Image Abstraction by Structure Adaptive Filtering

Jan Eric Kyprianidis and Jürgen Döllner

Hasso-Plattner-Institut, University of Potsdam, Germany *

Abstract

In this work we present a framework of automatic non-photorealistic image processing techniques that create simplified stylistic illustrations from color images, videos and 3D renderings. To smooth low-contrast regions while preserving edges, we present a new fast separated implementation of the bilateral filter. Our approach works by filtering in direction of the gradient and then filtering the intermediate result in perpendicular direction. When applied iteratively, our approach does not suffer from horizontal or vertical artifacts and creates smooth output at curved boundaries. To extract salient important edges we first apply a one-dimensional difference-of-Gaussians filter in direction of the gradient and then apply smoothing along a flow field which we derive from the smoothed structure tensor. Our method creates smooth coherent output for line and curve segments.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation
I.3.3 [Computer Graphics]: Display algorithms

Figure 1: *Examples of abstracted images created using our framework.*

1 Introduction

Photorealistic visualizations used in 3D applications designed for typical desktops usually show high visual information density. In this paper we present non-photorealistic image processing techniques to distill the perceptually important information and optimize the content for the limited screen space of small displays. Our method (Figure 2) extends the approach of Winnemöller et al. [WOG06] to use iterated bilateral filtering for abstraction and difference-of-Gaussians (DoG) for edge extraction. We present enhancements to these techniques to improve the quality of the output by adapting them to the local orientation of the input. Our method is simple and efficient to implement on CPU and GPU. Implemented on the GPU our approach enables real-time processing of video and application as a post-processing effect for real-time 3D renderings.
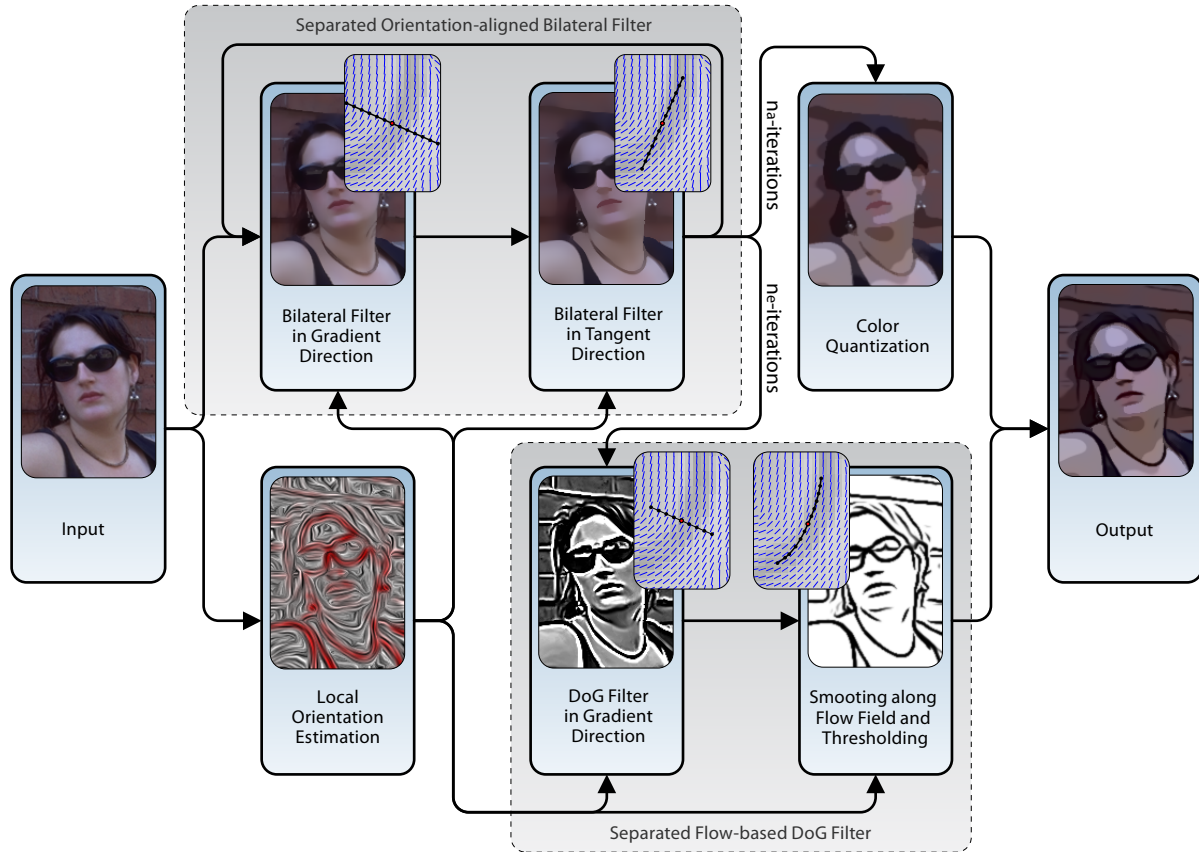
* {kyprianidis|doellner}@hpi.uni-potsdam.de

Figure 2: *Overview of our abstraction framework.*

To represent local orientation we construct a smooth tensor field. From the eigenvectors of this tensor field we derive a vector field that has similar characteristics as the edge tangent flow (ETF) of Kang et al. [KLC07], but its computation is much less expensive. Besides gradient calculation, only linear separable smoothing with a box or Gaussian filter is necessary. In contrast to that, ETF construction requires several iterations of a nonlinear filter with a large filter kernel.

