Chapter 5
Artistic Stylization by Nonlinear Filtering

Jan Eric Kyprianidis

5.1 Introduction

Digital image processing is a mature field providing a solid foundation for building artistic rendering algorithms. All image-based artistic rendering (IB-AR) approaches utilize image processing operations in some form to extract information or synthesize results. For instance, classical stroke-based rendering utilizes the image gradient for stroke placement. Nevertheless, few of the filters proposed for image processing are suitable in their original form, probably because in image processing, one is often concerned with the restoration and recovery of photorealistic imagery. By contrast, IB-AR generally aims for strong modification and simplification. As a result, researchers have often proposed specialized and adapted forms of existing techniques.

This chapter surveys a selection of nonlinear image processing algorithms that have been found to produce particularly interesting results. These techniques have in common that they perform some kind of edge-preserving simplification, often in combination with edge enhancement. In general, such an operation cannot be achieved by convolution filters, since these are fully determined by their impulse responses (i.e., applying a linear shift-invariant filter is equivalent to a convolution with the point spread function). By contrast, operations that preserve or selectively enhance edges must be guided by local (or even global) decisions based on the input source, leading directly to nonlinear operations that are not shift-invariant. Figure 5.1 illustrates a few examples of the techniques discussed in this chapter.

In contrast to approaches that emulate a specific artistic style, the techniques described here are based on heuristics developed through hands-on experience, showing that certain combinations of filters produce an artistic look. In some cases, the results obtained can be related to traditional styles such as cartoons, pen-and-ink

J.E. Kyprianidis (✉)
Hasso-Plattner-Institut, University of Potsdam, Potsdam, Germany
e-mail: kyprianidis@hpi.uni-potsdam.de

P. Rosin, J. Collomosse (eds.), *Image and Video-Based Artistic Stylisation*,
Computational Imaging and Vision 42, DOI 10.1007/978-1-4471-4519-6_5,

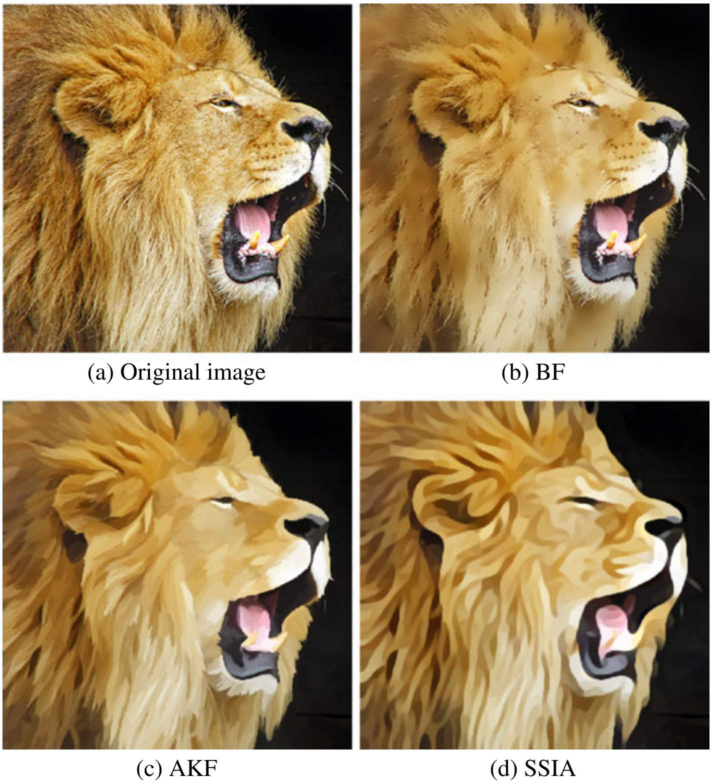


Fig. 5.1 Example showing popular edge-preserving/enhancing smoothing techniques often applied for image abstraction. (**b**) Bilateral filter [52]. (**c**) Anisotropic Kuwahara filter [34]. (**d**) Shape-simplifying image abstraction [24]. Tambako the Jaguar@flickr.com

illustrations, or watercolor paintings. In other cases, however, the connection is less obvious. The artistic look is thereby often achieved or further reinforced by taking the local image structure into account. Directional features and flow-like structures are considered pleasant, harmonic, or at least interesting by most humans [56]. 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 [58]. Particular examples include van Gogh and Munch, who have emphasized these features in their paintings.

Due to the local nature of image processing decisions, parallelization and GPU implementations of image filters are straightforward in most cases and often lead to real-time performance on modern multi-core CPUs and GPUs, making them practical for video processing—and applicable to footage that is otherwise challenging to parse (e.g., water, smoke, fur) using vision methods such as segmentation. This simplicity, however, comes at the expense of style diversity afforded by a higher-level interpretation of content.

The remainder of this chapter is organized as follows. In Sect. 5.2, the bilateral and difference of Gaussians filters are discussed. Together, these provide a powerful approach to the creation of cartoons, which will be discussed in detail. In Sect. 5.3, different variants of the Kuwahara filter are presented. Based on local image statistics, these are highly robust against high contrast noise, and driven by local image flattening, achieve a comparatively consistent level of abstraction across an image. In Sect. 5.4, techniques based on morphological operations are examined. Similar to the Kuwahara filter, these techniques effectively remove small-scale image features, and have been, for instance, successfully used to create watercolor renderings from images and videos. Section 5.5 presents techniques combining diffusion with sharpening. These allow for aggressive simplification while preserving sharp discontinuities. Finally, in Sect. 5.6, a brief overview of techniques operating in the gradient domain is given. Instead of directly operating on the image's gray or color values, these techniques operate on the gradient field of an image.

5.2 Bilateral Filter and Difference of Gaussians

A seminal work in image filtering-based NPR is the work of Winnemöller et al. [60] which, for the first time, presents a fully automatic pipeline for the creation of stylized cartoon renderings from images and video. Their pipeline employs the *bilateral* and *difference of Gaussians* (DoG) filter, and contains several influential ideas that other researchers later built upon. The bilateral filter smoothes low-contrast regions while preserving high-contrast edges, and may, therefore, fail for high-contrast images, where either no abstraction is performed or relevant information is removed because of the parameters chosen. In addition, the bilateral filter also often fails for low-contrast images, where typically too much information is removed. Moreover, iterative application of the bilateral filter may blur edges, resulting in a washed-out look (Fig. 5.1(b)). To some extent, these limitations can be alleviated by overlaying the output of the bilateral filter with outlines (e.g., generated with the DoG filter). Accordingly, the bilateral filter is rarely applied independently. Although the DoG filter can be used independently, preprocessing with the bilateral filter can often reduce artifacts caused by noise in the image. We start with a review of the bilateral and DoG filters, followed by a description the cartoon pipeline built from them.

5.2.1 Bilateral Filter

The *bilateral filter* is a well-known edge-preserving smoothing filter first introduced by Aurich and Weule [4], popularised by Tomasi and Manduchi [52]. A detailed review of the bilateral filter can be found in the survey by Paris et al. [45], which

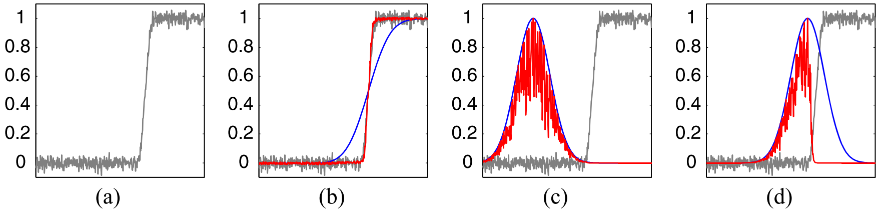


Fig. 5.2 Illustration of the working principle of the bilateral filter. (**a**) Noisy input signal (*gray*). (**b**) Result of the convolution with Gaussian (*blue*) and bilateral (*red*) filter. Note how the Gaussian kernel blurs the signal, while the bilateral filter keeps the sharp transition. In (**c**) and (**d**) the local filter kernel profiles of the Gaussian filter (*blue*) and bilateral filter (*red*) are shown at two different positions. The local filter kernel of the Gaussian filter does not depend on the signal (shift invariance) and is the same in both cases. The bilateral filter adapts its local filter kernel to the signal and thereby limits smoothing across the transition

also discusses various applications. For a given image I and position x_0 the bilateral filter is defined by

$$\frac{\sum_{x\in\Omega(x_0)} I(x)\,\overbrace{k_d(\|x-x_0\|)}^{\text{domain weight}}\,\overbrace{k_r(\|f(x)-f(x_0)\|)}^{\text{range weight}}}{\sum_{x\in\Omega(x_0)} k_d(\|x-x_0\|)k_r(\|f(x)-f(x_0)\|)} \tag{5.1}$$

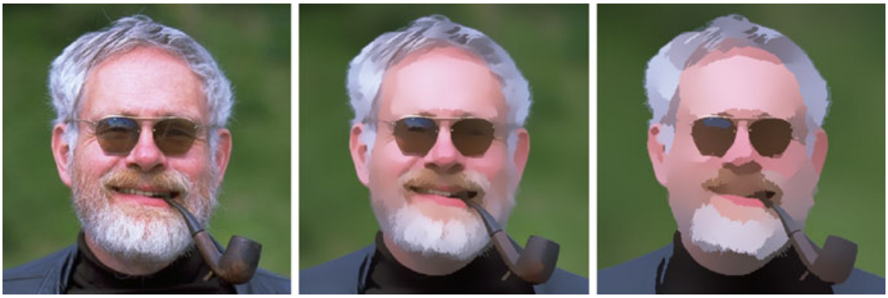
where $\Omega(x_0)$ denotes a sufficiently large neighborhood of x_0, and k_d and k_r are two weighting functions. The *domain weight* given by k_d is based on the spatial distance from the filter origin x_0, whereas the *range weight* given by k_r is based on the distance between the image's values at the corresponding positions. Typically, for both weighting functions a one-dimensional Gaussian

$$G_\sigma(r) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma}r^2\right) \tag{5.2}$$

is chosen, but other choices are possible. If k_d is chosen as Gaussian and $k_r \equiv 1$, then the bilateral filter simplifies to the Gaussian filter. The bilateral filter smoothes regions of similar color, while regions with detail are preserved. For instance, if the local neighborhood of a pixel contains an edge, then pixels on the opposite side of the edge receive a low and all others a high weight, resulting in the preservation of the edge (Fig. 5.2).

