

XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization[☆]

Holger Winnemöller^a, Jan Eric Kyprianidis^b, Sven C. Olsen

^aAdobe Systems, Inc.

^bHasso-Plattner-Institut

Abstract

Recent extensions to the standard difference-of-Gaussians (DoG) edge detection operator have rendered it less susceptible to noise and increased its aesthetic appeal. Despite these advances, the technical subtleties and stylistic potential of the DoG operator are often overlooked. This paper offers a detailed review of the DoG operator and its extensions, highlighting useful relationships to other image processing techniques. It also presents many new results spanning a variety of styles, including pencil-shading, pastel, hatching, and woodcut. Additionally, we demonstrate a range of subtle artistic effects, such as ghosting, speed-lines, negative edges, indication, and abstraction, all of which are obtained using an extended DoG formulation, or slight modifications thereof. In all cases, the visual quality achieved by the extended DoG operator is comparable to or better than those of systems dedicated to a single style.

Keywords: Cartoon Effects, Difference-of-Gaussians, Edge Detector, Hatching, Negative Edges, Threshold, Sketch, Woodcut

1. Introduction

It is well established that *edges* (i.e., significant intensity changes within a spatial neighborhood) play an important role in human and computer vision [1, 2, 3], and the computer vision community has contributed many fundamental results in the theory of edge detection [4, 5]. Given the importance of edges in popular art styles, such as line drawings or sketches, many stylization systems employ an edge detector as part of their processing pipeline. The Canny edge detector [5] is arguably the most popular such operator, due to its widespread use in the field of computer vision. However, its visual characteristics (lines whose thickness are independent of edge scale), are optimized for computer vision applications, and adapting them to the task of artistic stylization requires significant post-processing.

The *difference-of-Gaussians* (DoG) operator has been shown to yield aesthetically pleasing edge lines without post-processing, particularly when synthesizing line drawings and cartoons [6, 7, 8, 9]. Strictly speaking, the DoG is not an edge detector, and it cannot be directly compared to standard computer vision techniques, such as the Canny edge detector [5]. The two filters are related, however, as illustrated by Fig. 2(c-e).

Some previous works in artistic thresholding [6, 10] employ the DoG operator as part of a complex processing pipeline, though an equivalent effect can be achieved with a simple extension to the DoG operator itself (Sec. 4.1). We believe this situation arises because no previous work has investigated the

DoG operator in sufficient detail to reveal its significant potential for stylistic and practical applications.

To address this problem, our paper makes the following contributions:

Technical. We provide a detailed background of the DoG's theory and composition. We use many visual examples to tie these fundamentals to intuitions about the creative scope of the DoG operator, with specific emphasis on the effects of pushing its parameter values beyond their traditional ranges (Sec.2).

Effects. We demonstrate a number of subtle elements-of-style (effects) achievable with the extended DoG operator. We demonstrate ghosting, speed-lines, negative edges, indication, and abstraction in Sec. 3 and explain how these arise from the definitions given in Sec. 2.

Styles. We show many new DoG stylization results, most of which have previously been generated using complex, dedicated systems, but which are achieved in this paper within the continuous parameter space of a single operator. These results span traditional styles such as pencil-shading, pastel painting, hatching, and two tone black-and-white images such as woodcuts. The quality of most of our results is comparable to or better than those of the dedicated systems, yet easier and faster to compute. A small sample of these styles is shown in Fig. 1.

This paper represents an extended journal version of Winnemöller's NPAR 2011 paper [11]. Specifically, we significantly expand the background section to better explain the origin of the DoG operator and its relations to the broader theory of edges and edge detectors. We also expand our discussion of DoG operator extensions, allowing the reader to make

[☆]This is the authors' version of the work. The definitive version was published in *Computers & Graphics*, Vol. 36, Issue 6, 2012, pp. 720–753. doi: [10.1016/j.cag.2012.03.004](https://doi.org/10.1016/j.cag.2012.03.004).

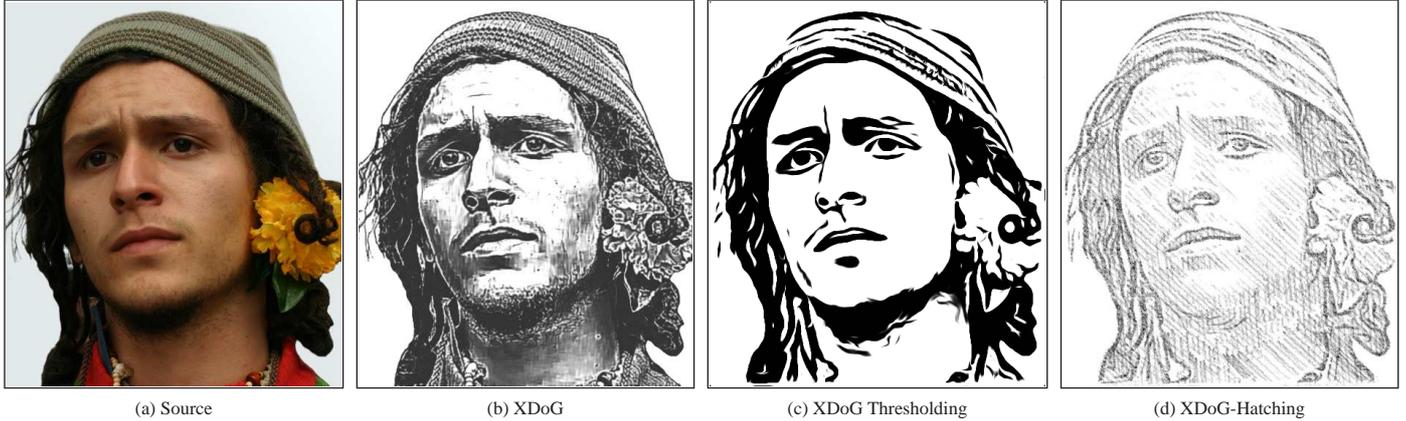


Figure 1: *Style variations*: All of these images are produced based on slight modifications to a *difference-of-Gaussians* operator. Source © Andrew Calder

a more informed decision when choosing between DoG variants and related edge detectors. We present a simple, reversible re-parameterization of [11], which offers a more intuitive mapping to known image-processing techniques (blurring, sharpening, etc.). We hope this re-parameterization will help readers to better understand the parameter space, and to more easily generate results similar to those presented in this paper. We discuss several tricks-of-the-trade particularly suitable for XDoG post-processing, which can further improve the quality of the results. Finally, we add more details about a stylistic variant of Woodcut, for which few prior works exist, but which can be generated conveniently with the XDoG operator.

2. Background

We believe the DoG operator has been under-utilized in stylization applications because its relationship to classical edge detectors is not widely understood. We therefore begin by explaining how shortcomings in early, primitive edge detectors lead to Marr and Hildreth’s [2] use of the DoG in their own edge detection system. We then present noteworthy connections to other disciplines like physics and biology, before summarizing recent applications of the DoG in the domain of image stylization.

2.1. Gradient-Based Edge Detection

The first approaches to edge detection focused on identifying pixels associated with high magnitude image gradients. A simple image gradient approximation would be generated by convolving the image with a small kernel, such as the Prewitt and Sobel filter masks [12]. Gradients having a magnitude above a certain threshold would be identified as “edges”. The small filter mask made these early edge detectors highly efficient, but also caused them to be very sensitive to noise. Thus, before approximating the gradients, the image would typically be pre-blurred to remove high-frequencies, thus reducing the noise in the results (Fig. 2(b)).