The xy-separated bilateral filter [PvV05] used by Winnemöller et al. suffers from horizontal and vertical artifacts. These artifacts appear in particular when the filter is applied iteratively (Figure 6(c)). Pham presents in [Pha06] a uv-separated bilateral filter implementation that is aligned along the flow curves of the structure tensor. It gives good results even under severe noise conditions, but is not optimal when used with color quantization (Figure 6(d)). Our approach, which filters in direction perpendicular to the gradient instead of filtering along the flow curve, creates smooth output at curved boundaries (Figure 6(b)) and works well with color quantization.

DoG edges look frayed and don't reassemble straight line and curve segments very well (Figure 9). To work around this limitation Kang et al. [KLC07] recently introduced the con-cept of flow-based difference-of-Gaussians (FDoG) which, compared to DoG edges, create more coherent lines. They replaced the DoG filter by a flow-guided anisotropic filter kernel whose shape is defined by the ETF. In this work we show that comparable high-quality results can be achieved by a separated implementation with corresponding reduced computational complexity. As first pass, we apply a one-dimensional DoG filter in gradient direction followed by a second pass that applies smoothing along the flow curves of the vector field induced by the smoothed structure tensor. To integrate the flow curves, we present a straightforward-to-implement method which, compared to Euler integration and line integral convolution [CL93], produces more uniformly distributed sampling positions.

2. Related Work

The structure tensor appears in the corner detector of Harris and Stephens [HS88]. Kass and Witkin constructed in [KW87] a vector field to estimate local orientation which can be interpreted as a specialization of the structure tensor. Bigün et al. show in [BGW91] that the structure tensor can be seen as an approximation to fitting a plane to the Fourier transform of an n-dimensional structure. Van Vliet and Ver-
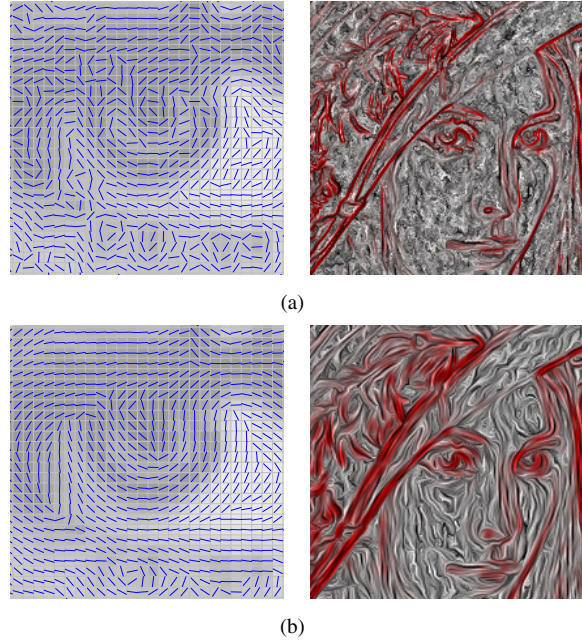
beek discuss in [vVV95] the connection between the structure tensor and double angle representation.

The bilateral filter is a nonlinear operation that smooths images while preserving edges [AW95, TM98, PKTD07]. It is a powerful tool, but computationally very expensive. A brute force implementation can calculate an output pixel in $O(r^2)$ time, where r denotes the filter radius. A faster calculation scheme is presented in [Wei06], but is limited to box spatial kernels and targeted toward vectorization on multiple CPUs. Another fast acceleration technique is presented in [PD06]. They express the bilateral filter as linear convolutions followed by two nonlinearities in a higher-dimensional space where the signal intensity is added to the original domain dimensions. Applications of this technique are presented in [CPD07]. Barash and Comaniciu show in [BC04] that there is a close connection between anisotropic nonlinear diffusion filters and iterated bilateral filters. This motivated Winnemöller et al. [WOG06] to apply iterated bilateral filtering for abstraction. For performance reasons, they were forced to use the xy-separated bilateral filter [PvV05].

DeCarlo and Santella [DS02, SD04] use eye-tracking data to guide image abstraction based on mean-shift segmentation [CM02] at different scales. Edge extraction is based on the Canny edge detector [Can86]. Other authors, e.g., Wang et al. [WXSC04] extended these ideas to video. Hays et al. [HE04] present techniques for transforming images and videos into painterly renderings using brush strokes. They use radial basis functions to interpolate brush orientation. Fischer et al. [FBS05] applied stylization techniques to reduce the visual realism of both the camera image and the virtual graphical objects of an augmented reality system. For edge detection, the Canny edge detector is used. Image abstraction is archived by bilateral filtering. To attain real-time frame rates the bilateral filter is applied to a down sampled version of the input image. A big drawback of Canny edges is that they can become disconnected. Winnemöller et al. [WOG06] therefore use difference-of-Gaussians (DoG) edges for their video abstraction framework motivated by the results of [GRG04], where DoG edges are successfully used for human facial illustrations. For the abstraction part they use iterated bilateral filtering. DoG edges are also used in [RD07], but abstraction is based on the Kuwahara [KHEK76] filter. An image abstraction tool based on gradient domain image processing techniques was recently presented by Orzan et al. in [OBBT07].

