Image and Video Abstraction by Multi-scale Anisotropic Kuwahara Filtering

Jan Eric Kyprianidis

Hasso-Plattner-Institut, Germany *

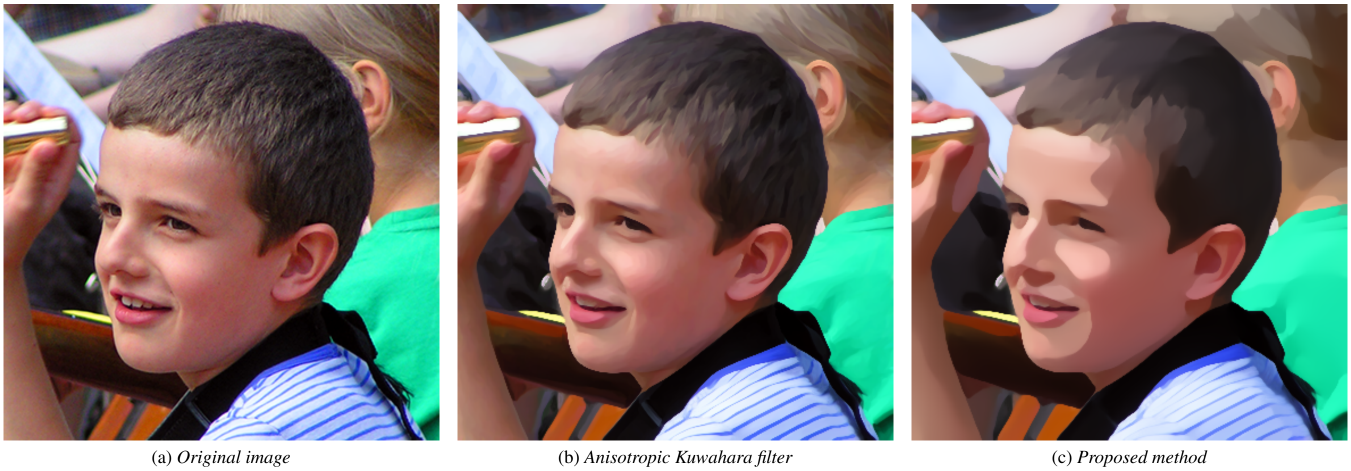

(a) *Original image* (b) *Anisotropic Kuwahara filter* (c) *Proposed method*

Figure 1: *Example comparing the proposed multi-scale approach with the single-scale approach.*

Abstract

The anisotropic Kuwahara filter is an edge-preserving filter that is especially useful for creating stylized abstractions from images or videos. It is based on a generalization of the Kuwahara filter that is adapted to the local structure of image features. In this work, two limitations of the anisotropic Kuwahara filter are addressed. First, it is shown that by adding thresholding to the weighting term computation of the sectors, artifacts are avoided and smooth results in noise-corrupted regions are achieved. Second, a multi-scale computation scheme is proposed that simultaneously propagates local orientation estimates and filtering results up a low-pass filtered pyramid. This allows for a strong abstraction effect and avoids artifacts in large low-contrast regions. The propagation is controlled by the local variances and anisotropies that are derived during the computation without extra overhead, resulting in a highly efficient scheme that is particularly suitable for real-time processing on a GPU.

CR Categories: I.3.3 [Computing Graphics]: Picture/Image Generation—Display algorithms; I.4.3 [Computing Methodologies]: Image Processing And Computer Vision—Enhancement Filtering

Keywords: Non-photorealistic rendering, image abstraction, anisotropic Kuwahara filter

1 Introduction

A common approach to creating non-photorealistic depictions is to transform an image or video using an interactive or automatic technique. A classical example is the painting system by Haeberli [1990], where properties such as color, size and orientation of interactively placed brush strokes are guided by an input image. An example of an automatic system for transforming videos using various painting styles is [Hays and Essa 2004], wherein the brush strokes are placed automatically. Temporal coherent results are achieved by using optical flow analysis.

Instead of focusing on simulating a particular artistic technique or style, image abstraction refers to the process of simplifying scene information by removing unnecessary information that is irrelevant for a particular purpose. A common approach to image abstraction is segmentation. Several methods based on mean shift have been proposed for abstracting images [DeCarlo and Santella 2002; Wen et al. 2006] and videos [Wang et al. 2004; Collomosse et al. 2005]. Typically, the segmented regions created by mean shift have rough boundaries and therefore require elaborate post-processing. The methods that deal with video are also complicated, since they perform processing in the spatiotemporal domain.

Another way to perform stylization and abstraction of images and videos is through the use of edge-preserving smoothing and enhancement filters. Prominent techniques in this area have in common that they remove detail in low-contrast regions without filtering across discontinuities, thus leaving the overall structure of the input image unaffected. Popular examples are the bilateral filter, the Kuwahara filter, and techniques based on or motivated by partial differential equations (PDE).

Among these, the anisotropic Kuwahara filter [Kyprianidis et al. 2009] is of particular interest in image and video abstraction. It creates a feature-preserving and direction-enhancing look and, unlike other nonlinear smoothing filters, is very robust against high-contrast noise. In addition, it avoids overblurring in low-contrast regions and provides a consistent level of abstraction across the image. Moreover, excellent temporal coherence is achieved when applied to video on a frame-by-frame basis. However, the level of abstraction that is achievable with the anisotropic Kuwahara filter is limited by the filter radius. Simply increasing the filter radius is typically not a solution, as it often results in artifacts. A possibility would be to control the radius adaptively per pixel depending on the local neighborhood, but the computational cost would be very high, as the filter

*http://www.hpi3d.de

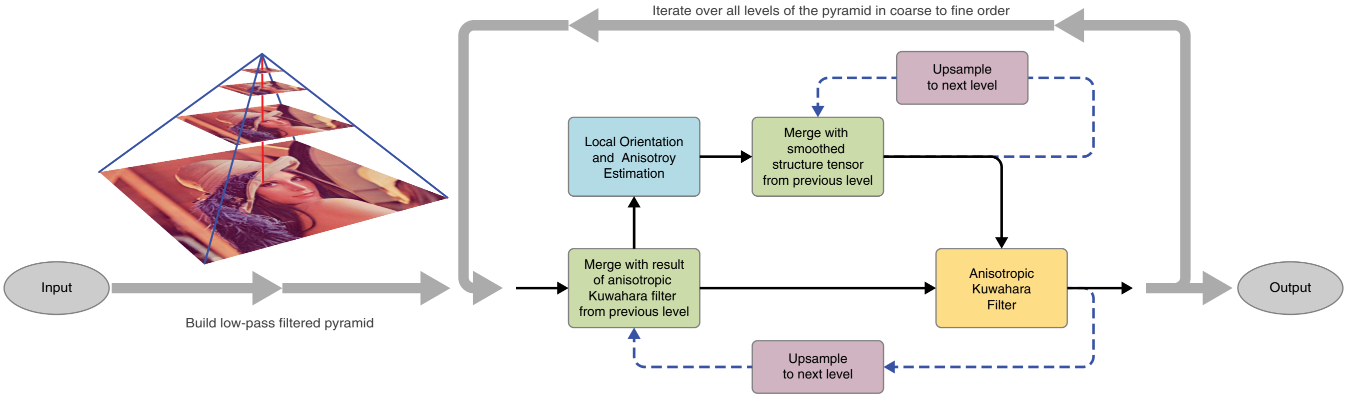

Iterate over all levels of the pyramid in coarse to fine order
Upsample to next level
Local Orientation and Anisotroy Estimation
Merge with smoothed structure tensor from previous level
Input
Build low-pass filtered pyramid
Merge with result of anisotropic Kuwahara filter from previous level
Anisotropic Kuwahara Filter
Output
Upsample to next level

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

depends quadratically on the radius. In this work, a better solution is provided by generalizing the anisotropic Kuwahara filter to operate on multiple scales. The computations are carried out on an image pyramid, where processing is performed in a coarse-to-fine manner, with intermediate results being propagated up the pyramid.

