-normalized LoG is used, $\tau_s$ does not depend upon $\sigma_g$. The threshold effectively prevents the creation of shocks in almost smooth regions, as can be seen in Figure 7(f).

The quality of the output is comparable to that of the coherence-enhancing shock filter, but computationally the gradient-directed shock filter is much more efficient. It only requires the smoothed structure tensor and the input image converted to grayscale. The directional LoG and minimum or maximum filters can be implemented in a single pass. Moreover, the gradient-directed shock filter provides finer artistic control. The parameter $\sigma_g$ restricts smoothing to the major eigenvector direction. This is especially useful for preserving small image features. To achieve a stronger abstraction, the isotropic smoothing parameter $\sigma_i$ is useful. In our examples, we typically use $\sigma_g = 1.5$ and $\sigma_i = 0$. Figures 7(g)-(l) show how the abstraction can be controlled using $\sigma_g$ and $\sigma_i$.



(a) Original image (b) [KL08] (c) Proposed method

**Figure 9:** *In contrast to shape-simplifying image abstraction, the proposed approach does not require manual masking to protect small image features such as eyes.*

### 3.3.3. Edge Smoothing

The shock filter creates very sharp transitions at region boundaries. To anti-alias the region boundaries, we apply the adaptive smoothing with a small standard deviation $\sigma_a \in [1, 1.5]$ as a final pass. In this case no adaption of the standard deviation is performed. We also do not recompute the structure tensor. Instead, we reuse the structure tensor that has been computed for the shock filter.

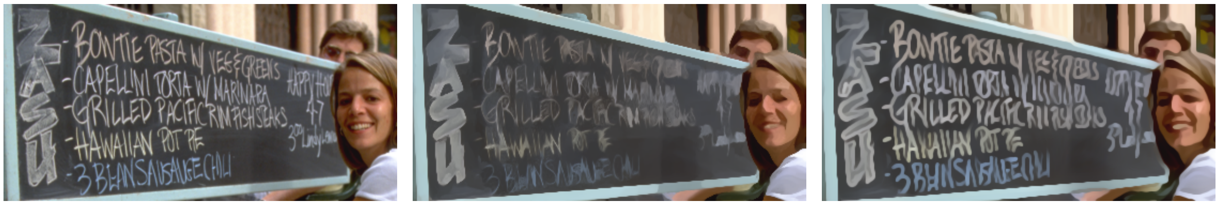
## 4. Implementation

We have implemented the proposed algorithm using C++ and CUDA. The implementation does not make use of special features of CUDA. Instead, we use pitch linear memory and textures. Therefore, similar results should be achievable using shading languages, such as GLSL, Cg or HLSL. Using an NVIDIA GTX 480 graphics card, processing an image with resolution 512 × 512 takes approximately 11 milliseconds for two iteration. This corresponds to an abstraction effect on the level of the anisotropic Kuwahara filter. Image content at HD 720p (1280 × 720) resolution takes approximately 31 milliseconds to process for two iterations. Hence, content at HD 720p resolution can be processed in real-time with more than 30 frames per second. For comparison, we have created a CUDA implementation of the anisotropic Kuwahara filter based on [KKD10]. Processing an image with resolution 512 × 512 takes 32 milliseconds. With resolution 1280 × 720 it takes 121 milliseconds. Our approach, thus, performs much better even though many computations are performed.