By using a suitable metric for the computation of the range weight, the bilateral filter extends naturally to color images. For instance, a possible choice is to use the Euclidean metric in RGB color space. Another choice, proposed by Tomasi and Manduchi [52], is using the Euclidean metric in CIELAB color space [62], which is known to correlate with human perception for short distances. Winnemöller et al. [60] and subsequent work adopted this approach.

If domain and range weight are chosen to be Gaussians, increasing the standard deviation of the domain weight generally does not lead to a stronger abstraction effect. Moreover, increasing the range weight results, in most cases, in blurred edges.



(a) Original image (b) 4 iterations (c) 25 iterations

Fig. 5.3 An iterative application of the bilateral filter smoothes the image while preserving edges, achieving a strong simplification effect. Original image courtesy of Philip Greenspun

Instead, to achieve a cartoon-like effect, it is better to apply multiple iterations of the bilateral filter (Fig. 5.3). This was already noted by Tomasi and Manduchi [52], and can be explained theoretically by the connection of bilateral filtering to anisotropic diffusion [5].

A limitation of the bilateral filter for practical applications, especially in the case of real-time processing, is that the direct evaluation of Eq. (5.1) is computationally expensive. For a local neighborhood with radius r the complexity is $O(r^2)$, which means that linear growth of the neighborhood leads to quadratic growth in computational costs. In contrast to the Gaussian filter, the bilateral filter is not separable, since it depends on local image values. Nevertheless, in the context of video compression, Pham and van Vliet [47] were able to show that for small filter sizes a separable implementation of the bilateral filter (Fig. 5.4(b)) can provide reasonable results. Their approach was adopted in the original cartoon pipeline by Winnemöller et al. [60] and was a crucial factor for achieving real-time performance on consumer GPUs at that time. Since then, several other approaches have been developed, such as the bilateral grid [12], approaches that avoid redundant operations by using histograms [48], and recently an approach based on domain transfer and normalized convolution [17].

Of particular interest from the IB-AR perspective are approaches taking local structure of an image into account. Kyprianidis and Döllner [31] proposed a separable implementation of the bilateral filter aligned to the local orientation (Fig. 5.4(c)). The first pass filters in perpendicular direction, while the second pass filters parallel to the local orientation. The adaptation to the local orientation helps to avoid artifacts, and produces more coherent region boundaries. In addition, the filter shape can be adapted to the local image structure, since the parameters for each pass can be controlled individually on a per-pixel basis. For instance, by decreasing the size in the direction of the gradient and increasing it in the direction of the tangent, the overall filter shape becomes elliptic, leading to an enhancement of anisotropic structures. An even stronger enhancement of anisotropic structures can be achieved with the *flow-based bilateral filter* proposed by Kang et al. [26], where the second pass

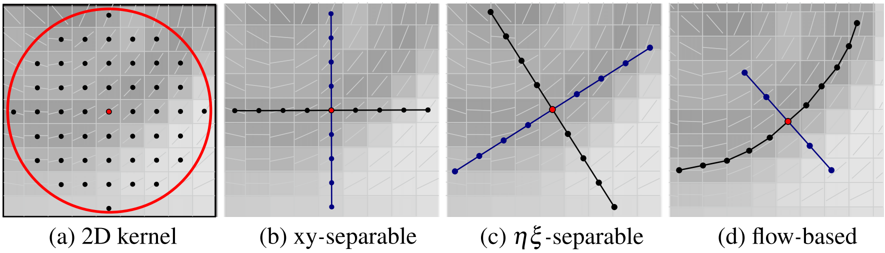


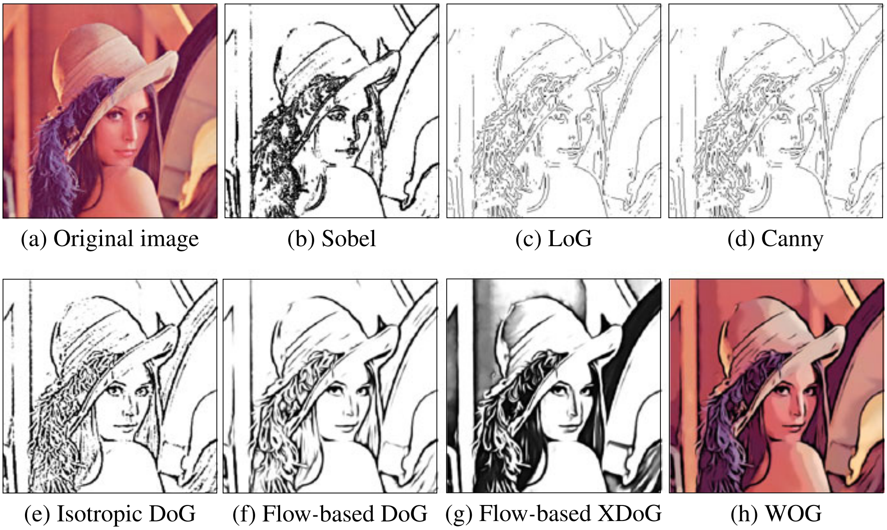
Fig. 5.4 Different variants of the bilateral filter. (**a**) A classical single pass bilateral filter performs a weighted average of a two-dimensional neighborhood. (**b**) Separable implementation with the first pass along the x-axis and second pass along the y-axis. (**c**) Orientation-aligned separable implementation with the first pass perpendicular and the second pass parallel to a vector field derived form the local structure, such as ETF or SST. (**d**) Separable implementation of the flow-based bilateral filter with the first pass perpendicular to the integral curve and the second pass along the integral curve defined by ETF or SST. Created by the author

performs an integration along the integral curves of a vector field given by smoothed tangents (Fig. 5.4(d)), similar to line integral convolution (LIC) [10].

For both approaches, a smooth vector field of high quality representing the local structure is critical (cf., Sect. 6.2.2). The two techniques that are known to produce vector fields of sufficient quality are the *edge tangent flow* (ETF) by Kang et al. [25, 26] and the *smoothed structure tensor* (SST) [31, 33, 34]. The ETF is essentially a bilateral filter, where the range weight has been specifically designed to measure the deviations between two vectors representing an axis. To obtain a reasonably smooth vector field, the ETF must be applied iteratively. The *structure tensor* is a well-known tool in computer vision, and is given by the outer product of the image gradients [9]. Smoothing the structure tensor and then performing an eigenanalysis corresponds to performing a principal component analysis on the gradient vectors. The major eigenvector can thus be interpreted as an averaged gradient and the minor eigenvector as an averaged tangent. In contrast to the EFT, the SST is suitable for speeding up the bilateral filter, since smoothing can be performed using a linear filter, such as a Gaussian filter, and eigenanalysis only involves solving a quadratic equation. Moreover, it is possible to define a coherence measure based on the SST's eigenvalues, which provides information about how anisotropic a local neighborhood is, and which can be used to adapt filters on a per-pixel basis. Local orientation estimation based on the SST fits well into the cartoon pipeline (Fig. 5.7), since the DoG filter, which will be discussed next, can also be significantly improved by adapting it to the local structure.

5.2.2 *Difference of Gaussians*

Early approaches to edge detection used simple approximations of the image gradient, such as the Prewitt and Sobel filter masks [49], and then thresholded the gradient



(a) Original image (b) Sobel (c) LoG (d) Canny

(e) Isotropic DoG (f) Flow-based DoG (g) Flow-based XDoG (h) WOG

Fig. 5.5 Example showing the output of different edge detection and stylization methods. (**a**) The original image (USC-SIPI image database). (**b**) The globally thresholded gradient magnitude of the Sobel filter. (**c**) Zero-crossing of the Laplacian of Gaussian [40]. (**d**) The output of the Canny edge detector [11]. (**e**) The thresholded DoG as proposed in Winnemöller et al. [60]. (**f**) The thresholded output of the separable implementation of the flow-based DoG [26, 31]. (**g**) The flow-based DoG with XDoG thresholding. The image is pre-processed with a bilateral filter to suppress noise. (**h**) Cartoon-style abstraction generated with bilateral and flow-based DoG filters using the WOG-pipeline

magnitude. However, due to the small size of the filter masks, such approaches are sensitive to noise and fail to detect edges at large scales. The Canny edge detector [11], therefore, combines first order differentials with appropriate smoothing, non-maximum suppression, and hysteresis thresholding. It is one of the most popular edge detectors for applications in computer vision. From an artistic point of view, however, the single pixel-wide edges it creates are typically not attractive and require further processing (Fig. 5.5(d)).