2 Related Work

The use of image pyramids is a popular tool in computer graphics and image processing. It goes back to the early work of Burt and Adelson [1983] and Williams [1983]. While image pyramids are often used to speed up computationally expensive operations on large images, scale-space theory [Lindeberg 1996] provides a sophisticated theory for representing and analyzing images at different scales. Several techniques in the field of non-photorealistic rendering make use of image pyramids or scale-space techniques. An example is the stroke-based painting technique by Hertzmann [1998]. Here, the final image is iteratively painted in a coarse-to-fine manner. Starting with a large brush size, the brush size is lowered for every iteration. Location, orientation, and color are determined by analyzing the source image at a scale related to the current brush size. An example for a technique that uses an image pyramid is the image abstraction technique of DeCarlo and Santella [2002]. Mean-shift segmentation is performed on each pyramid level and the results are then organized in a tree structure representing the relationships of the different color regions and boundaries. Guided by eye-tracking data, this structure is then used to highlight or abstract different parts of the image.

A well-known edge-preserving smoothing filter is the bilateral filter [Tomasi and Manduchi 1998]. Winnemöller et al. [2006] combined bilateral filtering with color quantization and difference of Gaussians edges, to create cartoon-style abstractions from images and videos. Kyprianidis and Döllner [2008] extended this approach and presented separable implementations of the bilateral and difference of Gaussians filters that are aligned to the local image structure. Kang et al. [2009] presented a similar system, wherein 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, typically resulting in an inconsistent abstraction.

Another popular edge-preserving smoothing filter is the Kuwahara filter [Kuwahara et al. 1976]. The general idea behind this filter is to divide the filter kernel into four rectangular subregions that overlap by one pixel. The filter response is then defined by the mean of a subregion with minimum variance. The Kuwahara filter produces clearly noticeable artifacts, which are due to the use of rectangular subregions. In addition, the subregion selection process is unstable if noise is present or if subregions have the same variance, which then results in randomly chosen subregions and corresponding artifacts. A more detailed discussion of limitations of the Kuwahara filter can be found in [Papari et al. 2007].

Several attempts have been made to address the limitations of the Kuwahara filter. Papari et al. [2007] defined a new criterion for overcoming the limitations of the unstable subregion selection process. Instead of selecting a single subregion, the result is defined as the weighted sum of the means of the subregions. The weights are defined based on the variances of the subregions, resulting in smoother region boundaries and fewer artifacts. To improve this further, the rectangular subregions are replaced by smooth weighting functions defined over sectors of a disc.

The anisotropic Kuwahara filter [Kyprianidis et al. 2009] builds upon the generalized Kuwahara filtering concept by Papari et al. [2007] and replaces the weighting functions defined over sectors of a disc by weighting functions defined over ellipses. By adapting shape, scale and orientation of these ellipses to the local structure of the input, artifacts are avoided. With this adaption, directional image features are better preserved and emphasized, resulting in overall sharper edges and the enhancement of directional image features. A further modification has been presented in [Kyprianidis et al. 2010b], wherein new weighting functions based on polynomials are defined that can be evaluated directly during the filtering process.

A further image abstraction technique has been presented by Kang and Lee [2008]. 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 their geometry. Since mean curvature flow does not properly protect directional image features, Kang and Lee constrained the mean curvature flow. Mean curvature and its constrained variant contract isophote curves to points [Grayson 1987]. 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 is not suitable for per-frame video processing. Another technique based on diffusion and shock filtering has been recently presented by Kyprianidis and Kang [2011], wherein flow-guided smoothing and sharpening orthogonal to the flow are combined. Instead of modeling the process by a PDE, approximations that operate as local filters on a neighborhood of a pixel are used. This makes the technique more stable and, in particular, suitable for per-frame video processing. Interestingly, there is a connection between PDE-based techniques and the Kuwahara filter. As shown by van den Boomgaard [2002], the Kuwahara filter can be interpreted as a PDE with linear diffusion and shock filter term.

An image abstraction technique based on image processing in the gradient domain has been presented by Orzan et al. [2007]. The technique is known not to create temporal coherent output for video. Bhat et al. [2010] have presented 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

A schematic overview of the proposed technique is shown in Figure 2. Processing starts with building an image pyramid of the input image. Next, the pyramid is processed from the coarsest level to the finest level. Processing starts with calculation of the smoothed structure tensor and anisotropic Kuwahara filter for the coarsest level. These results are then upsampled to the next finer level. Based on an approximation of the local variances, the upsampled filtering result of the anisotropic Kuwahara filter is then merged with the image data of the current pyramid level and the smoothed structure tensor is calculated from this merged result. Based on their anisotropy measure, the smoothed structure tensor and the upsampled structure tensor from the previous level are now merged. Using the merged structure tensor, the anisotropic Kuwahara filter is applied to the merged image data. The result of the anisotropic Kuwahara filter and the merged structure tensor are upsampled, and the process is repeated for the next finer level until the finest level of the pyramid is reached. The different parts of the algorithm are discussed in detail in the following sections.

3.1 Pyramid Construction

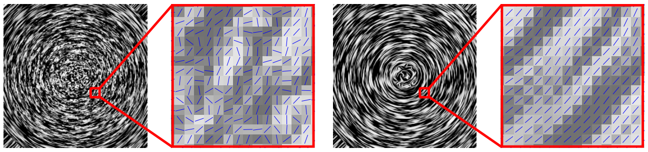
The pyramid construction is performed using resampling convolution [Schumacher 1992]. A down-sampling factor of 2 was used for all experiments. Motivated by scale-space theory [Lindeberg 1996] and the classical approach by Burt and Adelson [1983] for building a pyramid, a windowed Gaussian was used initially as the resampling filter. However, the Gaussian filter tends to remove too many low frequencies, resulting in an image that is less sharp. Therefore, several popular resampling filters were considered, including bilinear, cubic, Catmull-Rom, Mitchell-Netravali, and Lanczos. Among these, the Lanczos3 filter [Blinn 1989] was found to provide the best result for most examples. As pointed out by Blinn, the Lanczos3 filter keeps low frequencies and rejects high frequencies better than most other well-known filters, making it well suited for the technique presented. Using the Lanczos3 filter might create or enhance local extrema and violates the causality axiom of scale-space theory [Lindeberg 1996]; nevertheless, it works very well in this particular case.