3 Method

A schematic overview of our framework is shown in Figure 2. Input is typically an image, a frame of a video or the output of a 3D rendering. We start by estimating the local orientation by calculating the smoothed structure tensor. Then, the input is iteratively abstracted by using the separated orientation-aligned bilateral filter. We perform a total of n_a-iterations. To the result we apply color quantization as described in [WOG06]. After $n_e < n_a$ iterations, we extract edges from

(a)

(b)

Figure 3: *Tangent field induced by the structure tensor before (a) and after (b) smoothing. The right side shows the vector field visualized with line integral convolution. Gradients with high magnitude are colored red.*

the intermediate result using the separated FDoG filter. In our examples we use $n_e = 1$ and $n_a = 4$. Finally, the extracted edges are superimposed on the output of the color quantization.

3.1 Local Orientation Estimation

Our estimation of local orientation is based on the eigenvalues of the structure tensor. To make full use of RGB color information during computation, we combine it with Di Zenzo's multi-image gradient method [DZ86, Cum89]. Suppose $f : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ denotes our input image. We first calculate an approximation to the directional derivatives in x- and y-direction
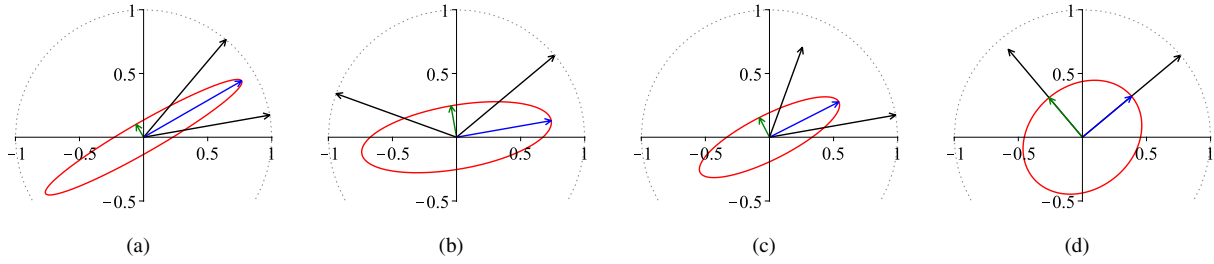
$$\frac{\partial f}{\partial x} = \left(\frac{\partial R}{\partial x} \quad \frac{\partial G}{\partial x} \quad \frac{\partial B}{\partial x} \right)^t, \quad \frac{\partial f}{\partial y} = \left(\frac{\partial R}{\partial y} \quad \frac{\partial G}{\partial y} \quad \frac{\partial B}{\partial y} \right)^t$$

using the Sobel filter. Let $J = (\partial f/\partial x, \partial f/\partial y)$ be the Jacobian matrix then the structure tensor of f is given by:

$$(g_{ij}) = J^T J = \begin{pmatrix} \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial x} & \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial x} \cdot \frac{\partial f}{\partial y} & \frac{\partial f}{\partial y} \cdot \frac{\partial f}{\partial y} \end{pmatrix} =: \begin{pmatrix} E & F \\ F & G \end{pmatrix}$$

The structure tensor induces for a given point a quadratic form which measures the squared rate of change in direction of a vector $n = (n_x, n_y)$:

$$S(n) = En_x^2 + 2Fn_xn_y + Gn_y^2$$

Figure 4: *Smoothing the structure tensor illustrated for two gradients of a grayscale image. (a) More weight is given to closer aligned vectors. (b) Vectors pointing in opposite directions do not cancel out. (c) More weight is given to vectors with higher magnitude. (d) Problematic case: For orthogonal vectors of same magnitude the eigenvalues are equal.*

The extremal values of $S(n)$ on the unit circle correspond to the eigenvalues of (g_{ij}). Since (g_{ij}) is a symmetric matrix, the eigenvalues are real numbers and the eigenvectors are orthogonal. All principal minor determinants of (g_{ij}) are nonnegative and therefore (g_{ij}) is positive semidefinite [Swa73], that is both eigenvalues are non-negative. The major eigenvalue λ_1 is zero if and only if both gradients are zero. The minor eigenvalue λ_2 is zero, e.g., in the case of grayscale images. By solving $\det((g_{ij}) - \lambda_i I) = 0$ the eigenvalues can be found to be:

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

Corresponding eigenvectors are:

$$v_1 = \begin{pmatrix} F \\ \lambda_1 - E \end{pmatrix} \quad v_2 = \begin{pmatrix} \lambda_2 - G \\ F \end{pmatrix}$$

Locally, the maximum rate of change is attained along $\mathbb{R}v_1$ and the minimum rate of change along $\mathbb{R}v_2$. The rate of change is given by $\sqrt{\lambda_1}$ resp. $\sqrt{\lambda_2}$. Note that v_1 and v_2 only represent orientation and do not point into a particular direction, because J and $-J$ map to the same structure tensor (g_{ij}).