Another popular approach to edge detection based on second derivatives goes back to Marr and Hildreth [40]. For one-dimensional functions, a maximum of the gradient magnitude is equivalent to a zero-crossing in the second derivative. This also generalizes to two dimensions, where the second derivative perpendicular to the zero-crossing has to be considered. However, since this direction is unknown at computation time and would have to be estimated, Marr and Hildreth proposed to use the *Laplacian*

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{5.3}$$

which is rotationally invariant. While the Laplacian was known at that time to be useful for sharpening images [18], it has not been used for edge detection due to its high sensitivity to noise. Marr and Hildreth's key insight was to smooth the image before applying the Laplacian. This has two important effects. First, noise is reduced and the differentiation regularized. Second, the bandwidth is restricted, which means that the range of possible scales at which edges can occur is reduced. For the smoothing filter, a two-dimensional Gaussian

$$G_\rho(x) = \frac{1}{2\pi\rho^2} \exp\left(-\frac{\|x\|^2}{2\rho^2}\right) \tag{5.4}$$

was chosen, since it is known that it minimizes uncertainty, which simultaneously measures the spread of a function in the spatial and frequency domains. Since the Laplacian commutes with convolution, it follows that

$$\nabla^2(G_\rho * I) = \left(\nabla^2 G_\rho\right) * I \tag{5.5}$$

Thus, instead of applying smoothing and differentiation in sequence, both operations can be combined into a single operator $\nabla^2 G_\rho$, which can be symbolically computed and is known as the *Laplacian of Gaussian* (*LoG*). Marr and Hildreth, moreover, showed that the LoG operator can be approximated by the difference of two Gaussians

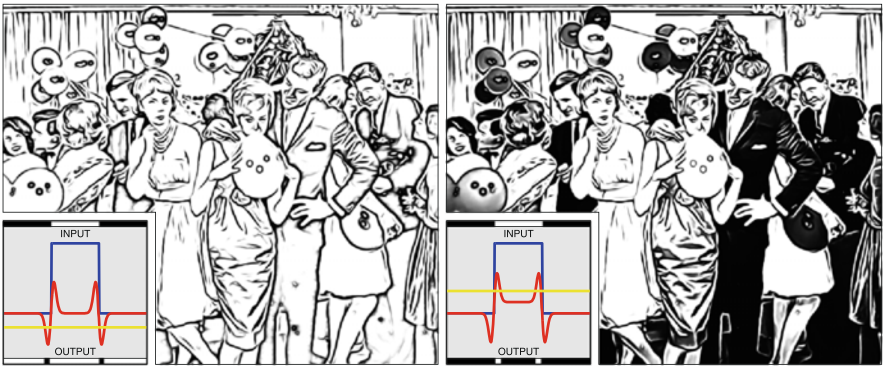
$$D_{\sigma,k}(x) = G_\sigma(x) - G_{k\sigma}(x) \tag{5.6}$$

with $k \approx 1.6$ being a good engineering solution. Results in biological vision, which showed that the ganglion cell receptive fields of cats can be modeled in this way [63], matched this result. This provided motivation for their approach and helped to popularize the technique. To extract edges from a LoG filtered image, the local neighborhood of a pixel is typically examined to detect the zero-crossings. This, however, again results in artistically questionable 1–2 pixel-wide edges (Fig. 5.5(c)) similar to those produced by the Canny edge detector (Fig. 5.5(d)). To achieve an artistically interesting effect, it turns out that simple thresholding works surprisingly well [19, 51, 60], which can be explained as follows. Let I denote a grayscale image. If we wish to generate a two-tone edge image, we essentially have two choices: Either we start with a white image and make certain image regions darker (i.e., set them to black), or we start with a black image and perform highlighting (i.e., set those regions to white). The DoG filter provides exactly this information by describing which high-frequency details have to be added to the low-pass filtered image $G_{k\sigma} * I$ to get

$$G_\sigma * I = G_{k\sigma} * I + D_{\sigma,k} * I \tag{5.7}$$

Hence, the sign of the DoG filter's response describes whether capturing the shape and structure of any nearby edges requires making each pixel darker or brighter than most of its neighbors.

The recently presented *XDoG* filter by Winnemöller [59] further refines the thresholding process by introducing additional parameters, and is defined by



(a) DoG

(b) XDoG

Fig. 5.6 Illustration of the XDoG thresholding scheme for a step edge (*blue*). The output of the DoG and XDoG operator before thresholding is shown in *red*; the threshold ε is indicated by the *yellow line*. (**a**) DoG has no tone mapping effect. Light regions are outlined. (**b**) XDoG allows for a tone mapping effect. Light regions get a black outline, dark regions receive a white outline. In both cases the flow-based variant was used. Original image by X-ray delta one@flickr.com

$$D_{\sigma,k,\tau}(x) = G_\sigma(x) - \tau \cdot G_{k\sigma}(x) \tag{5.8}$$

$$E_{\varepsilon,\varphi}(x) = \begin{cases} 1 & D_{\sigma,k,\tau}(x) > \varepsilon \\ 1 + \tanh(\varphi \cdot (D_{\sigma,k,\tau}(x) - \varepsilon)) & \text{otherwise} \end{cases} \tag{5.9}$$

The parameter σ controls the scale, whereas τ and ε control tone mapping and thresholding. By using tanh and the parameter φ, hard thresholding is avoided, which improves temporal coherence. In Fig. 5.5(e)–(f) and Fig. 5.6, a few examples with different parameter settings are shown. The relationship between the XDoG and the standard DoG can best be seen by rewriting Eq. (5.8) as

$$\begin{aligned} D_{\sigma,k,\tau}(x) &= G_\sigma(x) - \tau \cdot G_{k\sigma}(x) && (5.10) \\ &= (1-\tau) \cdot G_\sigma(x) + \tau \cdot D_{\sigma,k,\tau}(x) && (5.11) \end{aligned}$$

which shows that the XDoG approach is equivalent to a weighted average of the blurred image and standard DoG. Unfortunately, adjusting the parameters τ, φ, and ε is difficult, since they depend on each other and must be modified in concert. An alternative, which was proposed in [61], is to normalize the XDoG operator by dividing it by $\tau - 1$.

The XDoG filter is still relatively sensitive to noise. To some extent, ε can be used to reduce sensitivity, but a simple and highly effective approach is to apply 1–2 iterations of the bilateral filter before applying the XDoG filter (Fig. 5.7). An explanation for the high sensitivity can be given by looking at the decomposition of the LoG operator in the direction of the local gradient and tangent. The second derivative in the direction of the gradient contributes to the edge localization, while the one in the tangent direction merely increases the sensitivity to noise. This motivates considering detecting zero-crossings in the second derivative in the direction of the gradient. Such an edge detector was first proposed by Haralick [22], and also the maximum suppression of the Canny edge detector [11] is essentially equivalent to looking for a zero-crossing in the second derivative. A detailed discussion of the relationships between the Laplacian and directional derivatives has been given by Torre and Poggio [53].

The success of second derivative methods for edge detection suggests changing the XDoG filter from an isotropic to a directional operator. However, simply replacing the two-dimensional XDoG with its one-dimensional equivalent in the direction of the gradient does not lead to better results. In fact, the results are even worse. The reason for this is twofold: First, a one-dimensional XDoG is very sensitive to an accurate estimation of the gradient direction, which is typically performed using first order Gaussian derivative operators along the coordinate axes. The scale of these derivatives must be similar to the scale of the XDoG. For instance, if their scale is too large, the estimated gradient direction will, in general, not match the underlying image structure, which limits opportunities for noise suppression. Second, the missing regularization in the tangent direction further increases the sensitivity to noise.

The first work that addressed these issues and provided significantly improved quality over the isotropic DoG is the *flow-based difference of Gaussians* (FDoG) filter by Kang et al. [25]. In this work, the EFT was also initially introduced. It provided Kang et al. with a vector field closely aligned to the underlying image structure, and allowed them to derive an average gradient direction that is less affected by noise. The originally proposed FDoG performs steps along the integral curves of the EFT by using a Euler integration scheme. At each step, a one-dimensional DoG filter in the direction perpendicular to the integral curve is applied, and all these filter responses are accumulated by weighting them using a one-dimensional Gaussian. This accumulation performs regularization in the tangent direction, and shares a similarity with the hysteresis thresholding of the Canny edge detector. A separable implementation that achieves similar quality while being computationally less expensive and simpler to implement was presented by Kyprianidis and Döllner [31] and independently in a follow-up work by Kang et al. [26]. Similar to Fig. 5.4(d), the separable FDoG first performs a one-dimensional DoG, which is then followed by a second pass that performs line integral convolution with a Gaussian kernel. As in the case of the flow-based bilateral filter, the ETF can be replaced by the SST, which leads to a variant of the cartoon pipeline that delivers improved quality at a reasonable computational cost [31, 32]. To further increase the response of the FDoG, Kang et al. [25, 26] proposed to apply the FDoG iteratively by overlaying the previous FDoG response with the input image. While this results in stronger edges, it is also more sensitive to noise, and needs to be used with caution. The FDoG in combination with XDoG thresholding is very versatile. By properly adjusting parameters, a large variety of NPR effects can be created [59, 61].