3.2 Local Structure Estimation

The estimation of local orientation is based on the eigenanalysis of the structure tensor. This section provides background material on the calculation of image gradients, the structure tensor, and the anisotropy measure used. After that, the proposed multi-scale local structure estimation scheme is presented.

3.2.1 Gradient 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. The usage of Gaussian derivatives corresponds to smoothing the image before gradient calculation with a Gaussian filter. However, this smoothing process also results in a loss of information and leads to a less accurate estimation. A better approach is to use a small 3 × 3 filter for the gradient calculation and then smooth the structure tensor after its calculation. This can be done, for example, by using the Sobel filter. However, the classical Sobel filter is not rotationally symmetric; therefore, a better choice is the filter developed by Jähne et al. [1999], which is optimized for rotational symmetry:


(a) *No smoothing*

(b) *With smoothing*

Figure 3: *Visualization of the minor eigenvector of the structure tensor for a test ring pattern corrupted with Gaussian noise of variance $v = 0.02$.*

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

with $p_1 = 0.183$. Let f denote the input image. Then, approximations of the partial derivatives in x- and y-directions are given by convolution with the corresponding filter stencils:

$$\frac{\partial f}{\partial x} \approx D_x \star f \qquad \frac{\partial f}{\partial y} \approx D_y \star f$$

3.2.2 Structure Tensor

In this section, two approaches for defining the structure tensor are presented. The first approach follows the geometric point of view [Di Zenzo 1986; Cumani 1991] using Riemannian Geometry [Lee 2003], and the second approach is based on the formulation of an optimization problem and closely follows [Brox et al. 2006].

Geometric Definition Let $U \subset \mathbb{R}^2$ and let us assume that the input image is given by a differential function $f : U \to \mathbb{R}^n$. Here, case $n = 1$ corresponds to gray-scale images and case $n = 3$ corresponds to color images. Given this function, we can now consider its graph as a two-dimensional submanifold of $\mathbb{R}^{2+n}$. The map

$$F : \left\{ \begin{array}{ccc} U & \longrightarrow & \mathbb{R}^{2+n} \\ (u,v) & \mapsto & \big(u, v, f^1(u,v), \dots, f^n(u,v)\big) \end{array} \right.$$

then defines a smooth global parameterization of the embedding, and the Euclidean metric $\overline{g}$ of $\mathbb{R}^{2+n}$ induces a Riemannian metric on the graph that is given in chart coordinates by:

$$\begin{aligned} g = F^*\overline{g} &= F^*\Big(\big(\mathrm{d}x^1\big)^2 + \ldots + \big(\mathrm{d}x^{2+n}\big)^2\Big) \\ &= \mathrm{d}\big(x^1 \circ F\big)^2 + \ldots + \mathrm{d}\big(x^{2+n} \circ F\big)^2 \\ &= \mathrm{d}u^2 + \mathrm{d}v^2 + \textstyle\sum_{i=1}^n \big(\mathrm{d}f^i\big)^2 \end{aligned}$$

Here, ω^2 denotes the common abbreviation for the symmetric product of a tensor ω with itself. Now, let us consider a point $p \in U$ of the image domain and a tangent vector $X \in T_pU \simeq \mathbb{R}^2$. The Euclidean length of X in local coordinates corresponds to the length in the image domain and the length given by the induced Riemannian metric

$$\sqrt{\langle X, X\rangle_p} = \sqrt{g_p(X, X)}$$

corresponds to measuring length on the image embedded as a graph in $\mathbb{R}^{2+n}$. Hence, for tangent vectors of fixed Euclidean length, e.g. $\|X\| = 1$, the Riemannian metric g can be interpreted as measuring the squared local rate of change in direction X.

We are interested in finding the local orientation at p; i.e., the directions where the rate of change of the image regarded as a graph is either minimum or maximum. For unit length tangent vectors, the term $\mathrm{d}u^2 + \mathrm{d}v^2$ of g is constant and equal to one. To find the minimum and maximum local rate of change, it is therefore sufficient to consider only the term $\sum_{i=1}^n (\mathrm{d}f^i)^2$. The differential of the i-th component of f is given by

$$\mathrm{d}f^i = \frac{\partial f^i}{\partial u}\mathrm{d}u + \frac{\partial f^i}{\partial v}\mathrm{d}v$$

and its symmetric product with itself is given by

$$\begin{aligned}(\mathrm{d}f^i)^2 &= \mathrm{d}f^i \otimes \mathrm{d}f^i \\ &= \left(\frac{\partial f^i}{\partial u}\right)^2 \mathrm{d}v^2 + 2\frac{\partial f^i}{\partial u}\frac{\partial f^i}{\partial v}\mathrm{d}u\mathrm{d}v + \left(\frac{\partial f^i}{\partial v}\right)^2 \mathrm{d}v^2 \,.\end{aligned}$$

Thus we get

$$\textstyle\sum_{i=1}^n (\mathrm{d}f^i)^2 = E\,\mathrm{d}u + 2F\,\mathrm{d}u\mathrm{d}v + G\,\mathrm{d}v^2 \,, \qquad (1)$$

where

$$J := \begin{pmatrix} E & F \\ F & G \end{pmatrix} := \begin{pmatrix} \sum_{i=1}^n \left(\frac{\partial f^i}{\partial u}\right)^2 & \sum_{i=1}^n \frac{\partial f^i}{\partial u}\frac{\partial f^i}{\partial v} \\ \sum_{i=1}^n \frac{\partial f^i}{\partial u}\frac{\partial f^i}{\partial v} & \sum_{i=1}^n \left(\frac{\partial f^i}{\partial v}\right)^2 \end{pmatrix}$$

is the so-called *structure tensor*. For gray-scale images, the structure tensor is simply given by the outer product of the gradient

$$J = \nabla f \nabla f^\mathsf{T} \,,$$

and for color images it is given by the sum of the outer products of the gradients of the different color channels. For example, for an RGB color image we have:

$$J = \nabla R \nabla R^\mathsf{T} + \nabla G \nabla G^\mathsf{T} + \nabla B \nabla B^\mathsf{T}$$

The extremal values of the quadratic form (1) on the unit circle correspond to the eigenvalues of the structure tensor J. Since J is a symmetric matrix, the eigenvalues are real numbers and the eigenvectors are orthogonal. All principal minor determinants of J are non-negative and therefore J is positive semidefinite; i.e., both eigenvalues are non-negative [Swamy 1973]. The eigenvalues can be directly computed by solving $\det(J - \lambda_i \mathrm{I}) = 0$ and are given by

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

The corresponding eigenvectors are

$$\xi = \begin{pmatrix} F \\ \lambda_1 - E \end{pmatrix} \quad \text{and} \quad \eta = \begin{pmatrix} \lambda_2 - G \\ F \end{pmatrix} \,,$$

where ξ is the direction of maximum change and η is the direction of minimum change. Moreover, the local orientation in the direction of the minor eigenvector can be directly derived from the structure tensor and is given by:

$$\varphi = \frac{1}{2}\arctan\left(\frac{2F}{E-G}\right) + \frac{\pi}{2}$$

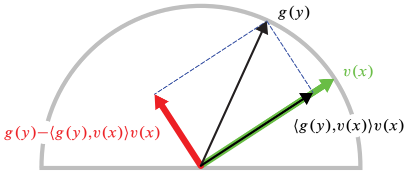

$g(y)$
$v(x)$
$g(y)-\langle g(y),v(x)\rangle v(x)$
$\langle g(y),v(x)\rangle v(x)$

Figure 4: *Deviation of a gradient $g(y) := \nabla(y)$ from a given direction $v(x)$. The term $\langle g(y), v(x)\rangle v(x)$ is the projection of $g(y)$ onto $v(x)$.*

So far we have made the assumption that the image is given by a smooth function. This is obviously wrong for typical natural images, as can be seen in Figure 3(a). Therefore, a common practice is to smooth the image prior to derivative computation with, for example, a Gaussian filter. However, a better approach is to smooth the structure tensor instead of the image as shown in Figure 3(b). This can be best understood from the optimization point of view.