The Canny edge detector [5] provided several improvements over simple thresholding of the gradient magnitude. Smoothing and differentiation were combined into a single operator and a

non-maximum suppression scheme was applied to detect local maxima and thereby localize edges. Finally, hysteresis thresholding enhanced the coherence of detected edge lines and reduced false positives. These attributes of the Canny detector, along with its widespread availability, have made it one of the most popular edge detectors, particularly for computer vision applications. However, from an artistic point of view, the results of the Canny detector are rather unattractive, as can be seen in Fig. 2. Therefore, techniques employing it for artistic purpose typically perform additional processing, such as scale-space analysis [13] or curve fitting [14].

2.2. Laplacian-Based Edge Detection

Even before Canny suggested using non-maxima suppression to localize edges, Marr and Hildreth [2] proposed to similarly limit the results of an edge detector by identifying zero-crossings in the second derivative (Fig. 3). While the relationship between zero crossings of the second derivative and locally maximal gradients is straightforward in the one-dimensional case, generalizing the relationship to two dimensions requires that the second derivative be taken in the direction perpendicular to the zero-crossing. Unfortunately, this presents a chicken-and-egg problem, as we need to compute the second derivative in a direction that is yet to be determined by the result of the computation. Marr and Hildreth suggested circumventing this problem by using the *Laplacian*

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

However, as a second-order derivative, the Laplacian is highly sensitive to noise. As mentioned above, noise may be conveniently eliminated by pre-blurring with a low-pass filter. A filter with desirable frequency and scale-space properties [15] is the Gaussian smoothing filter

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right). \quad (1)$$

Here, x refers to a two-dimensional coordinate, and σ represents the standard deviation of the Gaussian distribution in the spatial domain (which is inversely proportional to the cut-off

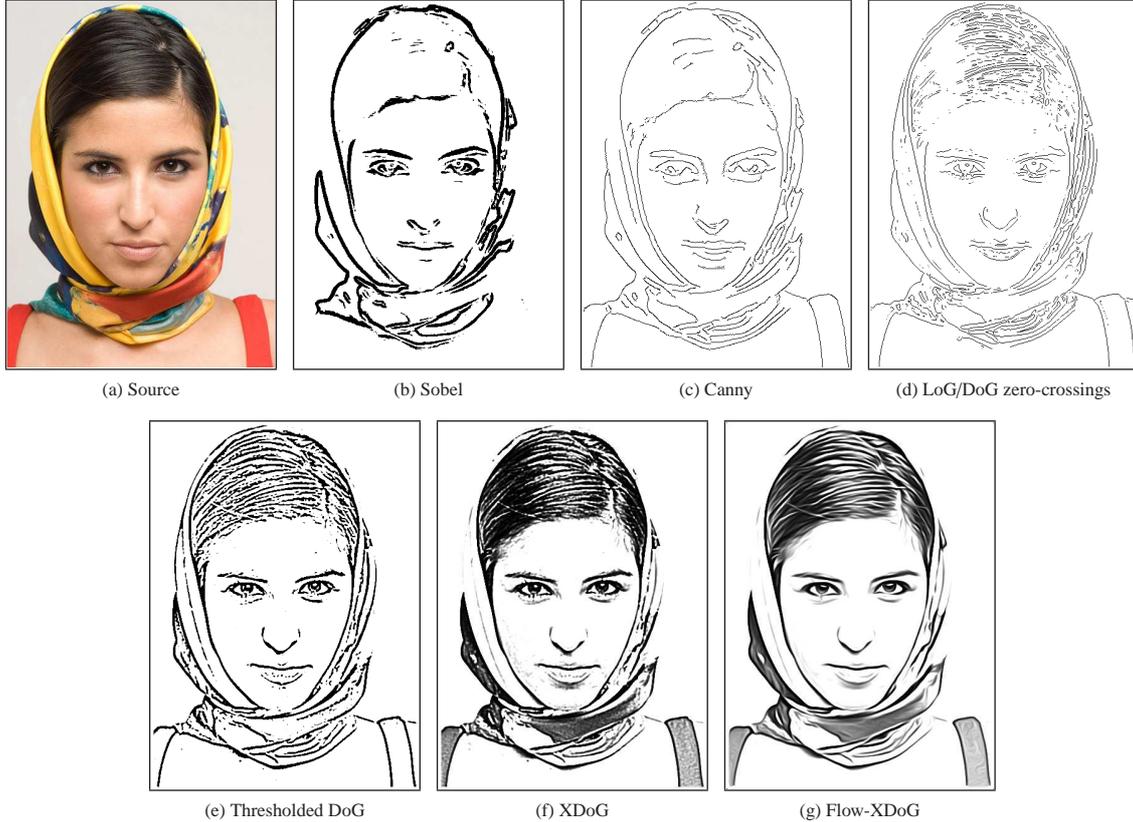


Figure 2: Comparison of popular edge detectors and difference-of-Gaussians based stylization. (b-e) For better comparison, all parameters are hand-tuned for optimal quality and mutual visual similarity; (f) Base parameters as (e), but demonstrating XDoG tonemapping; (g) flow-based version of (f). Source @Maryse Casol

frequency of a low-pass filter in the frequency domain). Since the Laplacian commutes with convolution, for an image I it follows that

$$\nabla^2(G_\sigma * I) = (\nabla^2 G_\sigma) * I,$$

where $*$ denotes the convolution operator. Thus, instead of applying smoothing and differentiation in sequence, both operations can be combined into a single operator $\nabla^2 G_\sigma$, which is known as the *Laplacian of Gaussian* (LoG). To extract edges from a LoG filtered image, the local neighborhood of a pixel is typically examined to detect the zero-crossings. However, as demonstrated in Fig. 2(d), this again results in artistically questionable 1-2 pixel-wide edges similar to those produced by the Canny edge detector. Nevertheless, the output of the LoG operator serves as a key component in the creation of the results in the bottom row of Fig. 2. The critical difference is that, rather than searching for zero crossings, thresholding is applied to the LoG response. Before explaining this in detail, we first discuss a fast approximation of the LoG, which is important for practical implementations and used throughout the rest of the paper.

2.3. Difference-of-Gaussians (DoG)

A limitation of the LoG is that it is computationally inefficient, since it is not separable. For this reason, Marr and Hildreth [2, App. B] proposed to approximate the LoG by the differences of two Gaussian functions (DoG), which are themselves separable. This approximation may be verified by looking at the difference-of-Gaussians with infinitesimally small

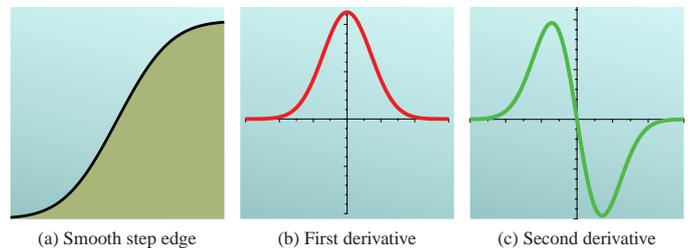


Figure 3: Edge localization: In the one-dimensional case, searching for a maximum in the first derivative is equivalent to finding a zero-crossing in the second derivative.

change in σ , corresponding to the differential quotient:

$$\lim_{k \rightarrow 1} \frac{G_{k\sigma}(x) - G_\sigma(x)}{k\sigma - \sigma} = \frac{\partial G_\sigma}{\partial \sigma} = \sigma \nabla^2 G.$$

Thus, we see that the *difference-of-Gaussians* filter

$$D_{\sigma,k}(x) = G_\sigma(x) - G_{k\sigma}(x) \approx -(k-1)\sigma^2 \nabla^2 G \quad (2)$$

approximates the negated scale-normalized Laplacian (as defined by Lindeberg [15]) up to a constant positive factor. The scale-normalization is a useful property of the DoG, since it ensures that the DoG response doesn't change when modifying the scale σ . Threshold values, for instance, can therefore be defined independent of scale. In this paper, we follow the commonly cited suggestion by Marr and Hildreth [2, App. B], to use $k = 1.6$ as a good engineering trade-off between accurate approximation and adequate sensitivity.