Kang et al.'s [25] work provided new ideas in the field of IB-AR, and popularized the use of local structure information. It lead to several interesting results of work in areas such as image filtering [24, 26, 31, 33, 34], stippling [27, 50], and texture transfer [37]. Moreover, the flow-based XDoG is used by the ToonPAINT mobile application, which is discussed in Sect. 17.5.

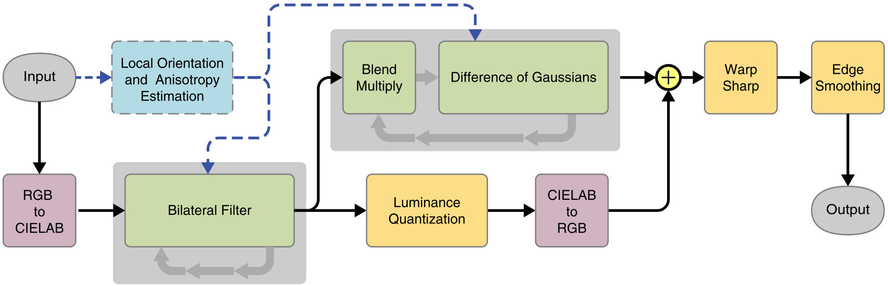


Fig. 5.7 The WOG-pipeline for the creation of a cartoon-like effect in modern generalized form [26, 31, 60]. Processing starts with the conversion of the input to CIELAB color space. Then, the input is iteratively abstracted by using a variant of the bilateral filter. After one or two iterations of the bilateral filter to suppress noise, outlines are extracted from the intermediate result using a variant of the DoG filter. Then more iterations of the bilateral filter are performed, typically up to four, with luminance quantization applied afterwards. DoG edges and the output of the luminance quantization are then composited, followed by an optional sharpening by warping and smoothing of the edges

5.2.3 Cartoon Pipeline

That multiple iterations of the bilateral filter lead to a cartoon-like effect was already noticed by Tomasi and Manduchi [52]. Motivated by this, Fischer et al. [16] applied the bilateral filter in the context of augmented reality to make virtual objects less distinct from the camera stream by applying stylization to the virtual and camera input. However, at that time computing the bilateral filter at full resolution was computationally too expensive. Due to this, Fischer et al. applied the bilateral filter at reduced resolution followed by upsampling, resulting in an inferior result. Winnemöller et al. [60] were faced with the same problem, but applied iteratively the separable implementation of the bilateral filter by Pham and van Vliet [47]. Although this brute force separation is prone to horizontal and vertical artifacts, it provides a reasonable tradeoff in terms of quality and speed, and enabled real-time processing on consumer GPUs of that time. In addition to the bilateral filter, Winnemöller et al. [60] added another processing step performing smooth luminance quantization. The quantization is applied in CIELAB space, with only the luminance channel being modified, creating a strong cartoon-like effect. The quantization is performed using a smooth step function whose steepness is chosen depending on the luminance gradient. This makes the output of the quantization less sensible to small changes in the input and increases temporal coherence when processing video frame-by-frame (cf., Sect. 11.3).

Creating artistic images using the DoG filter was also not new at that time. For instance, Sýkora et al. [51] used the thresholded output of the Laplacian of Gaussian, which is approximated by the DoG filter, to create outlines for colorizing hand-drawn black-and-white cartoons (see also Sect. 14.2 and Sect. 14.5.1), and Gooch et al. [19] used the DoG filter in combination with a model of brightness perception to create human facial illustrations. However, Winnemöller et al. [60] were the first to combine a bilateral and DoG filter into an effective pipeline.

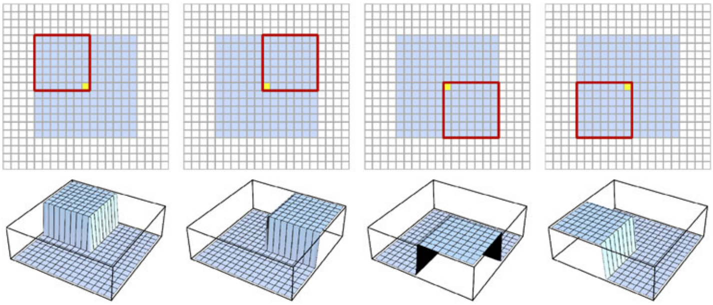
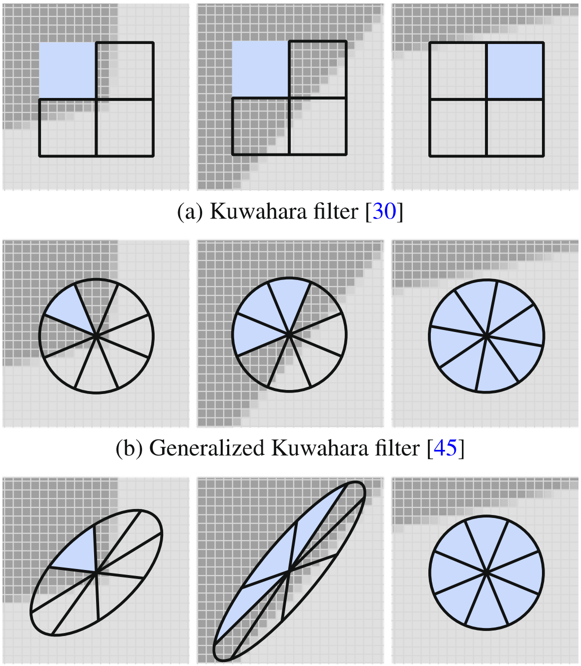


Fig. 5.8 The *top row* shows the four rectangular subregions used by the classical Kuwahara filter. The *bottom row* shows the weighting functions that can be used to describe the subregions—one over a specific subregion, or otherwise zero

A schematic overview of a modern generalized form of the pipeline proposed by Winnemöller et al. [60], hereafter referred to as the *WOG-pipeline*, is shown in Fig. 5.7. Input is typically an image, a frame of a video, or the output of a 3D rendering. In the original pipeline, the local orientation estimation step was not present; this step was added later to adapt the bilateral and DoG filters to the local image structure [25, 26, 31]. Also not present were the iterative application of the DoG filter, which was first proposed in [25], and the final smoothing pass to further reduce aliasing of edges. The introduction of the flow-based DoG filter significantly increased the quality of the produced outlines, and made the warp-based sharpening step of the original pipeline less important. Therefore, this step is typically not present in later work.

5.3 Kuwahara Filter

An interesting class of edge-preserving filters that perform comparatively well on high-contrast images are variants of the Kuwahara filter. Based on local area flattening, these filters properly remove detail in high-contrast regions and protect shape boundaries in low-contrast regions, resulting in a roughly uniform level of abstraction across the image. The *Kuwahara filter* [29] was initially proposed in the mid-1970s as a noise reduction approach in the context of biological image processing. The general idea behind it is to divide the local filter neighborhood into four rectangular subregions that overlap by one pixel (Fig. 5.8). For all subregions the variance, which is the sum of the squared distances to the mean, is computed, and the response of the filter is then defined as the mean of a subregion with minimum variance. As can be seen in Fig. 5.9(a), this avoids averaging between differently colored regions for corners and edges. However, for flat or homogeneous regions the variances of the different subregions are almost equivalent or even the same. A subregion with minimum variance is, therefore, generally not well-defined, and the selection highly unstable, especially in the presence of noise. For small filter sizes the Kuwahara



(a) Kuwahara filter [30]

(b) Generalized Kuwahara filter [45]

(c) Anisotropic Kuwahara filter [35]

Fig. 5.9 Comparison of different variants of the Kuwahara filter: (**a**) Classical Kuwahara filter with rectangular subregions; a single subregion is selected. (**b**) Generalized Kuwahara filter with sectors of a disc as subregions; multiple sectors are chosen. (**c**) Anisotropic Kuwahara filter, where the filter shape is derived from the local structure and divided into subregions; multiple filter responses are chosen. Note that the subregions in (**a**), (**b**) and (**c**) are defined to overlap slightly. Redrawn from [34]. © 2009 Blackwell Publishing. Used by permission

filter produces reasonable results. However, for IB-AR, comparatively large filter sizes are necessary to achieve an interesting abstraction effect, resulting in clearly noticeable artifacts. These are due to the unstable subregion selection process and the use of rectangular subregions. A more detailed discussion of limitations of the Kuwahara filter can be found in [44].