## 5. Results

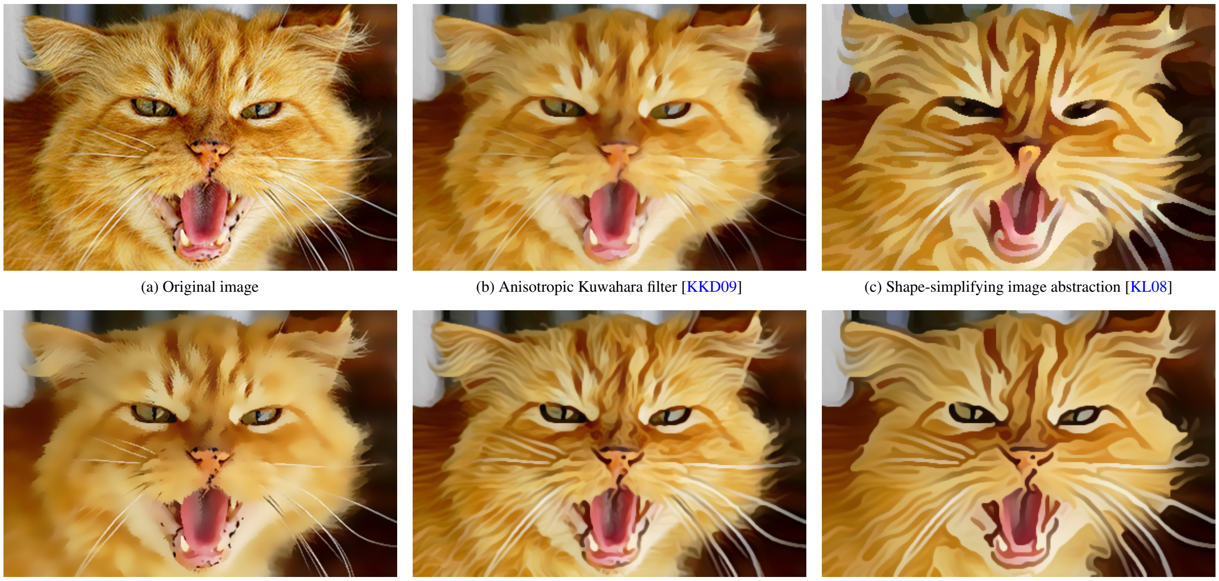
Figure 9 demonstrates the ability of our approach to preserve singular features like eyes. As shown in Figure 9(b), constrained mean curvature flow does not preserve curvature and contracts circular image features to points. By contrast, our adaptive smoothing preserves the curvature of image features and, therefore, does not require manual assistance, such as masking, to protect important image features. In addition to preserving curvature, the adaptive smoothing effectively enhances small highly anisotropic image features. This is demonstrated in Figure 10. By contrast, the anisotropic Kuwahara filter either erodes or blurs thin image features.

Further examples are shown in Figure 11. In contrast to the bilateral filter (Figure 11(d)), our approach creates a con-


-BONTIE PASTA W/ VEG & GREENS
-CAPELLINI TORTA W/ MARINARA
-GRILLED PACIFIC RIM FISH STEAKS
-HAWAIIAN POT PIE
-3 BEAN SAUSAGE CHILI

(a) Original image (b) Anisotropic Kuwahara filter [KKD09] (c) Proposed method

**Figure 10:** *In contrast to other techniques, our approach emphasizes and enhances highly anisotropic image features.*



(a) Original image (b) Anisotropic Kuwahara filter [KKD09] (c) Shape-simplifying image abstraction [KL08]

(d) Bilateral filter (4 iterations) [TM98] (e) Proposed method ($N = 2$) (f) Proposed method ($N = 10$)

**Figure 11:** *Comparison with anisotropic Kuwahara filter, shape-simplifying image abstraction, and bilateral filter.*

sistent abstraction across the entire image. The anisotropic Kuwahara filter (Figure 11(b)) also creates a consistent abstraction, but the result looks somewhat washed out because of missing contrast. For images with strong directional features, shape-simplifying image abstraction typically achieves very good results (Figure 11(c)). Our approach also performs excellently for these types of images and, by increasing the number of iterations, a strong abstraction effect can be obtained as well (Figure 11(f)). Figures 11(e) and 11(f) demonstrate the feasibility of our approach to control the strength of abstraction by changing the number of iterations. This is also claimed for shape-simplifying image abstraction, but only given from a certain number of iterations. For a small number of iterations, shape-simplifying image abstraction typically shows artifacts, which makes a light abstraction effect impossible. The anisotropic Kuwahara filter is limited to a light abstraction effect, since the amount of abstraction depends upon the filter radius. Using larger filter radii will also increase the amount of blurring and may destroy important image features.

Figure 12 demonstrates the stability of our approach. Here, different methods are applied to an image that has been artificially corrupted with Gaussian and impulse noise. Shape-simplifying image abstraction delivers the least satisfactory performance. Our approach performs decently, but is not able to remove the impulse noise. This is not surprising. Both filters are based on anisotropic smoothing, which will remove the Gaussian noise but also blur the impulse noise. Both techniques also use shock filtering that enhances noise. The anisotropic Kuwahara filter successfully restores the image, but the region boundaries are distorted. When the number of iterations is increased, our approach performs well. It converges to a nearly steady state that looks a little odd (Figure 12(h)), but it does not blow up like shape-simplifying image abstraction (Figure 12(g)).