Definition by Optimization Let $I : U \to \mathbb{R}$ be a gray-scale image and let $g(y) := \nabla I(y)$, $y \in U$ be the gradients of the image. Moreover, let $x \in U$ be a fixed but arbitrary point. Now, suppose a direction $v(x) \in \mathbb{R}^2$ is given. Then, the deviation of a single gradient $g(y)$ from $v(x)$ can be defined as (see also Figure 4):

$$e(x,y) = \|g(y) - \langle g(y), v(x)\rangle v(x)\|$$

The total squared error at x can then be defined by the convolution of e^2 with a Gaussian function G_ρ:

$$E(x) = \int G_\rho(x-y) \cdot e^2(x,y)\,\mathrm{d}y \qquad (2)$$

Using the linearity of the scalar product, $e^2(x,y)$ can be simplified to:

$$\begin{aligned} e^2 &= \langle e, e\rangle = \big\langle e, g - \langle g, v\rangle v\big\rangle = \big\langle e, g\big\rangle - \overbrace{\big\langle e, \langle g, v\rangle v\big\rangle}^{=0} \\ &= \langle g, g\rangle - \langle g, v\rangle^2 = g^\mathsf{T} g - v^\mathsf{T}\big(g g^\mathsf{T}\big)v \end{aligned}$$

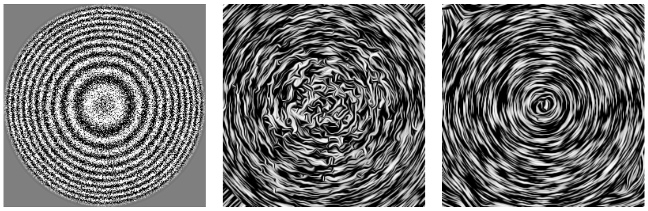
By substituting this into equation (2) we get:

$$\begin{aligned} E(x) = \int G_\rho(x-y)\, g(y)^\mathsf{T} g(y)\,\mathrm{d}y \; - \\ \int G_\rho(x-y)\, v(x)^\mathsf{T}\big(g(y)g(y)^\mathsf{T}\big)v(x)\,\mathrm{d}y \end{aligned}$$

The first term of $E(x)$ is constant. Therefore, minimizing $E(x)$ is equivalent to maximizing the second term. Using the linearity of the integral, this term can be rewritten as

$$\begin{aligned} &\int G_\rho(x-y)\, v(x)^\mathsf{T}\big(g(y)g(y)^\mathsf{T}\big)v(x)\,\mathrm{d}y \\ &\qquad = v(x)^\mathsf{T}\underbrace{\left(\int G_\rho(x-y)\; g(y)g(y)^\mathsf{T}\,\mathrm{d}y\right)}_{:=J_\rho(x)}v(x) \,, \end{aligned}$$

with $J_\rho(x)$ being the *smoothed structure tensor* at x. Using the method of Lagrange multipliers, it can now be shown [Brox et al. 2006] that maximizing $v^\mathsf{T} J_\rho\, v$ with the constrained $\|v\| = 1$ is equivalent to an eigenanalysis of $J_\rho\, v = \lambda v$, and that the vector that minimizes $E(x)$ is given by the major eigenvector. This shows that smoothing the structure tensor corresponds to solving a weighted least squared problem. In noise-corrupted images, this generally leads to more stable estimates.



(a) *Test image*

(b) *Optimized Sobel filter*

(c) *Multi-scale estimation*

Figure 5: *Visualization of the minor eigenvector field of the smoothed structure tensor ($\rho = 2$) for a test ring pattern corrupted by Gaussian noise of variance $\nu = 0.25$.*

3.2.3 Anisotropy Measure

Besides its superior robustness, as explained in the previous section, the smoothed structure tensor has a second advantage over orientation averaging techniques, such as the mean of angles [Mardia and Jupp 1999] or the edge tangent flow [Kang et al. 2007]. The minor eigenvalue λ_2 of the structure tensor measures how much the gradients deviate from the axis defined by the major eigenvector. It provides a way to measure the quality of the orientation estimation. If $\lambda_1 \gg \lambda_2$ there is a clear dominant orientation in the considered neighborhood. If, on the other hand, $\lambda_1 \approx \lambda_2$, there is no particular designated axis among the gradients. The normalized difference of the eigenvalues thus provides a *measure of anisotropy* [Yang et al. 1996]:

$$A = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}$$

The anisotropy A ranges from 0 to 1, where 0 corresponds to isotropic and 1 corresponds to entirely anisotropic regions.

3.2.4 Multi-scale Estimation

The multi-scale local structure estimation is inspired by the multi-scale orientation estimation methods developed by Wilson et al. [1990] and Feng and Milanfar [2002]. Similar to these approaches, a weighted linear combination is used to propagate the estimates from coarser to finer pyramid levels. However, in contrast to the other techniques that propagate gradient estimates, in this work the structure tensor is propagated.

The processing starts at the coarsest level of the pyramid, where the smoothed structure tensor is calculated as usual. For the other pyramid levels, the smoothed structure tensor is calculated and then combined with the upsampled structure tensor from the previous level, using linear interpolation:

$$\tilde{J}_\rho^k = \alpha^k J_\rho^k + \left(1 - \alpha^k\right) \bar{J}_\rho^{k+1}$$

Here, $\bar{J}_\rho^{k+1}$ denotes the upsampled structure tensor from the previous level, i.e., $\tilde{J}_\rho^{k+1}$ upsampled to the next level. J_ρ^k is the smoothed structure tensor computed from the merged image data of the current pyramid level. The linear weighting factor is defined per pixel and based on the anisotropy measure:

$$\alpha^k = \frac{A^k}{A^k + \bar{A}^{k+1}}$$

Here, A^k is the anisotropy from the current level and $\bar{A}^k + 1$ is the upsampled anisotropy from the previous level, which means that more weight is given to the structure tensor that is more anisotropic. This leads to a more robust estimation, as shown in Figure 5.