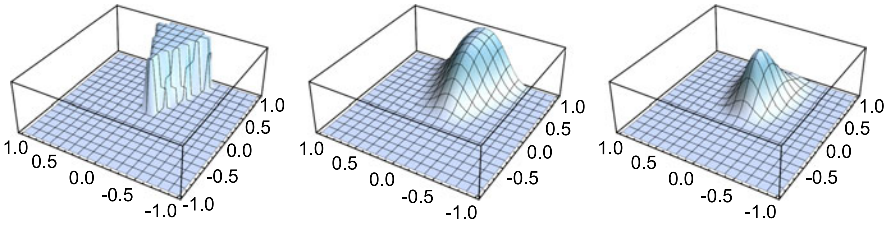
Several attempts have been made to address the limitations of the Kuwahara filter. The first work that provided an approach suitable for applications in IB-AR is the *generalized Kuwahara filter* by Papari et al. [44], which introduces two important ideas. First, the rectangular subregions are replaced with smooth weighting functions constructed over sectors of a disc in a way that their sum results in a 2D Gaussian. Neighboring weighting functions thereby have to overlap smoothly (Fig. 5.10). Using these weighting functions, for every sector the *weighted mean*

$$m_i(x_0) = \sum_{x \in \Omega(x_0)} w_i(x - x_0) \cdot f(x) \tag{5.12}$$

and *weighted variance*

$$s^2(x_0) = \sum_{x \in \Omega(x_0)} w_i(x - x_0) \cdot \big(f(x) - m_i(x_0)\big)^2 \tag{5.13}$$

can be computed. It should be noticed that if the weighting functions are chosen as characteristic functions of the rectangular subregions, as illustrated in Fig. 5.8, then



(a) χ_0 (b) $\chi_0 \star G_\rho$ (c) $(\chi_0 * G_\rho) \cdot G_\sigma$

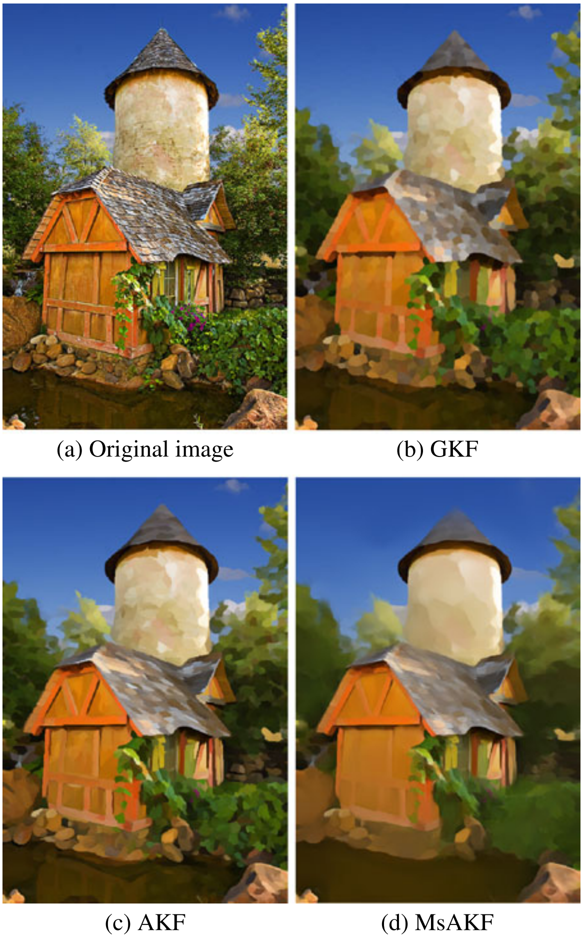
Fig. 5.10 Construction of the weighting functions of the generalized Kuwahara filter: (**a**) Characteristic function χ_0—one over the first sector, or otherwise zero. (**b**) Convolution of the characteristic function with a Gaussian $\chi_0 * G_\rho$ to create smooth transitions between different sectors. (**c**) Multiplication of the smoothed characteristic function with a Gaussian to create a smooth fall-off with increasing distance from the filter center: $(\chi_0 * G_\rho) \cdot G_\sigma$

the weighted mean and variance defined above are exactly the mean and variance of the subregions. Second, a new subregion selection method is defined. Instead of selecting a single subregion, the result is defined as the weighted sum of the weighted means, where the weights are based on the weighted variances, with sectors of low variance receiving a high weight and sectors of high variance receiving a low weight. This is achieved by taking the inverted weighted variance to the power of a user provided parameter q, and given by

$$\sum_{i \in N} m_i \cdot s_i^{-q} \Bigg/ \sum_{i \in N} s_i^{-q} \tag{5.14}$$

where N is the number of sectors. In Fig. 5.9(b), the behavior of the generalized Kuwahara filter is illustrated for different local neighborhoods. As can be seen, for corners and edges the filter adapts itself to the neighborhood, thus avoiding blurring across region boundaries. In homogeneous regions, the variances are similar, resulting in similar weights, which makes the filter approximate a Gaussian. In flat and smooth regions, the variances are very small and sensitive to noise, resulting in a poorly approximated Gaussian. To avoid this, a simple solution is to threshold the variances [30].

For highly anisotropic image regions, the flattening effect applied by the generalized Kuwahara filter is typically too aggressive, resulting in blurred anisotropic structures. Moreover, pixels tend to form clusters proportional to the filter size. The *anisotropic Kuwahara filter* by Kyprianidis et al. [34, 35] addresses these issues by replacing the weighting functions defined over sectors of a disc by weighting functions defined over ellipses, as shown in Fig. 5.9(c). By adapting shape, scale, and orientation of these ellipses to the local structure of the input, artifacts are avoided. In addition, directional image features are better preserved and emphasized, resulting in overall sharper edges and the enhancement of anisotropic image features (Fig. 5.1(c)). The local structure is estimated using the SST, where the coherence measure derived from the eigenvalues is used to define the eccentricity of the ellipse. A further modification has been presented in [36], wherein new weighting



(a) Original image

(b) GKF

(c) AKF

(d) MsAKF

Fig. 5.11 Examples created using different variants of the Kuwahara filter: (**a**) Original image courtesy of chefranden@flickr.com. (**b**) Generalized Kuwahara filter (2 iterations) [44]. (**c**) Anisotropic Kuwahara filter (2 iterations) [34]. (**d**) Multi-scale anisotropic Kuwahara filter [30]

functions based on polynomials that can be evaluated directly during the filtering process are defined.

The level of abstraction achievable with the generalized and 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. Another 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 depends quadratically on the radius. The *multi-scale anisotropic Kuwahara filter* by Kyprianidis [30], therefore applies the anisotropic Kuwahara filter at 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. Figure 5.11 shows an example image processed with different variants of the Kuwahara filter.

5.4 Morphological Filters

Mathematical morphology (MM) provides a set-theoretic approach to image analysis and processing. Besides being useful the for extraction of object boundaries, skeletons, and convex hulls, it has been also applied successfully to many pre- and post-processing tasks. A good introduction to the subject, covering aspects of image processing and computer vision, is the tutorial by Haralick et al. [23]. Fundamental operations in MM are *dilation* and *erosion*. From these, a large number of other operators can be derived, most notably *opening*, defined as erosion followed by dilation, and *closing*, defined as dilation followed by erosion. For grayscale images, dilation is equivalent to a maximum filter and erosion corresponds to a minimum filter. Therefore, opening removes light image features by removing peaks, while closing removes dark features by filling holes. Applying opening and closing in sequence results in a smoothing operation that is often referred to as *morphological smoothing*, which, similar to a median filter, quite effectively suppresses salt-and-pepper noise, while being computationally less expensive. In fact, openings and closings are closely related to order-statistics filters. A further in-depth discussion of morphological filters and their relations to other image processing operators can be found in [38, 39].

In Bousseau et al.'s [7, 8] work on watercolor rendering (cf., Sect. 13.3.2.1), morphological smoothing is applied to simplify input images and videos before their heuristically defined rendering approach is applied. In the case of video, a spatio-temporal kernel is used, aligned to the motion trajectory derived from optical flow. Applying opening and then closing generally results in a darkened result. Since watercolor paintings typically have light colors, Bousseau et al. proposed swapping the order of the morphological operators and applying closing followed by opening (Fig. 5.12). Since opening and closing are dual to each other, this is the same as inverting the output of the usual morphological smoothing applied to the inverted image.

Papari and Petkov [43] described another technique, which applied morphological filtering in the context of IB-AR. Motivated by *glass patterns*, and similar to line integral convolution [10], they performed a one-dimensional dilation in the form of a maximum filter over noise along the integral curves defined by a vector field. In contrast to line integral convolution, this technique is more capable of producing thick piece-wise constant coherent lines with sharp edges, resulting in a stronger brush-like effect. Moreover, it can also be applied to color images, by using the location of the first maximum noise value along the integral curve as a look-up position.