By assigning the eigenvector v_2 of the minor eigenvalue to every point of an image, we get a vector field that is aligned to local edge orientation (Figure 3(a)). Unfortunately, this vector field has discontinuities and accordingly, tracing its flow curves, does not give good results. This can be seen on the right side of Figure 3(b) which shows the vector field visualized with line integral convolution [CL93]. In order to smooth the vector field, we apply smoothing to the structure tensor. The result of applying a 5x5 box filter is shown in Figure 3(b). Even better results are achieved using a 7x7 or 9x9 Gaussian filter. The smoothing of the structure tensor is a linear operation, whereas the effect on the eigenvector field is highly nonlinear, which corresponds geometrically to principal component analysis. What is important, is that we, in contrast to [BvVV99, Knu89], do not normalize the structure tensor. This has the effect that more weight is given to vectors of higher magnitude. Figure 4 illustrates the effect of smoothing the structure tensor for two gradients of a

Figure 5: *Multiple iterations of the separated orientation-aligned bilateral filter. Left: Original image. Right: Result after four iterations ($\sigma_d = 3.0, \sigma_r = 4.25$).*

grayscale image. For a given pixel we will refer in the following to v_1 as the *gradient* direction and to v_2 as the *tangent* direction. The vector field induced by the tangents is called the *tangent field*.

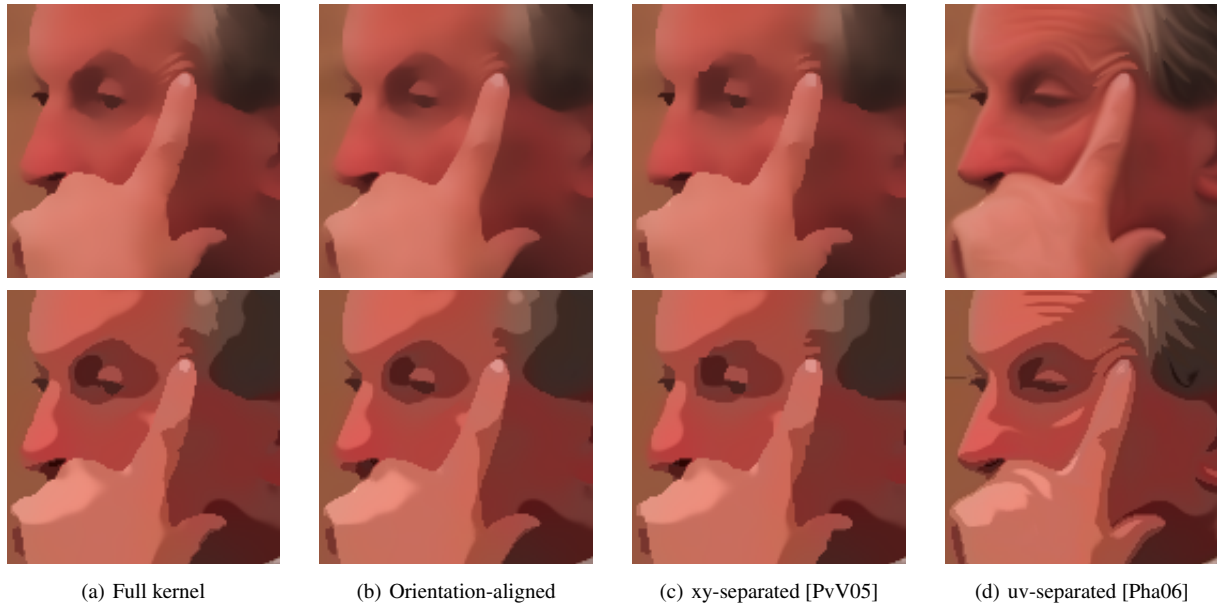### 3.2 Separated Orientation-aligned Bilateral Filter

Before filtering, we convert the input to CIE-Lab color space [WS82] to avoid color bleeding artifacts [TM98]. We use for the closeness and similarity function Gaussian functions of the Euclidean distance between their arguments. For an image f, the bilateral filter is then defined by

$$\frac{\int f(x) G_{\sigma_d}(\|x - \hat{x}\|) G_{\sigma_r}(\|f(x) - f(\hat{x})\|) \, \mathrm{d}x}{\int G_{\sigma_d}(\|x - \hat{x}\|) G_{\sigma_r}(\|f(x) - f(\hat{x})\|) \, \mathrm{d}x},$$
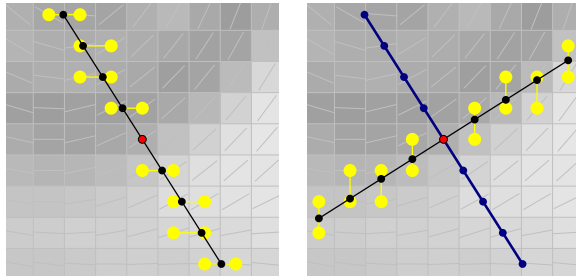
with $\hat{x}$ being the center of the filter neigborhood and G_σ being the one-dimensional Gaussian function of variance σ:

$$G_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{t^2}{2\sigma^2}\right).$$