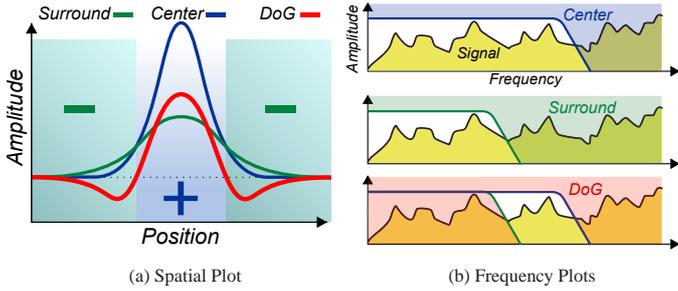


Figure 4: *DoG Composition*: (a) A wider *surround* Gaussian is subtracted from a narrower *center* Gaussian to produce the DoG trace; (b) Two low-pass filters of different cut-off frequency combine to produce a band-pass filter.

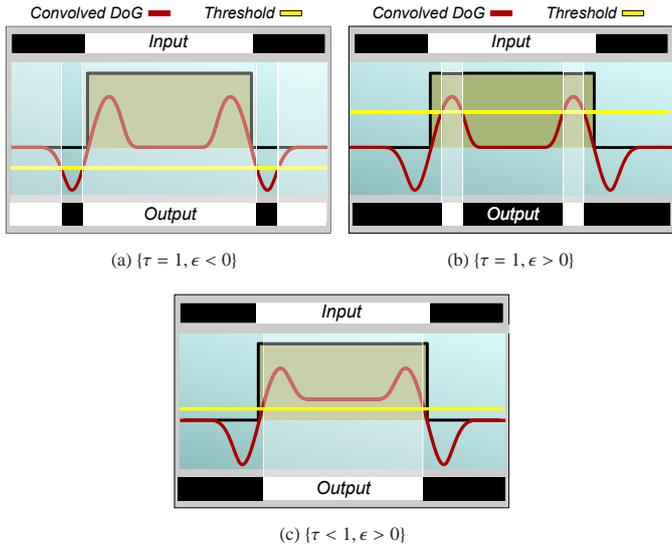


Figure 5: *XDoG Parameters*: Each subfigure shows (top) an input signal, (middle) a graph of a DoG (red) convolved with the input signal and a threshold level (yellow), and (bottom) the filter response output. (a) Unbiased, low threshold: typical edge detection. (b) Unbiased, high threshold: inverted edge detection. (c) Center-bias, high threshold: Can be used to effect luminance scaling.

Physical Interpretation. Understanding the origins of the DoG operator and its relations to the second-order properties of an image is instructive. However, the DoG’s ability to extract edge information from images may be explained even more simply by looking at the problem from a signal processing point-of-view. As mentioned above, a Gaussian filter is a *low-pass* filter. That is, it allows low spatial frequencies to pass, while attenuating or eliminating high spatial frequencies. Accordingly, the subtraction of two Gaussians with different σ creates a *band-pass* filter that attenuates all frequencies between the cut-off frequencies of the two Gaussians (Fig. 4b). A DoG filter will extract those image features falling within this characteristic frequency band [16]; and such features tend to correspond to edge lines.

DoGs in Neurobiology. While Rodieck [17] was among the first to quantitatively examine the neurophysiology of vision, including detection of features such as edges, it was Young [18] who found that certain retinal cells behaved exactly analogous to the *center-surround* activation mechanism originally proposed by Marr and Hildreth [2] and illustrated in Fig. 4(a).

In Young’s study, the *receptive field* of a cortical cell is modeled as an antagonistic system in which the stimulation of the *central* cell (blue) is inhibited by the simultaneous excitation of its *surrounding* neighbors (green). The combined response curve (red) can be modeled by subtracting two Gaussian distribution functions whose standard deviations are proportional to the spatial extent of the central cell and its receptive field [3].

2.4. Extended Difference-of-Gaussians (XDoG)

Comparing the two rows of images in Fig. 2 it becomes evident that edge *detection*, useful in computer vision, is qualitatively quite different from edge *enhancement* for stylistic and artistic applications. While the former is primarily concerned with the exact localization and extent of an edge, the latter is more appropriately focused on the weight (thickness) and structure (shape) of an edge.

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). Because it is a band-pass filter, the sign of the DoG response describes whether capturing the shape and structure of any nearby edges requires making each pixel darker or brighter than most of its neighbors. This is exactly the information we need to generate an ‘edge enhancement image’. Such an image may be formally defined as a thresholding of the DoG response, $T_\epsilon(D_{\sigma,k} * I)$, where,

$$T_\epsilon(u) = \begin{cases} 1 & u \geq \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The parameter ϵ is used to control the sensitivity to noise and is illustrated in Figures 5(a) and 5(b). Figure 2(e) demonstrates the effectiveness of the approach. Despite being comparatively simple, the result captures many important images features and is aesthetically pleasing.

In the context of computer vision, the word ‘edges’ is used to refer to the thin lines formed by locally maximal gradient points, such as the the DoG zero crossings shown in Fig. 2(d). However, in the context of image stylization, it is more natural to use the word ‘edges’ to refer to an image like the thresholded DoG shown in Fig. 2(e). As this paper focuses on stylization applications, we will use the term ‘edge image’ when referring to results like those in Fig. 2(e-g), though it should be noted that in the computer vision community, such images would not typically be said to contain ‘edges’.

The edge images generated by simple thresholding of the DoG are closely related to the biological models proposed by Young and others [18]. Inspired by those models, Winemöller et al. [7] generated edge images using a DoG variant in which the strength of the inhibitory effect of the larger Gaussian is allowed to vary, resulting in the following equation:

$$D_{\sigma,k,\tau}(x) = G_\sigma(x) - \tau \cdot G_{k\sigma}(x) \quad (4)$$

That modification made it possible to achieve a much wider range of stylistic effects, particularly after replacing the binary