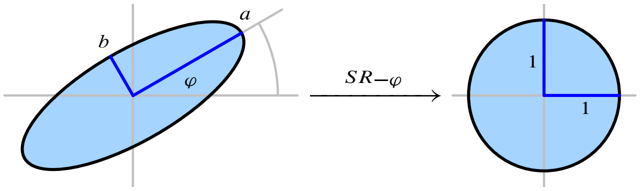


Figure 6: *The mapping $SR_{-\varphi}$ maps a rotated ellipse to the unit disc.*

3.3 Anisotropic Kuwahara Filter

In this section, first the working principles of single-scale anisotropic Kuwahara filtering are reviewed. The presentation mainly follows [Kyprianidis et al. 2009], but in addition, a new thresholding step for the weighting factor computation is proposed. Then, the multi-scale approach is presented.

3.3.1 Single-scale Filtering

Let $f : \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ denote the input image, let $(x_0, y_0) \in \mathbb{R}^2$ be a point, let φ be the local orientation and let A be the anisotropy at (x_0, y_0). To adjust the eccentricity of the filter shape depending on the amount of anisotropy, we set:

$$a = \frac{\alpha + A}{\alpha} r \quad \text{and} \quad b = \frac{\alpha}{\alpha + A} r \, .$$

Here r denotes the desired radius of the filter and $\alpha > 0$ is a tuning parameter that is typically set to $\alpha = 1$. In this case, since A is in $[0, 1]$, it follows that we have $r \leq a \leq 2r$ and $r/2 \leq b \leq r$. Now let

$$S = \begin{pmatrix} a^{-1} & 0 \\ 0 & b^{-1} \end{pmatrix}$$

and let $R_{-\varphi}$ be the matrix defining a rotation by $-\varphi$. The mapping $SR_{-\varphi}$ then defines a linear coordinate transform that maps a rotated ellipse to the unit disc (Figure 6). The anisotropic Kuwahara filter now partitions this ellipse into different sectors similar to the rectangular areas of the original Kuwahara filter. Let N denote the number of sectors, with typical values $N = 4$ or $N = 8$. The different sectors must overlap. To achieve this, the anisotropic Kuwahara uses weighting functions that define how much influence a pixel has on a sector. To define these weighting functions over the ellipse, the general idea is to define corresponding weighting functions over the unit disc and then pull these back to the ellipse.

Let χ_0 be the characteristic function that is 1 for all points of $\mathbb{R}^2$ with argument in $(-\pi/N, \pi/N]$ and 0 otherwise. Then

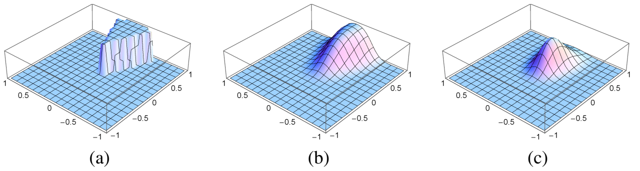
$$K_0 = (\chi_0 \star G_\rho) \cdot G_\sigma$$

defines smooth weighting function over the unit circle. Here, G_ρ and G_σ denote Gaussian functions and $\star$ denotes convolution. The convolution smoothes the characteristic function such that pixels from neighboring sectors are also considered in the weighting process. The multiplication achieves a decay with increasing radius (Figure 7). Reasonable values for σ_r and σ_s are $\sigma_r = 0.4$ and $\sigma_s = \sigma_r/3$. Weighting functions for the other sectors can be defined by smoothing the corresponding characteristic function or simply by rotating K_0:

$$K_i = K_0 \circ R_{-2\pi i/N}, \quad i = 0, \ldots, N-1$$

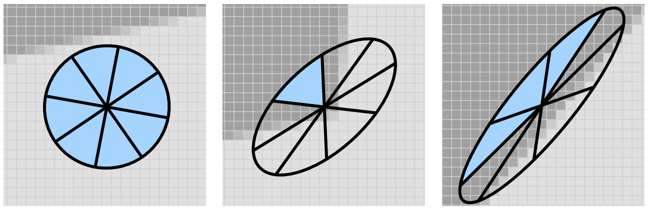
Here, $\circ$ denotes composition of functions. By pulling back K_i, we finally get weighting functions w_i defined over the ellipse:

$$w_i = K_i \circ SR_{-\varphi} = K_0 \circ R_{-2\pi i/N} SR_{-\varphi}$$



(a) (b) (c)

Figure 7: *Construction of the weighting functions of the anisotropic Kuwahara filter: (a) Characteristic function χ_0. (b) $\chi_0 \star G_\rho$. (c) $K_0 = (\chi_0 \star G_\rho) \cdot G_\sigma$.*



(a) *Homogeneous* (b) *Corner* (c) *Edge*

Figure 8: *The anisotropic Kuwahara filter uses weighting functions defined over an ellipse, whose shape is based on the local orientation and anisotropy. The filter response is defined as a weighed sum of the local averages, where more weight is given to averages with low standard deviation.*

Now let

$$m_i = \frac{1}{k} \int f(x)\, w_i(x - x_0)\, \mathrm{d}x$$

be the weighted local averages and

$$s_i^2 = \frac{1}{k} \int f^2(x)\, w_i(x - x_0)\, \mathrm{d}x \;-\; m_i^2$$

be the squared standard deviations, where k denotes the corresponding normalization factor. The output of the anisotropic Kuwahara filter is then defined as a weighted sum of the local averages of the sectors:

$$\frac{\sum \omega_i m_i}{\sum \omega_i} = \frac{\omega_0 m_0 + \cdots + \omega_{N-1} m_{N-1}}{\omega_0 + \cdots + \omega_{N-1}}$$

The weights ω_i in this sum are defined by:

$$\omega_i = \Big(\max\big(\tau_w, \|s_i\|\big)\Big)^{-q}$$

This definition of the weighting factors ω_i ensures that more weight is given to sectors with low standard deviation $\|s_i\|$, i.e., those that are more homogeneous. At color region boundaries, sectors that lie completely on one side of the boundary have low standard deviation and thus receive high weight. Sector that cross the boundary have high standard deviation and receive a low weight (Figure 8). Since by construction all sectors contain the filter origin, it is not possible that two sectors lie completely in different color regions. However, in homogeneous regions the standard deviation is low for all sectors. Small differences in the standard deviation, for example as a result of noise, can therefore lead to random chosen weights. This typically leads to artifacts, as shown in Figure 9. This problem can be avoided by thresholding the standard deviations before exponentiation. This also avoids the divide by zero problem for flat color regions, which have zero standard deviation. Typical values for the weight computation are $q = 8$ and $\tau_w = 0.02$.

3.3.2 Multi-scale Filtering

Multi-scale filtering is performed in a similar manner as the multi-scale local structure estimation. The pyramid is processed in coarse-to-fine order. For the coarsest level, the anisotropic Kuwahara filter is computed as usual. For the other levels, the upsampled filtering result from the previous level and the image data of the current level are merged using a linear combination:

$$\tilde{f}^k = \beta^k f^k + \Big(1 - \beta^k\Big) \bar{f}^{k+1}$$

Here, $\tilde{f}^k$ denotes the merged result, f^k is the original image data of the current level and $\bar{f}^{k+1}$ is the upsampled filtering result from the previous level. The merged result is now used to calculate the structure tensor of the current level as already described. Finally, the merged result is processed using the single-scale anisotropic Kuwahara filter. This process is then repeated until the finest level of the pyramid is reached. The weighting factor β^k is defined based on the standard deviation s^k by:

$$\beta^k = \mathrm{clamp}\Big(s^k \cdot p_s\,(p_d)^k - \tau_v,\; 0, 1\Big)$$

Additional user control is provided through parameters p_s and p_d. Parameter p_s applies to all levels in a uniform way, while parameter p_d takes scale into account. In order to account for small standard deviations due to noise, threshold parameter τ_v is provided. Typical values for these parameters are $p_s = 0.5$, $p_d = 1.25$, and $\tau_v = 0.1$.