We are interested in applying the bilateral filter recursively (Figure 5) to achieve a nonlinear diffusion effect [BC04]. Unfortunately, the bilateral filter is not separable. Therefore, it is computationally expensive, especially when applied recursively. We propose to approximate the bilateral filter by a separated implementation that first filters in gradient and then in tangent direction (Figure 7). A comparison of our

| (a) Full kernel | (b) Orientation-aligned | (c) xy-separated [PvV05] | (d) uv-separated [Pha06] |

Figure 6: *Comparison of different bilateral filter implementations ($\sigma_d = 3.0$, $\sigma_r = 4.25\%$). Top row shows the ouput of the filter after 4 iterations. Bottom row shows the filter result with color quantization [WOG06] applied.*

Figure 7: *Separated orientation-aligned bilateral filter. Yellow points indicate sampling positions which are linearly interpolated. Left: First pass filters in gradient direction. Right: Second pass filters in tangent direction.*

approach with other methods to separate the bilateral filter is shown in Figure 6. Sampling in the required direction is achieved by using a constant step direction which has unit size along either x or y axis:

$$\delta(d) = \begin{cases} \left(1, \dfrac{d_y}{d_x}\right) & |d_x| \geq |d_y| \\[2ex] \left(\dfrac{d_x}{d_y}, 1\right) & |d_x| < |d_y| \end{cases}$$

Using unit step size for one axis has the advantage that interpolation at a sampling position can be achieved by linear interpolation between two neighboring pixels (Figure 7). Let c_0 be the sampling origin and either $t_0 = \delta(v_1(c_0))$ to sample in gradient direction or $t_0 = \delta(v_2(c_0))$ to sample in tangent

direction. Then the sampling positions are $c_i^{\pm} = c_0 \pm i\,t_0$ and the response of the bilateral filter in direction t_0 is given by:

$$\frac{1}{k}\Bigg[f(c_0) + \sum_{i=1}^{N} G_{\sigma_d}\big(i\,\|t_0\|\big)\Big[G_{\sigma_r}\big(\|f(c_i^+) - f_0\|\big)f(c_i^+)$$
$$+ \; G_{\sigma_r}\big(\|f(c_i^-) - f_0\|\big)f(c_i^-)\Big]\Bigg]$$

Here k is the normalization factor, e.g., the sum of all filter weights, and N can for example be chosen to be $\lfloor 2\sigma_d / \|t_0\| \rfloor$.

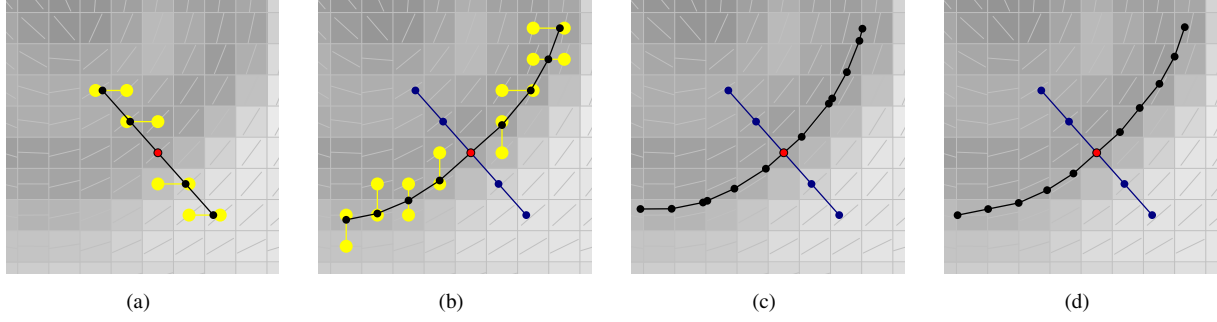### 3.3 Separated Flow-based Difference-of-Gaussians

We extract edges after n_e bilateral filter iterations from the luminance channel. By construction, the gradient points in direction where the rate of change is maximum. Thus, if a point lies on an edge, it is most likely found in gradient direction and applying a one-dimensional DoG filter in gradient direction

$$\int_{\gamma} f(x) D_{\sigma_e}(\|x - \hat{x}\|)\,\mathrm{d}x$$

gives the most filter response (Figure 8(a)). Here, f denotes the luminance channel of the input image, and D_{σ_e} is the DoG operator, which is defined by:

$$D_{\sigma_e}(t) = G_{\sigma_e}(t) - \tau \cdot G_{\sigma_{e'}}(t)$$

The parameter $\sigma_{e'}$ is set to $1.6 \cdot \sigma_e$ to approximate the Laplacian-of-Gaussian [MH80], τ controls the sensitivity of the edge detection. For smaller values, τ detects less noise, but important edges may be missed. We use $\tau = 0.99$ and

Figure 8: *Separated flow-based DoG. Yellow points indicate sampling positions which are linearly interpolated. (a) First pass filters in direction of the gradient. (b) Second pass filters along the flow curves induced by the tangent field. (c) Flow curve integration by line integral convolution [CL93]. (d) Flow curve integration by Euler's method.*

$\sigma_e = 1.0$ in our examples. To evaluate the one-dimensional DoG filter in gradient direction we use the same approach as described in the previous section.