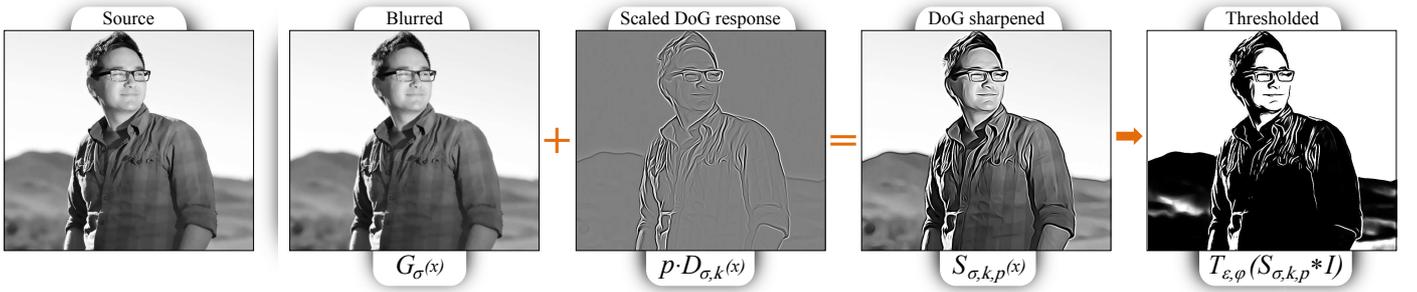


Figure 6: *Visualizing the XDoG*: The DoG response is combined with the blur result in order to create a sharpened image, as per Eq. (7). Thresholding yields a two-tone result. The images in this figure are generated using flow-guided filters, as discussed in Section 2.6. Exact parameter settings are supplied in Appendix A.

thresholding function T_ε with a continuous ramp:

$$T_{\varepsilon,\varphi}(u) = \begin{cases} 1 & u \geq \varepsilon \\ 1 + \tanh(\varphi \cdot (u - \varepsilon)) & \text{otherwise.} \end{cases} \quad (5)$$

Taken together, $T_{\varepsilon,\varphi}(D_{\sigma,k,\tau} * I)$ is referred to as the XDoG filter for a given image I . Figure 2(f) demonstrates how the base thresholded DoG is extended as a result of the soft thresholding and variable inhibition strength.

However, the XDoG is difficult to control. Increasing the sensitivity of the filter to edges typically requires adjusting τ , φ , and ε in concert. We can see the reason for this by decomposing $D_{\sigma,k,\tau}(x)$ as follows:

$$\begin{aligned} D_{\sigma,k,\tau}(x) &= G_\sigma(x) - \tau \cdot G_{k\sigma}(x) \\ &= (1 - \tau) \cdot G_\sigma(x) + \tau \cdot D_{\sigma,k}(x) \end{aligned} \quad (6)$$

This makes it clear that Eq. (4) is equivalent to a weighted average of the blurred image and the standard DoG. Notice that the average response of the standard DoG is zero, while the blurred image will have the same average brightness as the input image. Thus, the average brightness of $D_{\sigma,k,\tau} * I$ will decrease as τ increases. However, increasing τ is the only way to increase the weight of the edge emphasis lines. Thus, in order to create XDoG outputs having different edge emphasis strengths but the same average brightness, any adjustment to τ must be coupled with compensating changes to the soft thresholding parameters φ and ε .

2.5. Reparameterization of the XDoG

In order to simplify artistic control of the XDoG filter, a reparameterization having the following properties would be desirable: (1) Removal of the tight parameter-interdependency of the previous parameterization; (2) More intuitive parameters by mapping to known image processing operations, such as blurring or sharpening; (3) Invertibility, i.e., it should be possible to convert back-and-forth between the old and new parameter spaces. Fortunately, a parameterization that fulfills these requirements can be found by simply dividing Eq. (6) by $\tau - 1$, resulting in an representation of the XDoG filter as an adjusted image sharpening operator:

$$\begin{aligned} S_{\sigma,k,p}(x) &= \frac{D_{\sigma,k,p}(x)}{\tau - 1} = G_\sigma(x) + p \cdot D_{\sigma,k}(x) \\ &= (1 + p) \cdot G_\sigma(x) - p \cdot G_{k\sigma}(x) \end{aligned} \quad (7)$$

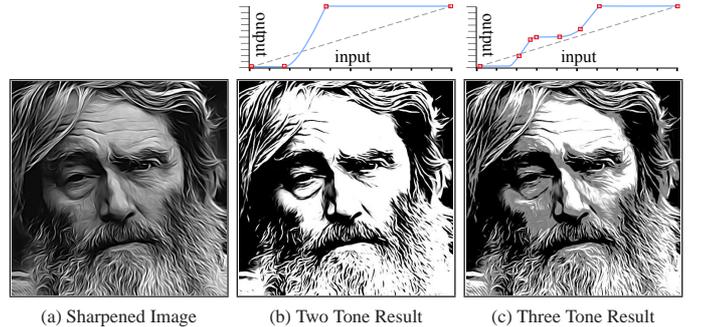


Figure 7: *Different types of luminance adjustment functions*: The two tone result is created using a soft thresholding function. The three tone result is generated by replacing soft thresholding with a smooth curve approximating three value quantization.

Obviously, the range of images that can be generated using $T_{\varepsilon,\varphi}(S_{\sigma,k,p} * I)$ is identical to the range of images that can be generated using the original formulation. However, replacing τ with p makes it possible to control the strength of the edge sharpening effect without influencing any other aspects of the filter. In addition, ε is now measured proportionally to image intensity, leading to an effective decoupling of the parameters.

The sharpened image generated by $S_{\sigma,k,p}$ can be understood as a digital approximation of the classical darkroom technique of *unsharp masking*. To perform an unsharp mask, a photographer uses a negative duplication technique to create a blurred version of the original negative. Using that blurred negative as a mask when creating a print has the effect of sharpening the original edges [19]. The same effect is present in Eq. (7), which can be understood as an unsharp mask of the blurred image $G_\sigma * I$, in which the brightness has been increased in order to compensate for any darkening due to the mask. Figure 6 demonstrates the relationship between the base DoG response, the sharpened image, and the XDoG result.

A wide range of different stylistic effects can be achieved by applying the soft thresholding function $T_{\varepsilon,\varphi}$ to the sharpened image $S_{\sigma,k,p} * I$. Larger or smaller φ control the sharpness of the black/white transitions in the image, while ε controls the level above which the adjusted luminance values will become white. However, $T_{\varepsilon,\varphi}$ is only one of many luminance adjustments that can be applied to the sharpened image. While most of the images in this paper are created using the soft thresh-

olding function, Fig. 7 demonstrates additional effects that can be achieved by replacing $T_{\varepsilon, \varphi}$ with a more general luminance adjustment function.

Figure 23 demonstrates some parameter variations for a single source image. Image (b) sets the Gaussian variance of $\sigma = 0$, thus the DoG result is zero and the filter implements pure tone-mapping. Compare this with images (c) and (d). Sharpening with a large p exaggerates both the black and white edges present in the result (see Sec. 3.4). Images (e) and (f) demonstrate how ε can be adjusted to create different line-art appearances. Together, the XDoG parameters ε, p , and φ enable a range of styles and effects, as evidenced in this paper.

Appendix A lists complete settings for many of our results, demonstrating the range over which we have found it useful to vary the XDoG parameters. We have found that choosing ε close to the midtone greyvalue of the image and p near 20, tends to lead to interesting stylizations; though some specialized styles require much larger p values. The soft thresholding steepness parameter φ varies more widely. Because it controls the slope of the falloff, when φ is close to zero it is very sensitive to small changes, while the parameter becomes much less sensitive to small changes as it increases.

2.6. Flow-based Difference-of-Gaussians (FDoG)

Equations (2) and (7) are isotropic formulations and evaluated identically for each pixel in an image. Consequently, images with stochastic noise or textures may result in an excessive number of small, disconnected edges, as in Fig. 8. Such noisy responses can often be avoided by lowering the edge emphasis strength, p , however, doing so may make it impossible to create outputs containing strong edge lines. A better solution, adopted in recent works, is to adapt the filter according to the approximated edge orientations [8, 9, 20]. The idea behind these approaches is to first respond to changes in luminance that occur across edges and then to smooth those responses using an edge aligned blur. The main difficulty inherent in such an approach is the same issue that Marr and Hildreth faced (Sec. 2.2): it requires a means of approximating the edge orientation at each point in the image, before the edge image itself has been defined.