Calculation of the standard deviation is computationally expensive, and therefore an approximation is used. During the computation of the anisotropic Kuwahara filter, the standard deviations s_i for each sector are computed. Since the sum of all weighting functions w_i is equivalent to a Gaussian, the sum of the thresholded standard deviations

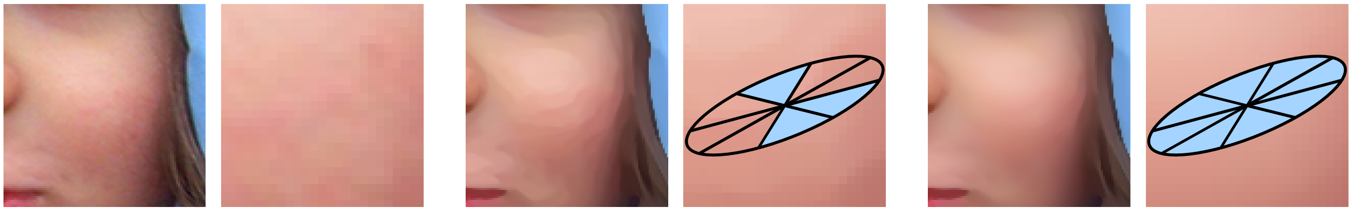
$$s_{\max} = \sum_{i=0}^{N-1} \max\big(\tau_w, \|s_i\|\big)$$

is an approximation of for the local standard deviation that can be easily computed during filtering at negligible computational cost. Because the standard deviation is required for the merging process before the actual computation of the anisotropic filter, the approximate standard deviation $s_{\max}$ is stored during the filtering as an additional result. This can be done, for example, by storing it in the alpha channel of the filter result. At each level of the pyramid, the approximate standard deviation from the previous level is then upsampled and used to calculate the weighting factor β_k (Figure 10).

Upsampling is performed with bilinear interpolation; typically, there is no benefit from using more sophisticated upsampling techniques. This might be surprising at first glance, especially since one might expect that usage of a better upsampling filter whould result in sharper color boundaries. However, the approximate standard deviation is high for pixels close to edges, and therefore the weight β^k is one. Thus, pixels close to the edges will be overridden during the merging process.

4 Results

The proposed technique was implemented using C++ and CUDA. The GLSL implementation of the anisotropic Kuwahara filter presented in [Kyprianidis et al. 2010a] was used as the initial starting point. The implementation does not make use of special features of CUDA. Instead, pitch linear memory and textures are used. Therefore, similar results should be achievable using shading languages, such as GLSL, Cg, or HLSL. Using an NVIDIA GTX 580 graphics card, processing an image with resolution 512 × 512 takes approximately 42 milliseconds. Image content at HD 720p (1280 × 720) resolution takes approximately 150 milliseconds to process.

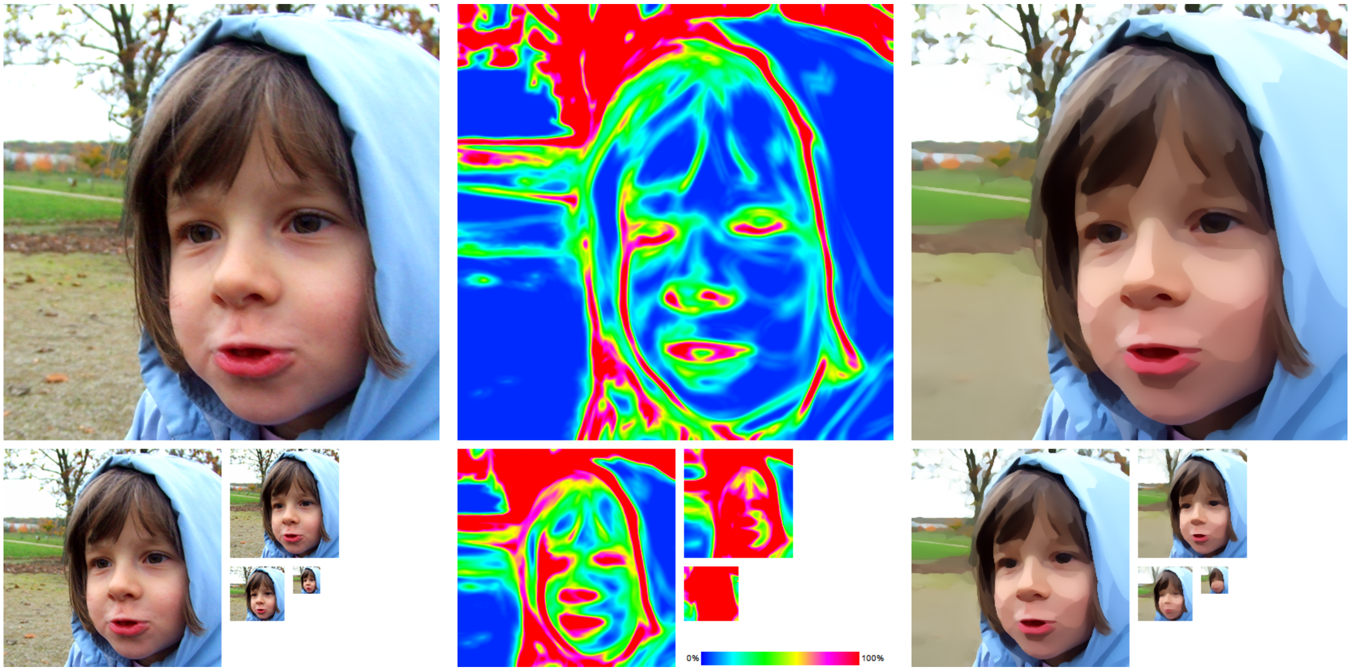


(a) *Original image*

(b) *Without thresholding*

(c) *With thresholding*

Figure 9: *Comparison of weight computation with and without thresholding.*



(a) *Pyramid of original image data*

(b) *Upsampled approximation to local standard deviation*

(c) *Merged and filtered result for each level*

Figure 10: *Original image, approximation to standard deviation and output of the anisotropic Kuwahara filter for the different pyramid levels.*

A comparison with the bilateral filter [Tomasi and Manduchi 1998] is shown in Figure 11. As can be seen, the level of abstraction is inconsistent for the bilateral filter. For example, the fur above the nose is less abstracted than the fur at the neck. The single-scale anisotropic Kuwahara filter, on the other hand, provides a very consistent level of abstraction over the whole image. The multi-scale version provides a much stronger abstraction. While there is slightly less abstraction above the nose, the overall look is also consistent. However, some of the cat's whiskers are lost. Moreover, the outputs of the single- and multi-scale methods look a little bit washed out.