To average the calculated filter response, we apply a one-dimensional Gaussian filter along the flow curves induced by the tangent field (Figure 8(b)). To calculate the sampling positions we adapt the approach for sampling along a line. At every sampling position c_{i-1} a new step direction t_i and a new step length l_i is calculated. The sampling positions are then given by:

$$c_i = c_{i-1} + l_i t_i$$

As step direction, the orientation of the tangent $v_2(c_{i-1})$ is taken. Since the tangent does not define a particular direction, the direction with the smaller change in orientation is chosen:

$$t_i = \text{sgn}\langle v_2(c_{i-1}), t_{i-1}\rangle \cdot v_2(c_{i-1})$$

The step length is chosen such that the new sampling position is aligned to the center of the pixel in either x or y axis:

$$l_i = \begin{cases} \left| \dfrac{c_{i-1,x} - \lfloor c_{i-1,x} \rfloor - \frac{1}{2} - \text{sgn}(t_{i,x})}{t_{i,x}} \right| & |t_{i,x}| \geq |t_{i,y}| \\ \left| \dfrac{c_{i-1,y} - \lfloor c_{i-1,y} \rfloor - \frac{1}{2} - \text{sgn}(t_{i,y})}{t_{i,y}} \right| & |t_{i,x}| < |t_{i,y}| \end{cases}$$

In general, the sampling points are not uniformly distributed along the flow curve. For this reason, the trapezium rule is used for integration. Let c_0 be the starting point, $t_0^+ = v_2(c_0)$ and $t_0^- = -t_0^+$ be the initial positive and negative step directions, and $L_i = \sum_{j=1}^{i} l_j$, then the result of smoothing along the flow curve is given by:

$$\frac{1}{k}\left[\frac{\|t_0^+\|}{2} f(c_0) + \sum_{i=1}^{N^+} \frac{\|t_i^+\| + \|t_{i-1}^+\|}{2} G_{\sigma_m}(L_i^+) f(c_i^+) \right.$$
$$\left. + \frac{\|t_0^-\|}{2} f(c_0) + \sum_{i=1}^{N^-} \frac{\|t_i^-\| + \|t_{i-1}^-\|}{2} G_{\sigma_m}(L_i^-) f(c_i^-) \right]$$

We choose N^+ and N^- such that $L_{N^-}, L_{N^+} < 2\sigma_m$. In our examples we use a variance of $\sigma_m = 3.0$. By construc-

tion, only linear interpolation between two neighboring pixels is necessary. The tangent at a sampling position can be calculated from the interpolated structure tensors of the two neighboring pixels or by using nearest-neighbor interpolation. In the latter case the tangents can be precomputed. For the purpose of comparison Figure 8(c) illustrates flow integration based on line integral convolution [CL93] and Figure 8(d) shows Euler's method.

As final step thresholding is applied. We use the smoothed step function approach described in [WOG06]:

$$T_{\varphi_e}(h) = \begin{cases} 1 & h > 0 \\ 1 + \tanh(\varphi_e \cdot h) & h \leq 0 \end{cases}$$

The parameter φ_e controls the sharpness of the edge output. To smooth aliasing artifacts caused by thresholding, we optionally process the thresholded output using a 3x3 Gaussian filter. Kang et al. [KLC07] use multiple iterations of their flow-based DoG filter to further enhance the filter response. With the exception of Figure 10, all examples shown were created using a single iteration with $\varphi_e = 2$. Figure 9 compares our approach to the isotropic DoG filter and Figure 10 to the FDoG filter. Note the smooth and coherent line and curve segments created by the FDoG variants.

4 Results

We have implemented our framework in C++ using the OpenGL Shading Language (GLSL) and Apple Quicktime for video decoding. Figure 11 shows an assortment of examples created with our method. The following table shows the average frame rate for videos at different resolutions:

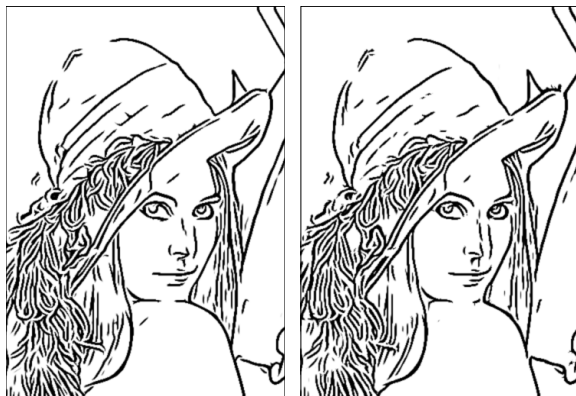| Resolution	8800 GTX	8800 GTS	FX 570M
640 x 480	87.8	53.1	18.1
768 x 576	61.5	37.5	12.4
1280 x 720	29.9	17.5	6.0
1920 x 1080	12.7	5.4	2.15

As test systems two desktop computers (Core2 Duo E6600, 2.4GHz) with Nvidia GeForce 8800 GTX/GTS graphics card and a notebook (Core2 Duo 7700, 2.4GHz) with Nvidia Quadro FX 570M graphics card were used.