A simple but effective way to estimate the local orientation is by using the *smoothed structure tensor* (SST), a well-known tool in computer vision, which is given by the smoothed outer products of the image gradients [21]. Performing an eigenanalysis of the SST essentially corresponds to performing a principal component analysis (PCA) of the gradient vectors. The major eigenvector can then be interpreted as smoothed gradient orientation and the minor eigenvector as smoothed tangent orientation. The SST is computationally highly efficient, since smoothing can be performed using a linear filter, such as a Gaussian, and eigenanalysis only involves solving a quadratic equation.

Taken together, the edge orientations calculated at each point form the *edge tangent flow* (ETF). A DoG response can be calculated across edges by evaluating one dimensional Gaussian blurs along lines orthogonal to the edge tangents. Edge-aligned

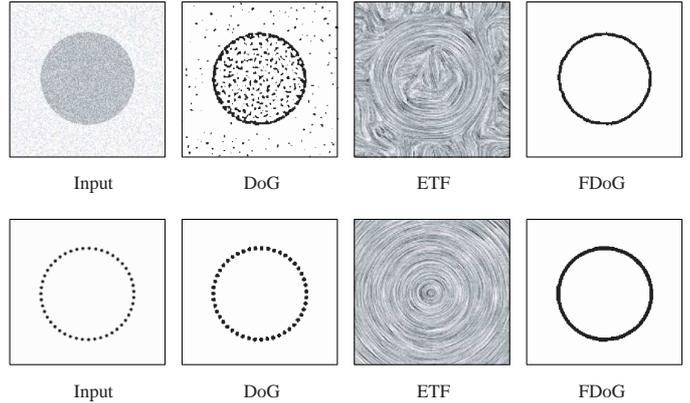


Figure 8: Kang et al.'s [8] FDoG results: (Top Row) The response of an FDoG filter is less susceptible to noise than an isotropic DoG filter. (Bottom Row) The noisy contour of a circle may be fused by the FDoG operator, thereby increasing coherence.

smoothing can then be performed using a line integral convolution pass that follows the edge tangent flow. The two pass flow guided approach is referred to as the *flow-based difference-of-Gaussians*, or FDoG.

Extending the basic DoG to an FDoG, implies replacing the parameter σ with three separate parameters:

- σ_c : Controls the width of the Gaussian used to blur the structure tensor. Small values can increase edge noise, while larger values can distort fine features.
- σ_e : Controls the width of the gradient aligned difference of Gaussians filter. Larger values discard more fine details, and result in wider edge lines.
- σ_m : Controls the width of the edge tangent aligned line integral convolution. Larger values increase the coherence of edge lines by combining several shorter disconnected segments into fewer, longer ones. However, it may also introduce noise into the edge lines, particularly if σ_m is significantly larger than σ_c .

The variable-threshold formulation (XDoG) and flow-alignment (FDoG) are mutually independent extensions to the DoG operator, and may therefore be combined, as desired. For the images in this paper we employ Eq. (5) to produce stylistic variations, while we rely on the FDoG implementation of Kyprianidis et al. [9] for noise suppression and increased coherence. A detailed discussion of implementation details for the FDoG can also be found in [22].

3. Effects

Artists have developed a large corpus of *elements-of-style* to enhance the visual appeal and visual communication effectiveness of their artwork. To distinguish these elements-of-style from the *styles* discussed in Section 4, we shall refer to the former as *effects*.

There exist dedicated systems to produce most of the effects presented in this section. The advantage of those systems is

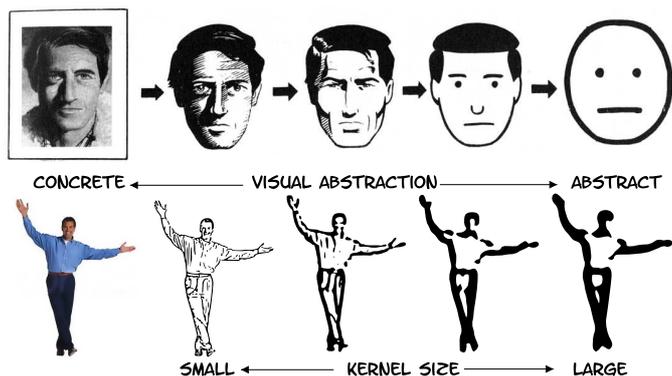


Figure 9: *Abstraction*: (Top) A successive reduction in level-of-detail abstracts the portrait of a man from a concrete instance to a generic representation (©1993 Scott McCloud). (Bottom) A DoG filter does not achieve the stylistic adaptation (including iconization) of a skilled artist, but it does remove concrete features and simplifies the shape with increasing kernel size, σ .

generally a larger flexibility in terms of user parameters. The advantages of the DoG operator are that it is a single, simple operator, which functions fully automatically. Given the visual quality and appeal of the DoG results, it should be considered as a contender for applications where speed or automation are paramount to customization of the effect parameters.

3.1. Level-of-Detail or Abstraction

Since Art is not bound by the laws of physics and optics (i.e. Art may not be *photorealistic*), artists are free to choose which detail in a scene to depict and which to omit. The visual art form of *Comics* or *Cartoons* leverages this principles extensively by employing a minimalistic visual language that focuses on strong shapes, commonly depicted with simplified outlines (e.g. edges). Artists may thus distill the essence of a scene or situation without having to depict its every nuance. The specific form of an instance of a class of objects may be simplified (abstracted) to focus on the common properties of the class (e.g. faces), rather than the accidental properties of the instance (e.g. Harry’s face), as in Figure 9, top row.

Given that small image details are represented by high spatial frequencies, it follows that filtering out such details leads to a type of (shape-) abstraction [4]. Intuitively, the more blurred two pictures of different faces are, the more similar they are likely to look. But while blurry (out-of-focus) images are generally undesirable, edge images may use the same effect while retaining their visual appeal. In Figure 9 bottom row, the image of a man with a specific pose is depicted using a DoG edge detector with varying spatial support, σ . Given a small spatial support, specific details, such as the eyes, shirt, and trouser pocket, are reproduced. With larger spatial support, such details are increasingly omitted and simplified until all that remains is the shape of a humanoid figure.

It should be noted that we do *not* claim that the automatic abstraction afforded by the DoG operator is as effective or sophisticated as the manual abstraction of a skilled artist, which may be a complex mixture of skill, experience, and semantics. The observation we would like to make (with this and the following effects) is rather than that the DoG operator, by the very nature

of its design (or with just minimal modifications), is capable of reproducing effects with similar visual qualities to those found in many artworks.

3.2. Indication

Indication is another, more subtle mechanism for abstraction. Here the aim is not shape-simplification, but rather *summarization* of repetitive image content (most commonly textures) which are *indicated* with a few representative elements, instead of being fully expressed. Humans are very adept at detecting visual patterns [3], and indication appears to be a short-hand used by artists to convey the structural rules of these patterns, leaving it to the observer to ‘fill in the blanks’ based on these rules. Figure 10(a) uses indication to hint at bricks, which, together with the water dispenser, places the protagonist in a school setting. Indication permits the artist to save some drawing effort, but more importantly, it focuses the viewer’s attention on the foreground, it avoids visual clutter, and it assists in visually parsing the scene.