In Figure 12 a comparison with coherence-enhancing filtering [Kyprianidis and Kang 2011] is shown. The level of abstraction is quite similar to the single-scale anisotropic Kuwahara filter, but coherence-enhancing filtering creates output with stronger contrast, which is probably due to the shock filter. The output of the multi-scale approach looks very similar, but more detail has been removed, such as from the jackets and the background.

Two problematic cases, where the multi-scale method fails to produce good-looking results for the default parameters are shown in Figure 13 and Figure 14. In the first case, parts of the image look blurred, and the abstraction of the rocks in the background is inconsistent. Moreover, parts above the plant are blended with the ground. By adjusting parameters p_s and p_d, the blurring can be removed, but then the abstraction is also less strong. In the second case, the image is very difficult to abstract due to its high-frequency texture. The multi-scale method performs decently, but there are clearly noticeable artifacts near the color region boundaries, such as around the eyes and where the nose meets the face.

5 Conclusions

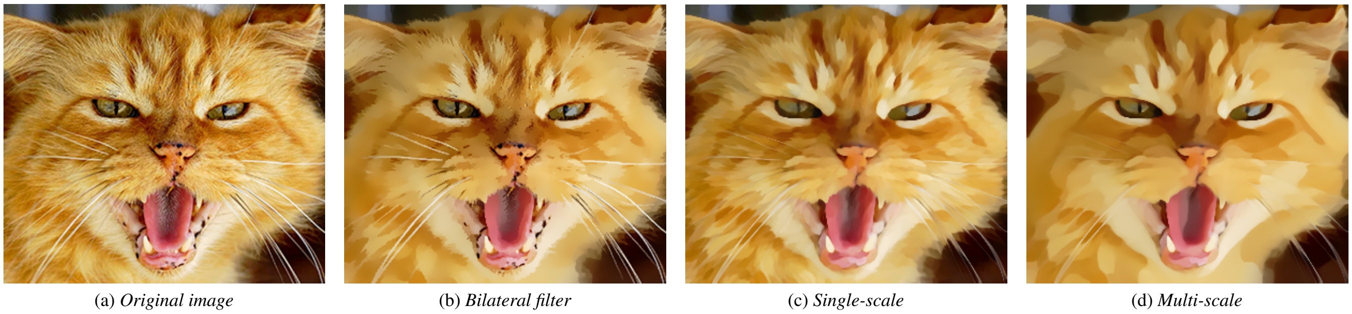
In this work, a generalization of the anisotropic Kuwahara filter to multiple scales was presented. Processing in a coarse-to-fine manner using a pyramid enabled aggressive abstraction and the creation of large homogeneous color regions. Moreover, the technique can be efficiently implemented on a GPU, and it creates temporally coherent output for video without further processing.

References

BHAT, P., ZITNICK, C. L., COHEN, M., AND CURLESS, B. 2010. GradientShop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics 29*, 2, 1–14.

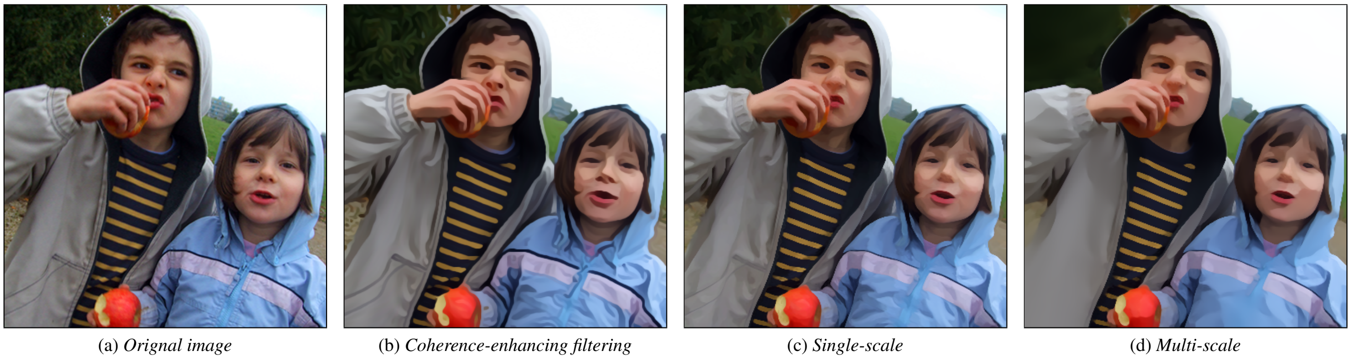
BLINN, J. 1989. Return of the Jaggy (high frequency filtering). *IEEE Computer Graphics and Applications 9*, 2, 82–89.

BROX, T., BOOMGAARD, R., LAUZE, F., WEIJER, J., WEICKERT, J., MRÁZEK, P., AND KORNPROBST, P. 2006. Adaptive Structure Tensors and their Applications. *Visualization and Processing of Tensor Fields*, 17–47.



(a) *Original image* (b) *Bilateral filter* (c) *Single-scale* (d) *Multi-scale*

Figure 11: *Comparison of anisotropic Kuwahara filter with bilateral filter [Tomasi and Manduchi 1998].*



(a) *Orignal image* (b) *Coherence-enhancing filtering* (c) *Single-scale* (d) *Multi-scale*

Figure 12: *Comparison of anisotropic Kuwahara filter with coherence-enhancing filtering [Kyprianidis and Kang 2011].*

BURT, P., AND ADELSON, E. 1983. The Laplacian Pyramid as a Compact Image Code. *IEEE Transactions on Communications 31*, 4, 532–540.

COLLOMOSSE, J. P., ROWNTREE, D., AND HALL, P. M. 2005. Stroke Surfaces: Temporally Coherent Non-photorealistic Animations from Video. *IEEE Transactions on Visualization and Computer Graphics 11*, 5, 540–549.

CUMANI, A. 1991. Edge detection in multispectral images. *CVGIP: Graphical Models and Image Processing 53*, 1, 40–51.

DECARLO, D., AND SANTELLA, A. 2002. Stylization and abstraction of photographs. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 769–776.

DI ZENZO, S. 1986. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing 33*, 1, 116–125.

FENG, X., AND MILANFAR, P. 2002. Multiscale principal components analysis for image local orientation estimation. In *Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, IEEE, 478–482.

GRAYSON, M. A. 1987. The heat equation shrinks embedded plane curves to round points. *Journal of Differential Geometry 26*, 2, 285–314.

HAEBERLI, P. 1990. Paint by numbers: abstract image representations. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 207–214.

HAYS, J., AND ESSA, I. 2004. Image and video based painterly animation. In *NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, 113–120.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 453–460.

JÄHNE, B., SCHARR, H., AND KÖRKEL, S. 1999. Principles of filter design. In *Computer Vision and Applications*, vol. 2. Academic Press, 125–151.

KANG, H., AND LEE, S. 2008. Shape-simplifying Image Abstraction. *Computer Graphics Forum 27*, 7, 1773–1780.

KANG, H., LEE, S., AND CHUI, C. K. 2007. Coherent Line Drawing. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, 43–50.