Some morphological operators (e.g., with convex polygonal structuring element) can be efficiently implemented by using distance transforms [15]. Criminisi et al. [13] recently demonstrated that edge-sensitive smoothing based on the *generalized geodesic distance transform* (GGDT) can be used for the creation of cartoon-style abstractions. The image is first clustered into a fixed number of colors. Then for every pixel, the probability of the pixel's value belonging to

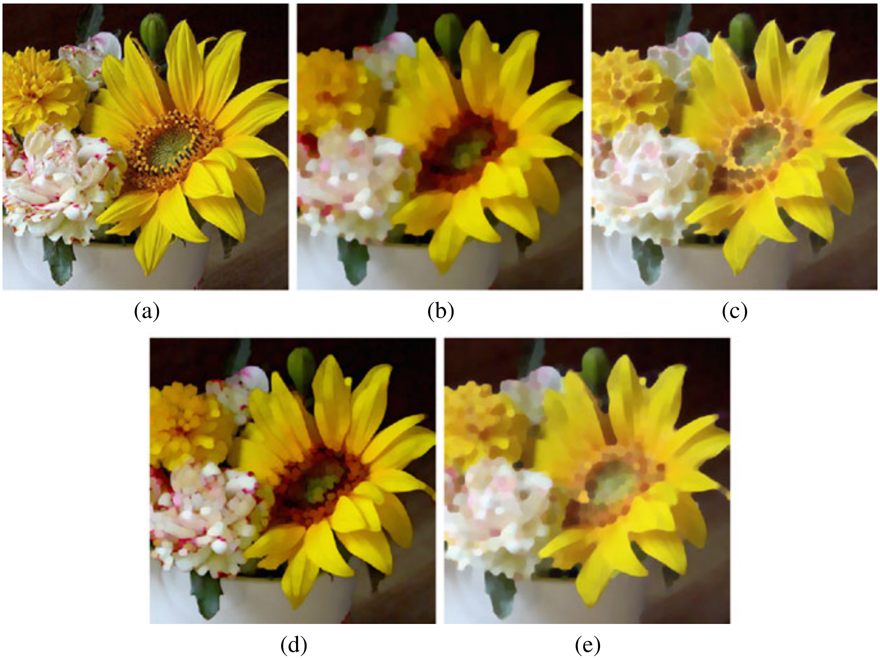


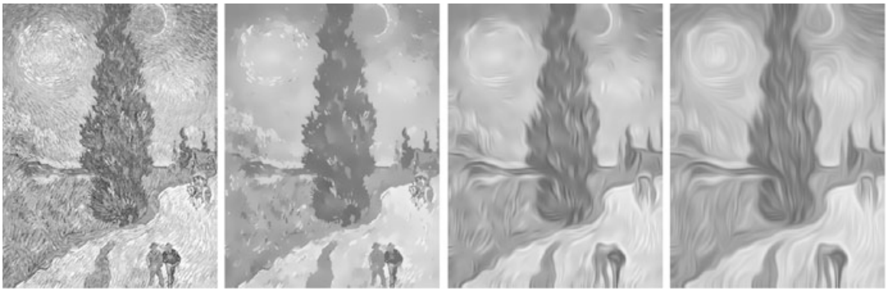
Fig. 5.12 Mathematical morphology operators. (**a**) Original image courtesy of PDPhoto.org. (**b**) Opening. (**c**) Closing. (**d**) Opening followed by closing. (**e**) Closing followed by opening: The morphological operator chosen by Bousseau et al. [7, 8]

a certain cluster is defined. These probabilities form a soft mask to which the GGDT is applied. The output is then defined as the weighted sum of the cluster's mean values, where the weights are defined based on the corresponding distances.

5.5 PDE-Based Methods

Methods based on *partial differential equations* (PDE) provide a powerful approach to image processing [3]. Interestingly, several local filtering approaches can be interpreted in terms of corresponding PDEs. For example, anisotropic diffusion is closely related to the bilateral filter [5], and PDE formulations for classical morphological processes have been established [55]. There is also a connection between PDEs and the Kuwahara filter. As shown by van den Boomgaard [54], the Kuwahara filter can be interpreted as a PDE with linear diffusion and shock filter terms.

In this section, *shape-simplifying image abstraction* by Kang and Lee [24] will be discussed. This technique applies a diffusion process to simplify the image, followed by shock filtering, which deblurs the image to maintain sharp edges at dis-



(a) Original image (b) AD (c) EED (d) CED

Fig. 5.13 Examples of different diffusion techniques. (**a**) van Gogh—Road with Cypress and Star. (**b**) Anisotropic diffusion [46]. (**c**) Edge-enhancing diffusion [55]. (**d**) Coherence-enhancing diffusion [55]

continuities. Before discussing it, we briefly review the concepts behind anisotropic diffusion and shock filters.

5.5.1 Anisotropic Diffusion

Let I be a grayscale image, then the solution of the heat equation

$$\frac{\partial u}{\partial t} = \Delta u = \operatorname{div}(\nabla u) \tag{5.15}$$

at a particular time t with initial condition $u(x, 0) = I(x)$ is given by convolution with a two-dimensional Gaussian having standard deviation $\sqrt{2t}$ [55]. To overcome the limitations of isotropic smoothing, Perona and Malik [46] added the regularization term

$$g\left(s^2\right) = \frac{1}{1 + \frac{s^2}{\lambda^2}} \quad (\lambda > 0) \tag{5.16}$$

to the heat equation that stops diffusion at the edges:

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(g\left(|\nabla u|^2\right) \nabla u\right) \tag{5.17}$$

This is known as *anisotropic diffusion*. Adding such penalization terms is a standard technique often found in PDE-based approaches. For instance, the edge-enhancing and coherence-enhancing diffusion techniques developed by Weickert [56] guide the diffusion using a tensor derived from the SST (Fig. 5.13). More details about anisotropic diffusion and other PDE-based image processing techniques can be found in the books by Weickert [55] and Aubert and Kornprobst [3].

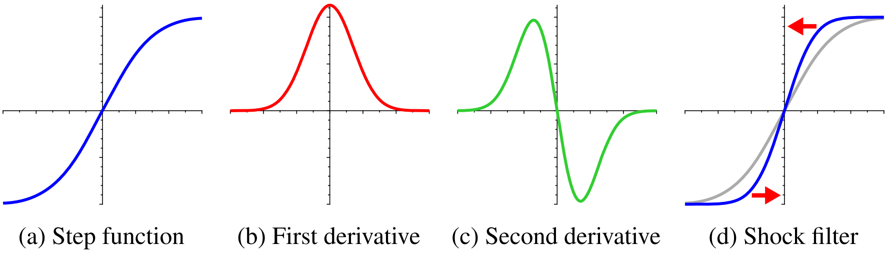

(a) Step function (b) First derivative (c) Second derivative (d) Shock filter

Fig. 5.14 Illustration of shock filtering and mean curvature flow. (**a**) A smooth step edge. (**b**) First derivative of the edge. (**c**) Second derivative of the edge. (**d**) A shock filter applies an dilation where the second derivative is positive and erosion where it is negative

5.5.2 Shock Filter

Osher and Rudin [42] were the first to study *shock filters* in image processing. The classical shock filter evolution equation is given by

$$\frac{\partial u}{\partial t} = -\operatorname{sign}\big(\mathcal{L}(u)\big)|\nabla u| \tag{5.18}$$

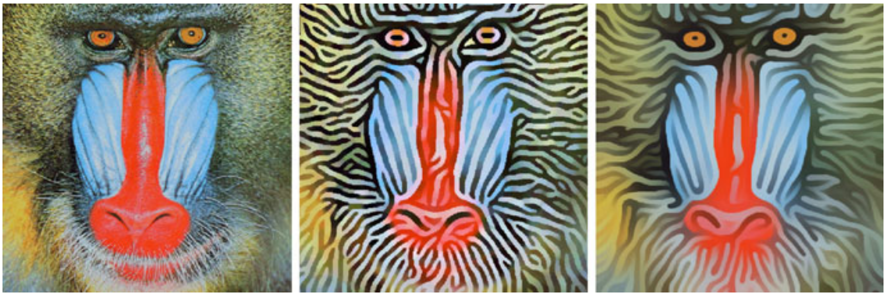
with initial condition $u(x,0) = I(x)$ and where $\mathcal{L}$ is a suitable detector, such as the Laplacian Δu or the second derivative in direction of the gradient:

$$u_{\xi\xi} = \frac{u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy}}{u_x^2 + u_y^2} \tag{5.19}$$

In the influence zone of a maximum, $\mathcal{L}(u)$ is negative, and therefore a local dilation, with a disc as the structuring element, is performed. Similarly, in the influence zone of a minimum, $\mathcal{L}(u)$ is positive, which results in local erosion. This sharpens the edges at the zero-crossings of Δu, as shown in Fig. 5.14. Shock filters have the attractive property of satisfying a maximum principle, and in contrast to unsharp masking, therefore do not suffer from ringing artifacts.

Instead of the second derivative in the direction of the gradient, also the second derivative in the direction of the major eigenvector of the SST can be used. This was first proposed by Weickert [57] and shares some similarity with the flow-based DoG discussed in Sect. 5.2.2. To achieve an higher robustness against small-scale image details, the input image can be regularized with a Gaussian filter prior to second derivative or SST computation [2]. As demonstrated in Fig. 5.15(b), this provides an aggressive simplification method. Equation (5.18) is typically implemented using a finite difference scheme. Thereby, $\mathcal{L}(u)$ can be approximated using central differences. Discretization of $|\nabla u|$ requires the use of an upwind scheme [3].

Shock filter can also be related to local neighborhood filters. Guichard and Morel [21] showed that the classical Osher–Rudin shock filter, with the Laplacian as the edge detector, corresponds asymptotically to a filter by Kramer and Bruckner [28], which replaces the current gray level value by either the minimum or maximum of the filter region, depending on which is closer to the current value.