The DoG operator by itself is not capable of indication, because it lacks a mechanism to prioritize edges, i.e. to decide which edges to indicate and which edges to omit. Winnemöller et al. [7] proposed a bilateral pre-processing pass to act as such a prioritization mechanism. A bilateral filter is essentially a blur operator and therefore capable of removing extraneous detail. The amount of local blur is guided by the image content, so that low-contrast regions are blurred more than high-contrast regions. This has the effect of attenuating weak edges, while supporting strong edges, effectively performing a simple indication of mostly homogeneous (photometrically and spatially) textures.

Similar to the disclaimer in Section 3.1 we point out that the indication mechanism described here has significant limitations and does not compare with the skillful indication of a trained artist. For example, the DoG based indication does not deal well with complex (structure at multiple scales) or foreshortened textures [7, 14]. However, it is worth mentioning that *abstraction* and *indication* remain some of the fundamentally unsolved problems in non-photorealistic rendering (NPR), partly because semantics play such an important role. Given this fact, the quality of these effects afforded by a simple edge detector is arguably good, and might be used as the starting point for deeper research into these elemental problems. As an aside, there is strong evidence to suggest that anisotropic diffusion (such as bilateral filtering) forms part of human texture perception. The above generative approach to automatically produce primitive indication is functionally equivalent to the first two stages of Malik and Perona’s analytical perceptual model for preattentive texture discrimination [23].

3.3. Motion

Speed and motion in cartoons are commonly suggested by (1) applying lines in the direction of motion (*speed-lines*), and (2) drawing faded, offset duplicates of an object in its direction of motion (*ghosting* or *streaking*), as in Figure 11.

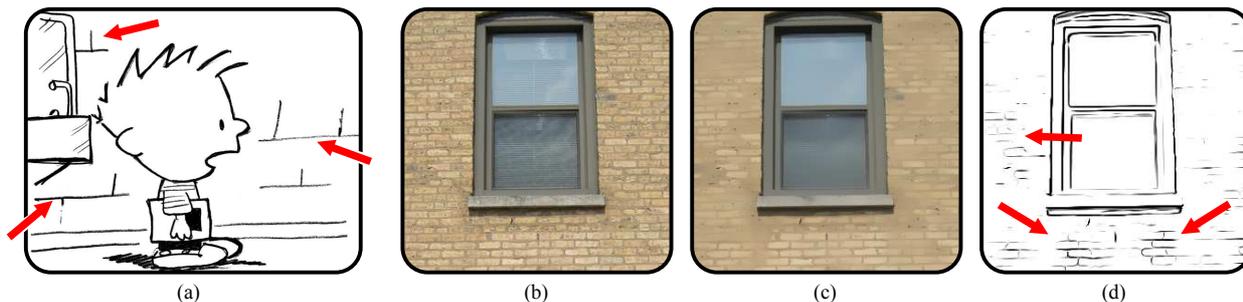


Figure 10: *Indication*: (a) Cartoonists commonly only indicate unimportant background elements (from Calvin&Hobbes, ©Bill Watterson). (b) A brick wall with window. (c) Bilateral filter result of (b) with simplified texture. (d) Automatic DoG indication of (b) by detecting edges on (c).

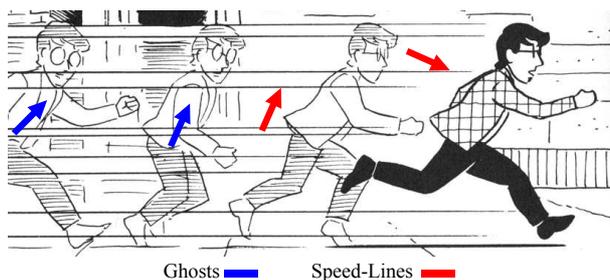


Figure 11: *Stylized Motion*: Speed and motion are represented in cartoons using speed-lines and ghosting (©1993 Scott McCloud).

Speed-lines. Motion blur is a temporal accumulation effect that occurs when a photographed object moves relative to the camera during exposure. This relative movement may be a complex motion, but we shall focus on a simple translation for brevity. A linear motion blur is similar to a standard Gaussian blur (Eq. (1)), except that the blur kernel is not a radially symmetric Gaussian shape, but rather an elongated line in the direction of motion. Convolution with such a kernel blurs detail along that line, but not perpendicular to it. Comparing Figs. 12(a) and (b) illustrates this principle. The car is traveling horizontally, resulting in a horizontal motion blur. A horizontal feature, like the hood of the car, is blended with itself, therefore remaining relatively unperturbed. A vertical feature, like the back of the car, is blended with the background, thereby becoming blurred. The DoG operator thus detects horizontal edges instead of vertical ones, because the latter are smoothed away by the motion blur. The net effect is that of edges appearing as speed-lines in the direction of motion.

Ghosting. Successively faded and offset contours of an object (ghosts) are another stylistic device that artists use to depict motion, as in Fig. 11. Such an effect emerges from a simple DoG operator for certain video inputs. Fig. 12(c, left) shows a single frame of a video sequence where the shutter speed of the camera is higher than the frame-rate of the video, resulting in multiple exposures of the moving object within a single frame (strobing). The edge image created by the DoG operator thus contains ghosts.

Dedicated systems exist for the generation of cartoon-style motion-effects [24, 25, 26], which allow for more control over the effect parameters. However, these systems require either

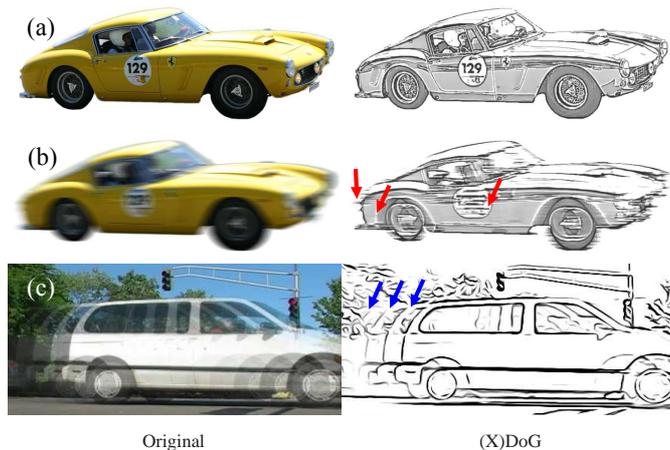


Figure 12: *Speed-lines*: (a) Images of a stationary car, and corresponding DoG image; (b) Moving car with motion-blur, and DoG image with speed-lines; (c) Moving car with strobing, and DoG image with ghosting.

user-input, camera arrays, or other means for robust object and motion segmentation. As a trade-off, the DoG operator offers less control, as it operates on the motion-information implicitly encoded in the input image (blur, strobing), but this frees the operator from requiring explicit motion information (such as dense optical flow) and allows it to work fully automatically.

3.4. Negative Edges

Some artists, such as Frank Miller or Shigeru Mizuki (Fig. 13a), have mastered the depiction of images with just two tones (black and white), lending their artwork a dramatic, stark look. Given such a limited palette these artists use inversion techniques to depict scene detail in dark image regions. Fig. 13(a) shows example of both traditional edges (blue arrows), as well as inverted edges (red arrows), which we call *negative edges*.