In Figure 13, our approach is compared with techniques based on gradient domain image processing. In contrast to these techniques, our approach also performs a regularization of the image that results in smooth color boundaries. However, regions of the same color are not as smooth as the ones created by GradientShop. This is because the adaptive smoothing only performs smoothing in the tangential direction. On the other hand, this retains important visual detail, as for example, the specular reflection on the nose. Also note that, unlike our technique, both these techniques currently do not perform in real-time.

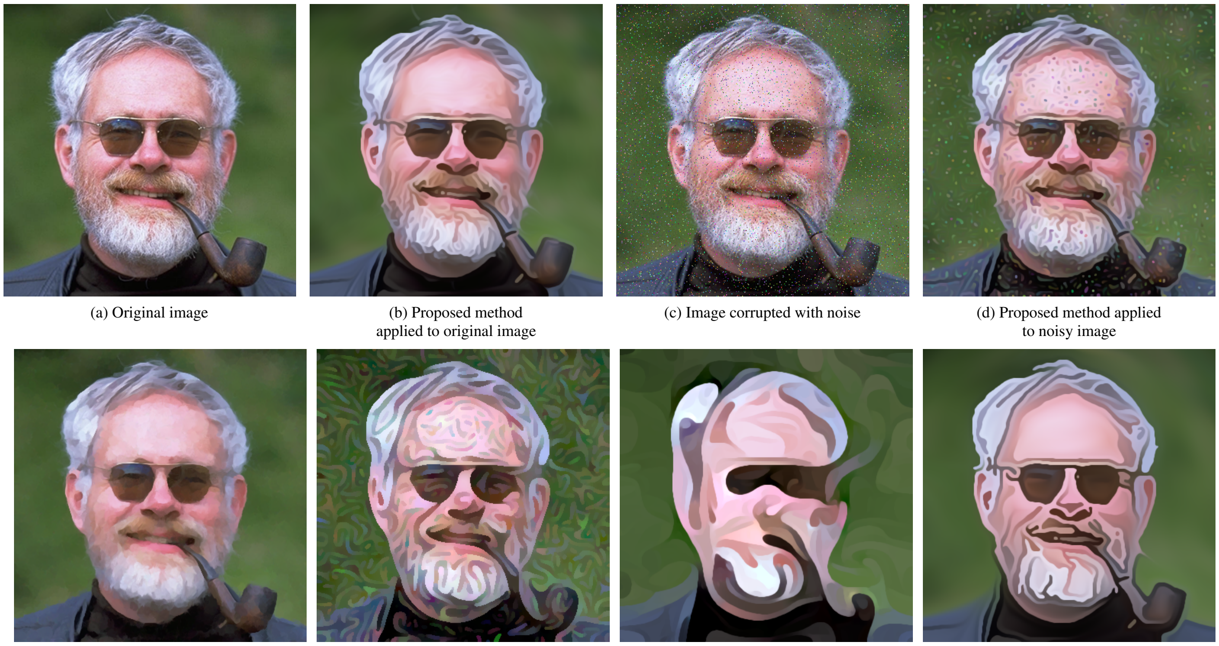

(a) Original image
(b) Proposed method applied to original image
(c) Image corrupted with noise
(d) Proposed method applied to noisy image
(e) [KKD09] applied to noisy image
(f) [KL08] applied to noisy image
(g) Limit case: [KL08] with $50$ CMCF iterations
(h) Limit case: proposed method with $N = 50$

**Figure 12:** *(a)-(f): Anisotropic Kuwahara filter, shape-simplifying image abstraction, and proposed method applied to an image corrupted with 2% Gaussian noise and 5% impulse noise. (g)-(h): Limit cases.*

An image that is difficult to handle for our approach is shown in Figure 14(a). Here, two prevailing orientations exist. One of the water, which is almost horizontal, and one of the contour of the rock. Since the gradient magnitude is high at the rock's contour, the orientation of the contour will dominate in a small neighborhood of the contour, during the structure tensor smoothing. Therefore, the adaptive smoothing will perform smoothing on both sides of the contour. This results in a halo-like effect, as shown in Figure 14(b). In Figure 14(c), an image with high contrast texture is shown. If processed by our method with typical parameters, the texture will be emphasized. To obtain the result in Figure 14(d), we had to use a large number of iterations and $\sigma_i = 5$ to prevent the shock filter from regarding the texture as edges. However, flow-like structures are still clearly observable and are, again, due to the directional smoothing.