(a) Original image (b) CESF (c) CEF

Fig. 5.15 Shock filters in conjunction with regularization provide an aggressive image simplification method. Original image (USC-SIPI image database). (**b**) Weickert's coherence-enhancing shock filter [57]. (**c**) A further refinement, applying curvature-preserving smoothing and shock filtering iteratively [33]

5.5.3 Mean Curvature Flow

Previously, Osher and Rudin [42], as well as Weickert [57], made comments about the artistic look of shock filtered results, but the work of Kang and Lee [24] was the first to apply diffusion in combination with shock filtering for targeting IB-AR. The *mean curvature flow* (MCF) diffusion method was chosen, which evolves isophote curves under curvature speed in normal direction, resulting in simplified isophote curves with regularized geometry. In contrast to other popular edge-preserving smoothing techniques, such as the bilateral or the Kuwahara filter, MCF smoothes not only irrelevant color variations while protecting region boundaries, but also simplifies the shape of those boundaries. The evolution equation of MCF is given by

$$\frac{\partial u}{\partial t} = \kappa |\nabla u| \quad \text{with } \kappa = \frac{u_x^2 u_{xx} - 2u_x u_y u_{xy} + u_y^2 u_{yy}}{(u_x^2 + u_y^2)^{3/2}} \tag{5.20}$$

denoting the curvature. Equation (5.20) can be implemented using central differences. A better approach, however, is to use a finite difference scheme with harmonic averaging [14].

MCF performs strong simplification of an image, but also creates blurred edges. Therefore, Kang and Lee [24] performed deblurring with a shock filter after some MCF iterations, which helps to keep important edges during the evolution (Fig. 5.16). From an artistic point of view, however, shock filtered MCF is typically still too aggressive, and does not properly protect directional image features (Fig. 5.17). Similar to Eq. (5.17), Kang and Lee therefore constrained the mean curvature flow by using the ETF to penalize diffusion that deviates from the local image structure. The evolution equation is given by

$$\frac{\partial u}{\partial t} = \left((1-r) + r \cdot \left| \left\langle \frac{E}{\|E\|}, \frac{\nabla u^{\perp}}{\|\nabla u\|} \right\rangle \right| \right) \|\kappa\| \tag{5.21}$$

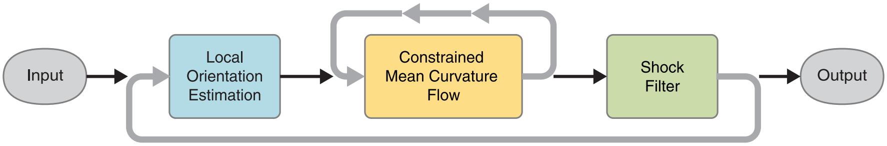
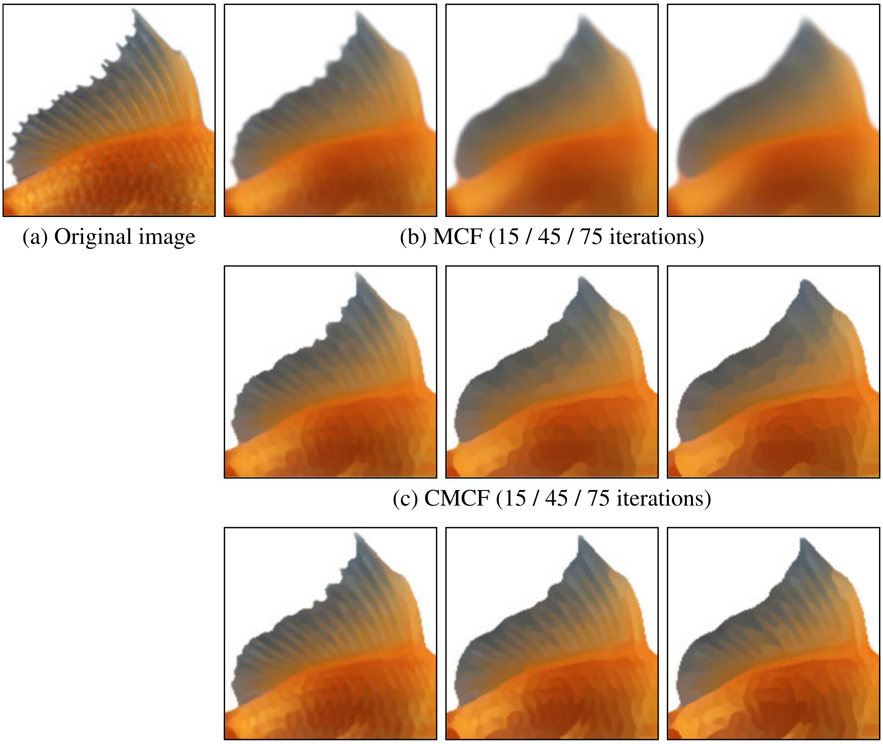


Fig. 5.16 Pipeline for shape-simplifying image abstraction [24]. Processing starts with the estimation of local orientation. Then multiple iterations of constrained mean curvature flow are applied, followed by shock filtering for deblurring. This process is repeated until the desired amount of abstraction has been reached


(a) Original image

(b) MCF (15 / 45 / 75 iterations)

(c) CMCF (15 / 45 / 75 iterations)

(d) CMCF + Shock (15 / 45 / 75 iterations)

Fig. 5.17 Comparison of mean curvature flow with/without shock filtering and constraint. (**a**) Original image (licensed by Getty images). (**b**) Mean curvature flow. (**c**) Mean curvature flow with shock filtering after 15 iterations. (**d**) Constrained mean curvature flow with shock filtering after 15 iterations. In all cases, a time step of 0.25 was used

where $\langle\cdot,\cdot\rangle$ denotes the per-pixel scalar product of EFT vectors and vectors perpendicular to the image gradients. The control parameter $r \in [0, 1]$ allows for blending between the unconstrained and the constrained MCF. Alternatively, instead of the ETF, the minor eigenvector field of the SST can be used.

MCF, and its constrained variant, contract isophote curves to points [20]. 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. In order to avoid these issues, Kyprianidis and Kang [33] combine curvature-preserving flow-guided smoothing and shock filter-based sharpening orthogonal to the flow, but instead of modeling the process by a PDE, approximations that operate as a local neighborhood filter are used (Fig. 5.15(c)). This makes the technique more stable and particularly suitable for per-frame video processing.

5.6 Gradient Domain Techniques

In recent years, *gradient domain* methods have become very popular in computer vision and computer graphics [1]. The basic idea behind such methods is to construct a gradient field representing the result. However, such constructed fields are rarely conservative, and therefore the result needs to be found as an approximation by solving an optimization problem. In the case of a best-fit in the least squares sense, this corresponds to solving Poisson's equation.

Orzan et al. [41] were the first to apply gradient domain image editing for IB-AR. By performing a scale-space analysis, they extracted a multi-scale Canny edge representation with lifetime and best scale information. This representation is then used to define the gradient field, and allows for image operations, such as detail removal and shape abstraction. Moreover, line drawings can be extracted from the multi-scale representation and overlaid with the reconstructed image. A limitation of the technique is that handling contrast is problematic and requires correction. Besides being computationally expensive, this technique is also known not to create temporal coherent output for video.

Bhat et al. [6] have presented a robust optimization framework that allows for the specification of zero-order (pixel values) and first-order (gradient values) constraints over space and time. The resulting optimization problem is solved using a weighted least squares solver. By using temporal constraints, the framework is able to create temporal coherent video output. The framework makes use of several computationally expensive techniques, such as steerable filters and optical flow, and is therefore currently limited to offline processing.

References

1. Agrawal, A., Raskar, R.: Gradient domain manipulation techniques in vision and graphics. In: ICCV Course (2007)
2. Alvarez, L., Mazorra, L.: Signal and image restoration using shock filters and anisotropic diffusion. SIAM J. Numer. Anal. **31**(2), 590–605 (1994). doi:10.1137/0731032
3. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations. Springer, Berlin (2006)