In 3D computer graphics, DeCarlo and Rusinkiewicz [27] demonstrated how such inversion techniques can be applied to better illustrate 3D models. Similarly, Lee et al. [28] added bright highlight lines as additional shape cues to 3D models. For images, Rosin and Lai [10] obtained negative edges by computing standard edges on an inverted source image and applying a set of hand-crafted compositing rules.

In contrast, the XDoG operator is capable of producing both black and white edges ‘out-of-the-box’ (Figs. 13 b,c). The rea-

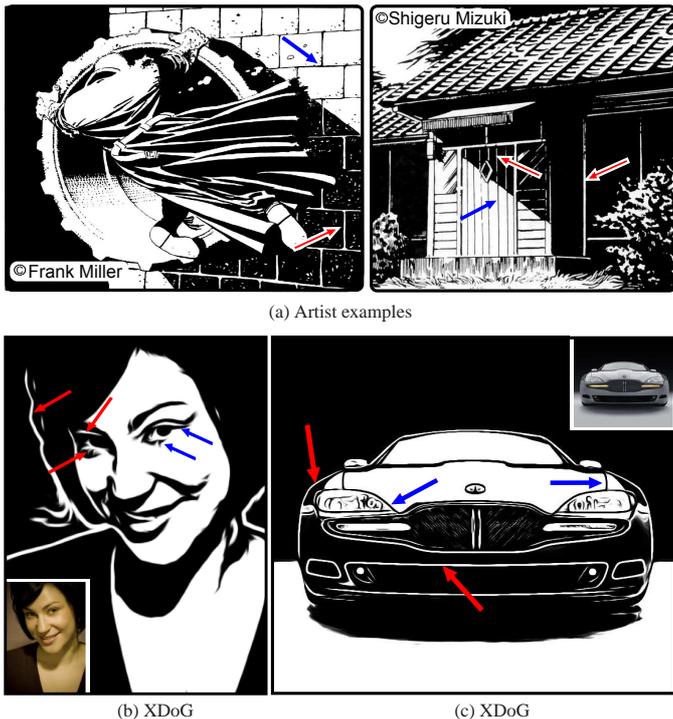


Figure 13: *Negative Edges*: Examples of standard black edges (blue arrows) and negative white edges (red arrows). (a) Negative edges used by artists; (b,c) XDoG: Source images are inset. Compare corresponding features in lit and shadowed regions, such as features around the eye in (b), the outline of the headlights in (c).

son for this becomes evident when re-examining Fig. 6. The DoG response displayed in that figure shows response values close to zero as greys, strong positive responses as white, and strong negative responses as black. Notice that most edges in the image induce a matched pair of edge lines in the DoG response, one positive and one negative. When a strong positive DoG response occurs inside a darker image region, it results in a white edge line. And when a negative DoG response occurs inside a bright region, it leads to a more traditional black edge.

4. Duo-tone Styles

Given the DoG’s relationship to edge detectors with binary output, we first focus our attention on duo-tone (black-and-white) styles, such as *thresholding*, and *woodcut*. Compared to naïve approaches that either rely solely on edge detection or brightness thresholding in isolation, the styles discussed in this section significantly benefit from an XDoG formulation, which combines contrast enhancement and tone-mapping.

4.1. Thresholding

Traditional thresholding may be considered a tone-mapping operator ($[0, 1] \mapsto \{0, 1\}$), which maps values in a continuous range below a certain threshold value to 0, and those values above the threshold to 1. The ostensible simplicity of an image containing only black and white belies the stark visual appeal that skilled artists can achieve in this medium (Fig. 13a).

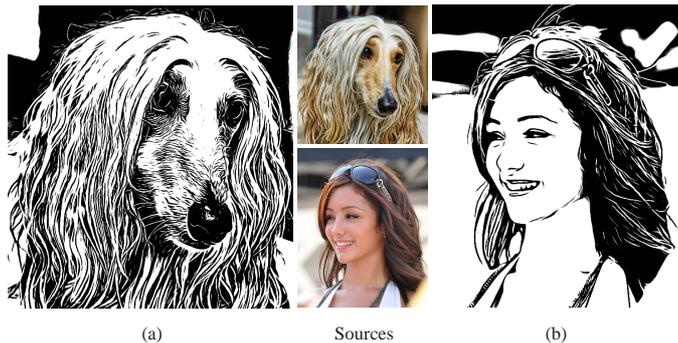


Figure 14: *Thresholding*: (a) High-quality preservation of detail even in relatively low-contrast regions, such as the hair and snout; (b) Fine reproduction of reflection and transparency in glasses

Related Work. Various recent works have re-examined the problem of thresholding. Gooch et al. [6] used DoG edges composited on top of simple luminance thresholding to generate facial abstractions. As such, their solution is a multi-pass approach which is based on simple thresholding and does not preserve detail in dark image regions (negative edges). Mould and Grant [29] proposed a “complex algorithm” (cit.) consisting of four stages: computing image statistics, image segmentation using graph cuts or loopy-belief-propagation (LBP), removal of small regions, and contour smoothing via vectorization. The LBP stage was run iteratively to merge small regions into larger ones. Final images were composited from a base-layer and one or more detail layers. Compared to Mould and Grant’s results, our images are simpler to compute and exhibit additional artistic effects, such as *negative edges*. Xu and Kaplan [30] also employed region segmentation. They formulated the problem of thresholding as an optimization to label the segmentation with *black* and *white* labels to minimize the total of several cost functions. The resulting images lent themselves to shape simplification, but they were computationally expensive and small parameter changes were bound to lead to significant changes in the output image, due to the nature of the optimization. Rosin and Lai [10] produced images that were multi-tone instead of strictly black-and-white. They used a posterized (multi-tone) base layer in addition to a detail edge-image. The edge image was based on Kang et al.’s [8] FDOG edges and is therefore related to the results presented here. Rosin and Lai approximated the effect of negative edges by computing two edge images (one on the normal input for black edges, and one on the inverted input for white edges) and combining these onto a gray background. Note that, though related, this is not fully equivalent to negative edges of the XDoG operator. Finally, the base and detail layers were combined using a hand-crafted table of compositing rules. Compared to Rosin and Lai’s approach, the XDoG operator is less versatile, but may produce similar results and can be obtained with a single XDoG operator, without requiring separate base and several detail layers, or a composition table.

Implementation. The results in Fig. 14 are obtained by simply applying the XDoG operator (Eq. (5)) with appropriate parameter values. Using $p \approx 20$ ensures strong emphasis lines,

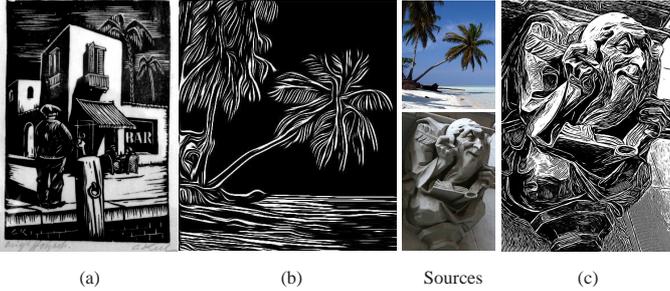


Figure 15: *Woodcut*: (a) Woodcut by Carl Eugen Keel; (b,c) Large flow distortion and negative edges result in a *Woodcut* appearance.

while setting $\varphi \gg 0.01$ effectively changes the *tanh* soft-ramp in Eq. 5 to a step-function. Note the use of negative edges in Fig. 14(a,b). More threshold results are shown in Figs. 13(b,c), and 24(a). Note also the direct comparison with Xu and Kaplan [30], Figs. 20, 21, and Mould and Grant [29], Fig. 22.