(a) Separated FDoG	(b) DoG

Figure 9: *Separated FDoG vs. isotropic DoG.*

(a) Separated FDoG	(b) FDoG [KLC07]

Figure 10: *Multiple FDoG iterations. The images show the result of applying* 3 *iterations of the filter as described in [KLC07].*
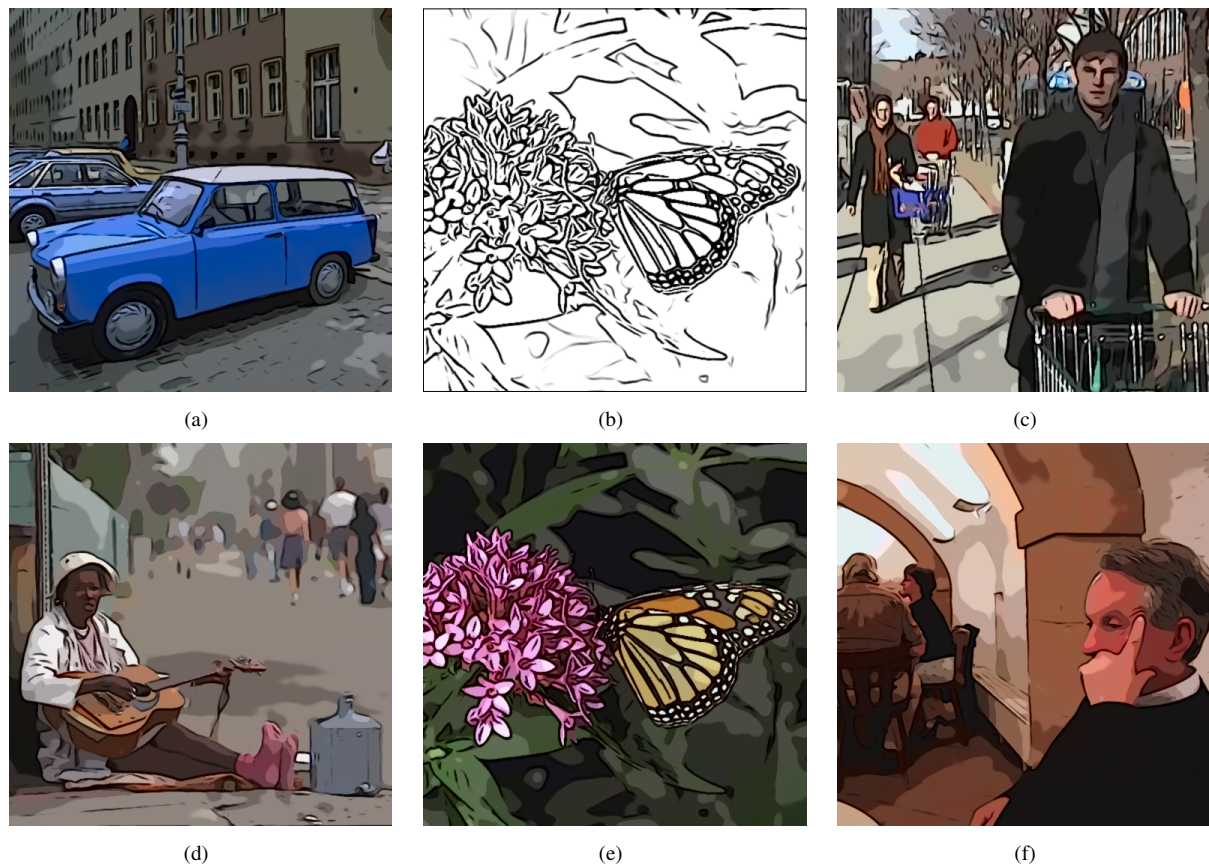
5   Conclusions

In this work we presented a framework of image processing techniques that abstracts visual contents in HD resolution in real-time. We showed that the quality of the xy-separated bilateral filter can be increased by aligning the filter to the local structure. Moreover, we have shown that flow-based DoG filtering can be implemented as a separated algorithm. It outperforms the classical DoG filter in quality, although it is nevertheless efficient to compute. Both presented techniques rely on a smooth vector field whose flow curves follow salient image features. We showed that such a vector field can be constructed from the eigenvalues of the smoothed structure tensor in a highly efficient way.

6   Acknowledgments

References

[AW95] AURICH V., WEULE J.: Non-linear gaussian filters performing edge preserving diffusion. In *Proc. DAGM-Symposium* (1995), pp. 538–545.

[BC04] BARASH D., COMANICIU D.: A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. *Image and Vision Computing 22*, 1 (2004), 73–81.

[BGW91] BIGÜN J., GRANLUND G., WIKLUND J.: Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 8 (1991), 775–790.

[BvVV99] BAKKER P., VAN VLIET L., VERBEEK P.: Edge preserving orientation adaptive filtering. In *Proc. CVPR, IEEE Conference on Computer Vision and Pattern Recognition* (1999), vol. 1, pp. 1535–1540.