4. Aurich, V., Weule, J.: Non-linear Gaussian filters performing edge preserving diffusion. In: Proc. DAGM-Symposium, pp. 538–545 (1995)
5. Barash, D., Comaniciu, D.: A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. Image Vis. Comput. **22**(1), 73–81 (2004)
6. Bhat, P., Zitnick, C.L., Cohen, M.F., Curless, B.: GradientShop: a gradient-domain optimization framework for image and video filtering. ACM Trans. Graph. **29**(2), 10 (2010). doi:10.1145/1731047.1731048
7. Bousseau, A., Kaplan, M., Thollot, J., Sillion, F.X.: Interactive watercolor rendering with temporal coherence and abstraction. In: Proc. NPAR, pp. 141–149 (2006). doi:10.1145/1124728.1124751
8. Bousseau, A., Neyret, F., Thollot, J., Salesin, D.: Video watercolorization using bidirectional texture advection. ACM Trans. Graph. **26**(3), 104 (2007). doi:10.1145/1276377.1276507
9. Brox, T., Boomgaard, R., Lauze, F., Weijer, J., Weickert, J., Mrázek, P., Kornprobst, P.: Adaptive structure tensors and their applications. In: Visualization and Processing of Tensor Fields, pp. 17–47. Springer, Berlin (2006). doi:10.1007/3-540-31272-2_2
10. Cabral, B., Leedom, L.C.: Imaging vector fields using line integral convolution. In: Proc. SIGGRAPH, pp. 263–270 (1993). doi:10.1145/166117.166151
11. Canny, J.F.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**, 769–798 (1986). doi:10.1109/TPAMI.1986.4767851
12. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. ACM Trans. Graph. **26**(3), 103 (2007). doi:10.1145/1276377.1276506
13. Criminisi, A., Sharp, T., Rother, C., Pérez, P.: Geodesic image and video editing. ACM Trans. Graph. **29**(5), 134 (2010). doi:10.1145/1857907.1857910
14. Didas, S., Weickert, J.: Combining curvature motion and edge-preserving denoising. In: Proc. SSVM 2007. LNCS, vol. 4485, pp. 568–579. Springer, Berlin (2007). doi:10.1007/978-3-540-72823-8
15. Fabbri, R., Costa, L.D.F., Torelli, J.C., Bruno, O.M.: 2D Euclidean distance transform algorithms. ACM Comput. Surv. **40**(1), 2 (2008). doi:10.1145/1322432.1322434
16. Fischer, J., Bartz, D., Straber, W.: Stylized augmented reality for improved immersion. In: Proc. VR, pp. 195–202 (2005). doi:10.1109/VR.2005.1492774
17. Gastal, E.S.L., Oliveira, M.M.: Domain transform for edge-aware image and video processing. ACM Trans. Graph. **30**(4), 69 (2011). doi:10.1145/2010324.1964964
18. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice Hall, New York (2006)
19. Gooch, B., Reinhard, E., Gooch, A.: Human facial illustrations: Creation and psychophysical evaluation. ACM Trans. Graph. **23**(1), 27–44 (2004). doi:10.1145/966131.966133
20. Grayson, M.A.: The heat equation shrinks embedded plane curves to round points. J. Differ. Geom. **26**(2), 285–314 (1987)
21. Guichard, F., Morel, J.M.: A note on two classical enhancement filters and their associated PDE's. Int. J. Comput. Vis. **52**(2), 153–160 (2003). doi:10.1023/A:1022904124348
22. Haralick, R.M.: Digital step edges from zero crossing of second directional derivatives. IEEE Trans. Pattern Anal. Mach. Intell. **6**(1), 58–68 (1984). doi:10.1109/TPAMI.1984.4767475
23. Haralick, R.M., Sternberg, S.R., Zhuang, X.: Image analysis using mathematical morphology. IEEE Trans. Pattern Anal. Mach. Intell. **9**(4), 532–550 (1987). doi:10.1109/TPAMI.1987.4767941
24. Kang, H., Lee, S.: Shape-simplifying image abstraction. Comput. Graph. Forum **27**(7), 1773–1780 (2008). doi:10.1111/j.1467-8659.2008.01322.x
25. Kang, H., Lee, S., Chui, C.K.: Coherent line drawing. In: Proc. NPAR, pp. 43–50 (2007). doi:10.1145/1274871.1274878
26. Kang, H., Lee, S., Chui, C.K.: Flow-based image abstraction. IEEE Trans. Vis. Comput. Graph. **15**(1), 62–76 (2009). doi:10.1109/TVCG.2008.81
27. Kim, D., Son, M., Lee, Y., Kang, H., Lee, S.: Feature-guided image stippling. Comput. Graph. Forum **27**(4), 1209–1216 (2008). doi:10.1111/j.1467-8659.2008.01259.x

28. Kramer, H.P., Bruckner, J.B.: Iterations of a non-linear transformation for enhancement of digital images. Pattern Recognit. **7**(1–2), 53–58 (1975)
29. Kuwahara, M., Hachimura, K., Ehiu, S., Kinoshita, M.: Processing of ri-angiocardiographic images. In: Digital Processing of Biomedical Images, pp. 187–203. Plenum, New York (1976)
30. Kyprianidis, J.E.: Image and video abstraction by multi-scale anisotropic Kuwahara filtering. In: Proc. NPAR, pp. 55–64 (2011). doi:10.1145/2024676.2024686
31. Kyprianidis, J.E., Döllner, J.: Image abstraction by structure adaptive filtering. In: Proc. EG UK TPCG, pp. 51–58 (2008). doi:10.2312/LocalChapterEvents/TPCG/TPCG08/051-058
32. Kyprianidis, J.E., Döllner, J.: Real-time image abstraction by directed filtering. In: ShaderX7, pp. 285–302. Charles River Media, London (2009)
33. Kyprianidis, J.E., Kang, H.: Image and video abstraction by coherence-enhancing filtering. Comput. Graph. Forum **30**(2), 593–602 (2011). doi:10.1111/j.1467-8659.2011.01882.x
34. Kyprianidis, J.E., Kang, H., Döllner, J.: Image and video abstraction by anisotropic Kuwahara filtering. Comput. Graph. Forum **28**(7), 1955–1963 (2009). doi:10.1111/j.1467-8659.2009.01574.x
35. Kyprianidis, J.E., Kang, H., Döllner, J.: Anisotropic Kuwahara filtering on the GPU. In: GPUPro, pp. 247–264. AK Peters, Wellesley (2010)
36. Kyprianidis, J.E., Semmo, A., Kang, H., Döllner, J.: Anisotropic Kuwahara filtering with polynomial weighting functions. In: Proc. EG UK TPCG, pp. 25–30 (2010)
37. Lee, H., Seo, S., Ryoo, S., Yoon, K.: Directional texture transfer. In: Proc. NPAR, pp. 43–50 (2010). doi:10.1145/1809939.1809945
38. Maragos, P., Schafer, R.: Morphological filters—Part I: Their set-theoretic analysis and relations to linear shift-invariant filters. IEEE Trans. Acoust. Speech Signal Process. **35**(8), 1153–1169 (1987). doi:10.1109/TASSP.1987.1165259
39. Maragos, P., Schafer, R.: Morphological filters—Part II: Their relations to median, order-statistic, and stack filters. IEEE Trans. Acoust. Speech Signal Process. **35**(8), 1170–1184 (1987). doi:10.1109/TASSP.1987.1165254
40. Marr, D., Hildreth, R.C.: Theory of edge detection. Proc. R. Soc. Lond. B, Biol. Sci. **207**, 187–217 (1980)
41. Orzan, A., Bousseau, A., Barla, P., Thollot, J.: Structure-preserving manipulation of photographs. In: Proc. NPAR, pp. 103–110 (2007)
42. Osher, S., Rudin, L.I.: Feature-oriented image enhancement using shock filters. SIAM J. Numer. Anal. **27**(4), 919–940 (1990). doi:10.1137/0727053
43. Papari, G., Petkov, N.: Continuous glass patterns for painterly rendering. IEEE Trans. Image Process. **18**(3), 652–664 (2009). doi:10.1109/TIP.2008.2009800
44. Papari, G., Petkov, N., Campisi, P.: Artistic edge and corner enhancing smoothing. IEEE Trans. Image Process. **16**(10), 2449–2462 (2007). doi:10.1109/TIP.2007.903912
45. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: Bilateral filtering: theory and applications. Found. Trends Comput. Graph. Vis. **4**(1), 7–73 (2009). doi:10.1561/0600000020
46. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990). doi:10.1109/34.56205
47. Pham, T.Q., van Vliet, L.J.: Separable bilateral filtering for fast video preprocessing. In: Proc. ICME, pp. 454–457 (2005). doi:10.1109/ICME.2005.1521458
48. Porikli, F.: Constant time O(1) bilateral filtering. In: Proc. CVPR, pp. 1–8 (2008). doi:10.1109/CVPR.2008.4587843
49. Pratt, W.K.: Digital Image Processing, 3rd edn. Wiley, New York (2001). doi:10.1002/0471221325
50. Son, M., Lee, Y., Kang, H., Lee, S.: Structure grid for directional stippling. Graph. Models **73**(3), 74–87 (2011). doi:10.1016/j.gmod.2010.12.001
51. Sýkora, D., Buriánek, J., Žára, J.: Colorization of black-and-white cartoons. Image Vis. Comput. **23**(9), 767–782 (2005). doi:10.1016/j.imavis.2005.05.010
52. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. ICCV, pp. 839–846 (1998). doi:10.1109/ICCV.1998.710815

53. Torre, V., Poggio, T.A.: On edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **8**(2), 147–163 (1986). doi:10.1109/TPAMI.1986.4767769
54. van den Boomgaard, R.: Decomposition of the Kuwahara–Nagao operator in terms of linear smoothing and morphological sharpening. In: Proc. ISMM, pp. 283–292. CSIRO, Collingwood (2002)
55. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner, Leipzig (1998)
56. Weickert, J.: Coherence-enhancing diffusion of colour images. Image Vis. Comput. **17**(3), 201–212 (1999)
57. Weickert, J.: Coherence-enhancing shock filters. In: DAGM-Symposium, pp. 1–8. Springer, Berlin (2003). doi:10.1007/978-3-540-45243-0_1
58. Wikipedia: Expressionism—Wikipedia, The Free Encyclopedia (2012)
59. Winnemöller, H.: XDoG: Advanced image stylization with eXtended difference-of-Gaussians. In: Proc. NPAR, pp. 147–155 (2011). doi:10.1145/2024676.2024700
60. Winnemöller, H., Olsen, S.C., Gooch, B.: Real-time video abstraction. In: Proc. SIGGRAPH, pp. 1221–1226 (2006). doi:10.1145/1141911.1142018
61. Winnemöller, H., Kyprianidis, J.E., Olsen, S.C.: XDoG: an extended difference-of-Gaussians compendium including advanced image stylization. Comput. Graph. **36**(6), 740–753 (2012). doi:10.1016/j.cag.2012.03.004
62. Wyszecki, G., Stiles, W.S.: Color Science: Concepts and Methods, Quantitative Data and Formulae. Wiley-Interscience, New York (1982)
63. Young, R.A.: The Gaussian derivative model for spatial vision: I. Retinal mechanisms. Spat. Vis. **2**(4), 273–293 (1987). doi:10.1163/156856887X00222