Our approach can also be applied to video using per-frame filtering without extra processing. Very good temporal coherence is provided for 1-3 iterations. Applying more iterations typically results in temporal artifacts becoming noticeable. However, compared to shape-simplifying image abstraction or segmentation based approaches these are rather minor. We currently investigate ways to stabilize the method for larger number of iterations as part of our future work.

## 6. Conclusion

We have presented an automatic technique for image and video abstraction, based on adaptively controlled flow-guided smoothing and directional shock filtering. Our approach aggressively smoothes out unimportant regions, but protects important features by enhancing contrast and directional coherence, providing a good balance between content abstraction and feature enhancement consistently across the image. For abstraction at the level of the anisotropic Kuwahara filter, the GPU implementation of our method processes video at HD resolution in real-time and creates temporally coherent output without further processing.

## References

[AM94] Alvarez L., Mazorra L.: Signal and image restoration using shock filters and anisotropic diffusion. *SIAM Journal on Numerical Analysis 31*, 2 (1994), 590–605.

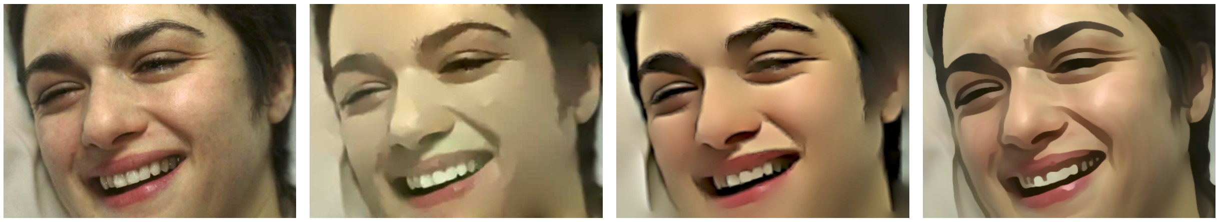
[BBL*06] Brox T., Boomgaard R., Lauze F., Weijer J., Weickert J., Mrázek P., Kornprobst P.: Adaptive structure tensors and their applications. *Visualization and Processing of Tensor Fields* (2006), 17–47.

[BHM00] Briggs W. L., Henson V. E., McCormick S. F.: *A Multigrid Tutorial*, 2 ed. Society for Industrial and Applied Mathematics, 2000.

[BZCC10] Bhat P., Zitnick C. L., Cohen M., Curless B.: Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics 29*, 2 (2010), 1–14.

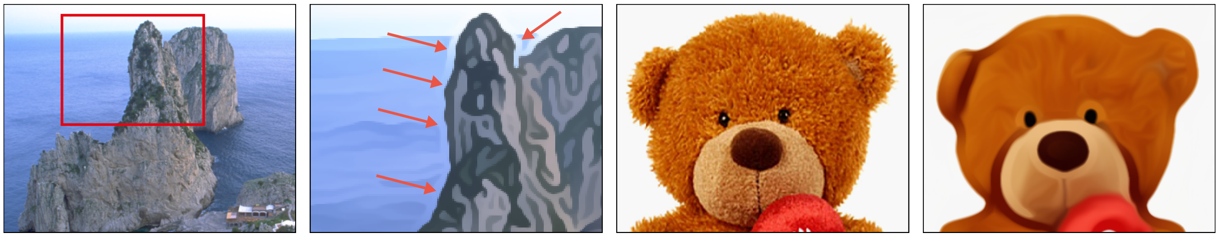
[CL93] Cabral B., Leedom L. C.: Imaging vector fields using line integral convolution. In *Proc. ACM SIGGRAPH* (1993), pp. 263–270.

[DS02] DeCarlo D., Santella A.: Stylization and abstraction of photographs. In *SIGGRAPH '02: Proceedings of the 29th an-*



(a) Original image (b) Structure-preserving manipulation of photographs [OBBT07] (c) GradientShop [BZCC10] (d) Proposed method ($N = 10$)

**Figure 13:** *Comparison with gradient domain processing techniques. ((a) - (c) courtesy [BZCC10])*



(a) Original image (b) Proposed method (c) Original image (d) Proposed method ($N = 30, \sigma_i = 5$)

**Figure 14:** *Examples of images that are difficult to handle for our approach.*

*nual conference on Computer graphics and interactive techniques* (2002), pp. 769–776.