[Can86] CANNY J. F.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8* (1986), 769–798.

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

[CM02] COMANICIU D., MEER P.: Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (2002), 603–619.

[CPD07] CHEN J., PARIS S., DURAND F.: Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics 26*, 3 (2007), 103.

[Cum89] CUMANI A.: *Edge Detection in Multispectral Images*. Tech. rep., Istituto Elettrotecnico Nazionale "Galileo Ferraris", 1989.

[DS02] DECARLO D., SANTELLA A.: Stylization and abstraction of photographs. *ACM Transactions on Graphics 21*, 3 (2002), 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.

[FBS05] FISCHER J., BARTZ D., STRABER W.: Stylized augmented reality for improved immersion. In *Proc. VR, IEEE Virtual Reality* (2005), pp. 195–202.

[GRG04] GOOCH B., REINHARD E., GOOCH A.: Human facial illustrations: Creation and psychophysical evaluation. *ACM Transactions on Graphics 23*, 1 (2004), 27–44.

[HE04] HAYS J., ESSA I. A.: Image and video based painterly animation. In *Proc. ACM NPAR* (2004), pp. 113–120.

[HS88] HARRIS C., STEPHENS M.: A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference* (1988), pp. 147–151.

[KHEK76] KUWAHARA M., HACHIMURA K., EIHO S., KINOSHITA M.: *Digital processing of biomedical images*. Plenum Press, 1976, pp. 187–203.

[KLC07] KANG H., LEE S., CHUI C. K.: Coherent line drawing. In *Proc. ACM NPAR* (2007), pp. 43–50.

[Knu89] KNUTSSON H.: Representing local structure using tensors. In *The 6th Scandinavian Conference on Image Analysis* (Oulu, Finland, June 1989), pp. 244–251.

[KW87] KASS M., WITKIN A.: Analyzing oriented patterns. *Computer Vision, Graphics, and Image Processing 37*, 3 (March 1987), 362–385.

[MH80] MARR D., HILDRETH R. C.: Theory of edge detection. *Proc. Royal Society London 207* (1980), 187–217.

(a)   (b)   (c)

(d)   (e)   (f)

Figure 11: *Examples. Separated orientation-aligned bilateral filter: $n_a = 4$, $\sigma_d = 3.0$, $\sigma_r = 4.25\%$. Separated flow-based DoG: $n_e = 1$, $\tau = 0.99$, $\sigma_e = 1.0$, $\sigma_m = 3.0$, $\varphi = 2.0$. Color quantization: $q = 8$, $\varphi_q = 3.4$.*

[OBBT07] ORZAN A., BOUSSEAU A., BARLA P., THOLLOT J.: Structure-preserving manipulation of photographs. In *Proc. ACM NPAR* (2007).

[PD06] PARIS S., DURAND F.: A fast approximation of the bilateral filter using a signal processing approach. In *European Conference on Computer Vision* (2006), vol. 4, pp. 568–580.

[Pha06] PHAM T.: *Spatiotonal adaptivity in Super-Resolution of Undersampled Image Sequences*. PhD thesis, Quantitative Imaging Group, Delft University of Technology, 2006.

[PKTD07] PARIS S., KORNPROBST P., TUMBLIN J., DURAND F.: A gentle introduction to bilateral filtering and its applications. In *ACM SIGGRAPH courses* (2007).

[PvV05] PHAM T., VAN VLIET L.: Separable bilateral filtering for fast video preprocessing. In *IEEE International Conference on Multimedia and Expo* (2005).

[RD07] REDMOND N., DINGLIANA J.: Adaptive abstraction of 3d scenes in real-time. In *EUROGRAPHICS Short Presentations* (2007).

[SD02] SANTELLA A., DECARLO D.: Abstracted painterly renderings using eye-tracking data. In *Proc. ACM NPAR* (2002), pp. 75–82.

[SD04] SANTELLA A., DECARLO D.: Visual interest and npr: An evaluation and manifesto. In *Proc. ACM NPAR* (2004), pp. 71–150.

[Swa73] SWAMY K.: On sylvester's criterion for positive-semidefinite matrices. *IEEE Transactions on Automatic Control 18*, 3 (1973), 306–306.

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

[vVV95] VAN VLIET L., VERBEEK P.: Estimators for orientation and anisotropy in digitized images. In *Proc. ASCI, Conference of the Advanced School for Computing and Imaging, Heijen* (1995), vol. 95, pp. 16–18.

[Wei06] WEISS B.: Fast median and bilateral filtering. *ACM Transactions on Graphics 25*, 3 (2006), 519–526.

[WOG06] WINNEMÖLLER H., OLSEN S. C., GOOCH B.: Real-time video abstraction. *ACM Transactions on Graphics 25*, 3 (2006), 1221–1226.

[WS82] WYSZECKI G., STILES W. S.: *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley Interscience Publishers, New York, 1982.

[WXSC04] WANG J., XU Y., SHUM H.-Y., COHEN M. F.: Video tooning. *ACM Transactions on Graphics 23*, 3 (2004), 574–583.