KANG, H., LEE, S., AND CHUI, C. K. 2009. Flow-Based Image Abstraction. *IEEE Transactions on Visualization and Computer Graphics 15*, 1, 62–76.

KUWAHARA, M., HACHIMURA, K., EHIU, S., AND KINOSHITA, M. 1976. Processing of ri-angiocardiographic images. In *Digital processing of biomedical imagesmages*, K. Preston and M. Onoe, Eds. Plenum Press, 187–203.

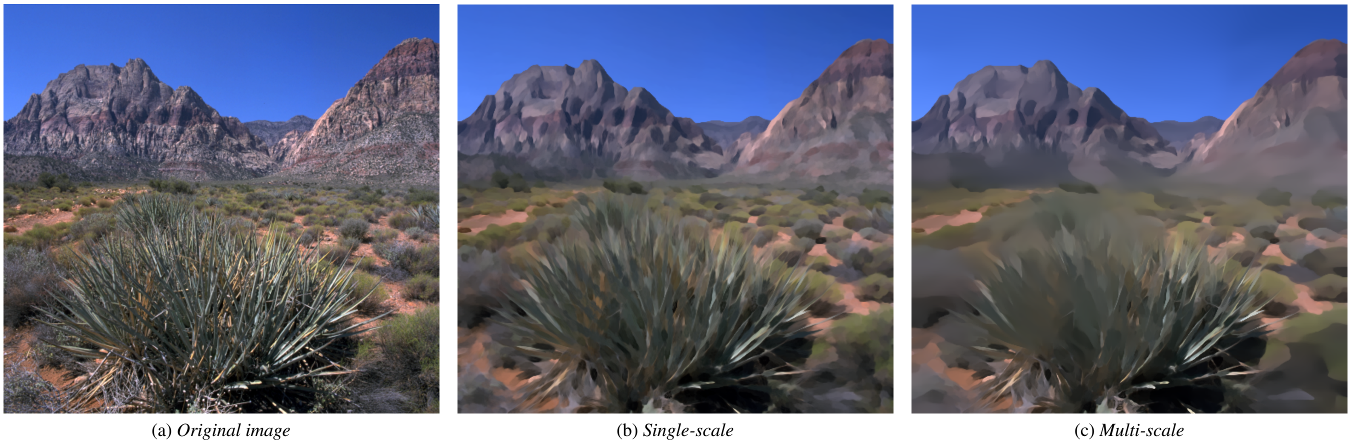
KYPRIANIDIS, J. E., AND DÖLLNER, J. 2008. Image Abstraction by Structure Adaptive Filtering. In *Proc. EG UK Theory and Practice of Computer Graphics*, 51–58.

KYPRIANIDIS, J. E., AND KANG, H. 2011. Image and Video Abstraction by Coherence-Enhancing Filtering. *Computer Graphics Forum 30*, 2.

KYPRIANIDIS, J. E., KANG, H., AND DÖLLNER, J. 2009. Image and Video Abstraction by Anisotropic Kuwahara Filtering. *Computer Graphics Forum 28*, 7, 1955–1963.

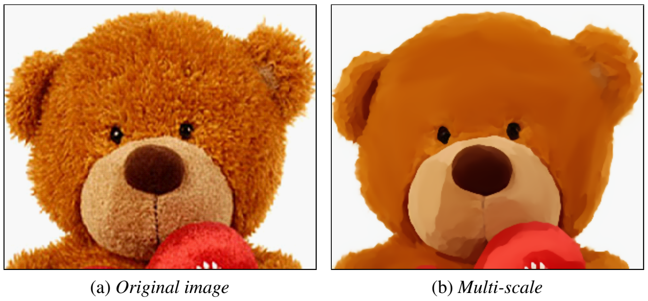
KYPRIANIDIS, J. E., KANG, H., AND DÖLLNER, J. 2010. Anisotropic Kuwahara Filtering on the GPU. In *GPUPro - Advanced Rendering Techniques*, W. Engel, Ed. AK Peters.

KYPRIANIDIS, J. E., SEMMO, A., KANG, H., AND DÖLLNER, J. 2010. Anisotropic Kuwahara Filtering with Polynomial Weighting Functions. In *Proc. EG UK Theory and Practice of Computer Graphics*.



(a) *Original image* (b) *Single-scale* (c) *Multi-scale*

Figure 13: *Example showing a problematic case, where the multi-scale approach creates an inconsistent level of abstraction.*



(a) *Original image* (b) *Multi-scale*

Figure 14: *The multi-scale anisotropic Kuwahara filter applied to an image with high-frequency texture.*

LEE, J. M. 2003. *Introduction to Smooth Manifolds*. Springer.

LINDEBERG, T. 1996. Scale-space: A framework for handling image structures at multiple scales. In *CERN School of Computing, Egmond aan Zee, The Netherlands, 8-21 September*.

MARDIA, K. V., AND JUPP, P. E. 1999. *Directional Statistics*, 2nd ed. Wiley.

ORZAN, A., BOUSSEAU, A., BARLA, P., AND THOLLOT, J. 2007. Structure-preserving manipulation of photographs. In *NPAR '07: Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, 103–110.

PAPARI, G., PETKOV, N., AND CAMPISI, P. 2007. Artistic Edge and Corner Enhancing Smoothing. *IEEE Transactions on Image Processing 16*, 10, 2449–2462.

SCHUMACHER, D. 1992. General filtered image rescaling. In *Graphics Gems III*, D. Kirk, Ed. Academic Press, July, 8–16.

SWAMY, K. 1973. On Sylvester's criterion for positive-semidefinite matrices. *IEEE Transactions on Automatic Control 18*, 3, 306.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral Filtering for Gray and Color Images. In *Proceedings International Conference on Computer Vision (ICCV)*, 839–846.

VAN DEN BOOMGAARD, R. 2002. Decomposition of the Kuwahara-Nagao Operator in terms of Linear Smoothing and Morphological Sharpening. In *Proc. of the 6th Interational Symposium on Mathematical Morphology*, CSIRO Publishing, 283–292.

WANG, J., XU, Y., SHUM, H.-Y., AND COHEN, M. F. 2004. Video tooning. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, 574–583.

WEN, F., LUAN, Q., LIANG, L., XU, Y.-Q., AND SHUM, H.-Y. 2006. Color sketch generation. In *NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, 47–54.

WILLIAMS, L. 1983. Pyramidal parametrics. *ACM SIGGRAPH Computer Graphics 17*, 3, 1–11.

WILSON, R., CLIPPINGDALE, S. C., AND BHALERAO, A. H. 1990. Robust estimation of local orientations in images using a multiresolution approach. In *Visual Communications and Image Processing '90*, SPIE, 1393–1403.

WINNEMÖLLER, H., OLSEN, S. C., AND GOOCH, B. 2006. Real-time video abstraction. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, 1221–1226.

YANG, G. Z., BURGER, P., FIRMIN, D. N., AND UNDERWOOD, S. R. 1996. Structure adaptive anisotropic image filtering. *Image and Vision Computing 14*, 2, 135–145.

Original photograph in Figure 13 courtesy Phillip Greenspun. Figure 11(a) is from flickr.com and kindly provided under Creative Commons license by pasma.