[DZ86] Di Zenzo S.: A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing 33*, 1 (1986), 116–125.

[FG87] Förstner W., Gülch E.: A fast operator for detection and precise location of distinct points, corners and centers of circular features. In *Proc. of the ISPRS Intercommission Workshop* (1987), pp. 281–305.

[GM03] Guichard F., Morel J.: A note on two classical enhancement filters and their associated pde's. *International Journal of Computer Vision 52*, 2 (2003), 153–160.

[Gra87] Grayson M. A.: The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry 26*, 2 (1987), 285–314.

[JSK99] Jähne B., Scharr H., Körkel S.: Principles of filter design. In *Computer Vision and Applications*, vol. 2. Academic Press, 1999, ch. 6, pp. 125–151.

[KD08] Kyprianidis J. E., Döllner J.: Image abstraction by structure adaptive filtering. In *Proc. EG UK Theory and Practice of Computer Graphics* (2008), pp. 51–58.

[KHEK76] Kuwahara M., Hachimura K., Eiho S., Kinoshita M.: *Digital processing of biomedical images*. Plenum Press, 1976, pp. 187–203.

[KKD09] Kyprianidis J. E., Kang H., Döllner J.: Image and Video Abstraction by Anisotropic Kuwahara Filtering. *Computer Graphics Forum 28*, 7 (2009), 1955–1963.

[KKD10] Kyprianidis J. E., Kang H., Döllner J.: Anisotropic Kuwahara Filtering on the GPU . In *GPUPro - Advanced Rendering Techniques*, Engel W., (Ed.). AK Peters, 2010.

[KL08] Kang H., Lee S.: Shape-simplifying image abstraction. *Computer Graphics Forum 27*, 7 (2008), 1773–1780.

[KLC07] Kang H., Lee S., Chui C. K.: Coherent line drawing. In *Proc. NPAR* (2007), pp. 43–50.

[KLC09] Kang H., Lee S., Chui C. K.: Flow-based image abstraction. *IEEE Transactions on Visualization and Computer Graphics 15*, 1 (2009), 62–76.

[OBBT07] Orzan A., Bousseau A., Barla P., Thollot J.: Structure-preserving manipulation of photographs. In *Proc. NPAR* (2007), pp. 103–110.

[OR90] Osher S., Rudin L.: Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis 27*, 4 (1990), 919–940.

[OS88] Osher S., Sethian J.: Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Comp. Physics 79* (1988), 12–49.

[Par08] Paris S.: Edge-preserving smoothing and mean-shift segmentation of video streams. In *Proc. ECCV* (2008), pp. 460–473.

[PPC07] Papari G., Petkov N., Campisi P.: Artistic edge and corner enhancing smoothing. *IEEE Transactions on Image Processing 16*, 10 (2007), 2449–2462.

[TM98] Tomasi C., Manduchi R.: Bilateral filtering for gray and color images. In *Proc. International Conference on Computer Vision (ICCV)* (1998), pp. 839–846.

[Wei99] Weickert J.: Coherence-enhancing diffusion of colour images. *Image and Vision Computing 17*, 3 (1999), 201–212.

[Wei03] Weickert J.: Coherence-enhancing shock filters. *Pattern Recognition. Lect. Notes in Comput. Sc. 2781* (2003), 1–8.

[Wik10] Wikipedia: Expressionism – Wikipedia, The Free Encyclopedia, 2010. [Online; accessed 10-December-2010].

[WOG06] Winnemöller H., Olsen S. C., Gooch B.: Real-time video abstraction. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (2006), pp. 1221–1226.

[WS02] Weickert J., Scharr H.: A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. *Journal of Visual Communication and Image Representation 13* (2002), 103–118.

Original photographs in Figure 1(b)(c), 10(a), 12(a), 14(a) courtesy Phillip Greenspun. Photographs from flickr.com kindly provided under Creative Commons license by Ivan Mlinar (Figure 1(a)), Tambako the Jaguar (Figure 1(a)) and pasma (Figure 11(a)). Figure 7(a) courtesy USC-SIPI Image Database.