4.2. Woodcut

Woodcut (aka *xylography*) is an ancient printing technique, in which a volumetric material with a planar side (traditionally wood) is carved to produce indentations and excavations in the material. Paint is then rolled onto the material and paper pressed against the wood to absorb the paint. This results in paint being applied in all places except those carved out by the artist (akin to a stamp). As such, the technique is often associated with strong black-and-white contrast, as well as distinct carving lines, as in Fig. 15a.

Related Work. While this technique is related to other engraving techniques, such as digital facial engraving [31], not much prior work exists on the woodcut technique itself. Mizuno et al. mainly focused on systems for virtual carving and printing of woodcuts [32]. Mello et al. proposed a procedural system, based on image segmentation, edge detection, and virtual carving-cuts along flow-lines, to produce simple, but directable woodcuts [33]. Like Mello et al.’s system, the XDoG operator is limited in the stylistic range of woodcut images it can produce. However, the XDoG images are easier to compute and, arguably, of higher visual quality.

Implementation. Figs. 15(b,c) were generated with the above-mentioned *threshold* settings (Sec. 4.1), but use very aggressive flow-blurring and extreme edge emphasis settings to produce shape abstraction, and long, coherent carving-cuts ($\sigma_c \approx 5$ and $p \approx 100$).

4.3. Anti-aliasing

To produce high-quality output images, we prefer the final images to be slightly anti-aliased. This is particularly important for thresholded images, whose response function is close to a step-function. Recently, Yang et al. [34] suggested a method for restoring antialiased edges that suffered degradation from applying certain types of non-linear filters, such as those used in many NPR algorithms. While their approach may be efficiently implemented on a GPU, we propose an even simpler solution



Figure 16: *Anti-Aliasing (AA)*: (a) top: no AA, middle: soft AA, bottom: extreme AA (stylistic)

for the case of the XDoG operator. Since many of the examples in this paper use the ETF field to compute coherent edges, we can easily re-use the ETF to apply a very small line integral convolution along the field, thereby producing image-coherent and visually pleasing anti-aliasing. In doing so, the integration radius is commonly only 0.5-2 pixels (Fig. 16a, middle). However, larger integration values may be used for stylistic effect (Fig. 16a, bottom).

5. eXpanded Styles

While Winnemöller et al.’s [7] main motivation for an extended DoG formulation was to increase temporal coherence for video abstraction, Winnemöller [35] later noted the potential of this extension for stylistic purposes. First, we specify how several threshold results may be combined with natural media textures to produce a convincing *hatching* look. Furthermore, when the parameters values are dialed away from the duo-tone settings of Section 4, we may emphasize the tone-mapping and image sharpening characteristics of the XDoG to obtain a variety of *natural media styles*.

5.1. Hatching

Our hatching approach (Fig. 17) is based on the concept of tonal art maps, where layers of strokes add up to achieve a desired tone [36, 37]. First, we compute a standard DoG edge image (Fig. 17(a), top). We then create two high contrast XDoG images by setting $\varphi \gg 0.01$. To obtain multiple threshold results, we merely adjust ε . For efficiency, we may choose to compute all threshold results in a single pass and write them out to different channels of the same image, thereby incurring only a negligible overhead.

Hatching textures are generated by tiling small patches of scanned hatches, as in Fig. 17(b). For hatching with global directions, the textures may be pre-computed using texture synthesis [38, 39]. For hatching with local directions, local regions are defined by segmenting the (colored) source image (e.g. using graph-cuts) and then masking regions that overlap with the threshold results. Each region is then tiled independently, as above, with a local hatching direction.

The results in Fig. 17(a) may be used as masks for the hatching textures in Fig. 17(b), and multiplied together to compute the final image. For added effect, the hatching output can be

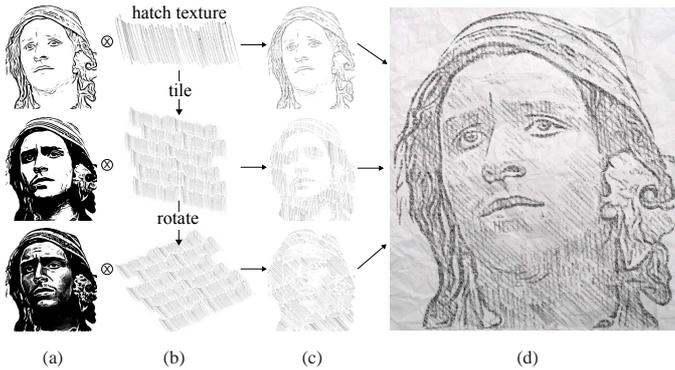


Figure 17: *Hatching*: (a) Threshold variations; (b) Tiled and rotated hatching textures; (c) Textures in (b) masked by (a); (d) Final result by multiplying all (c)’s onto a paper texture.

composed onto a paper texture, as in Fig. 17(d). While simple to implement and efficient to compute, the hatching style retains many of the XDoG features (clean edges, tone control, negative edges) and produces high-quality results. Since the threshold results are merely used as masks, the visual resolution of the final image mostly depends on the hatching textures, which may be generated at any desired quality. Additionally, different hatching markers may be simulated (pencil, ink, felt-tip) by simply scanning and using these markers in the hatching texture generation, as in Fig. 24(b).

5.2. Natural Media

The various XDoG parameters may be adjusted to allow for a large range of natural-media-style appearances. The *pencil-shading* look of Fig. 1(b) relies on high-frequency detail resembling graphite on paper. The high details are obtained with $\sigma \approx 0.4$, and we ensure a proper tone-response with $\varphi \approx 0.01$. The *charcoal* appearance of Fig. 18(c) is due to a much larger spatial support ($\sigma_e \approx 7$), creating broad strokes. The *pastel* style in Figs. 18(b) and 24(c,d) can be achieved with an intermediate $\sigma_e \approx 2$. The latter two styles employ the flow-based smoothing of Kyrianiadis et al. [9], with an identical flow-field computation. Here, the structure tensor is only minimally smoothed ($\sigma_c \approx 0.1$) and the FDoG integration uses a relatively large kernel along the flow-direction ($\sigma_m \approx 20$), resulting in noticeable turbulence and noise along the image edges, which appears as dry brush or charcoal on canvas, depending on the edge width, edge emphasis, and anti-aliasing settings. We achieve the colored pastel look in Fig. 24(d) by modulating the natural media appearance of Fig. 24(c) with source image colors, which are weighted by inverting Fig. 24(c).

6. Discussion

The DoG operator has been employed in a variety of applications, ranging from computer vision [4] to stylistic rendering [6, 7, 8]. However, its use has always been limited to edge detection, or the creation of straight-forward edge images. To achieve more complex styles and effects, researchers have resorted to building complex, multi-stage systems, often including expensive optimization schemes [29, 30, 10]. In this paper,



Figure 18: *Variations*: Parameter variations of the XDoG operator produce a range of natural media looks. (a) Source; (b) Pastel; (c) Charcoal. (Source © Flickr user tibchris)

we have demonstrated that many of these styles and effects can be achieved more directly and computationally efficiently with slight variations of an extended DoG formulation.

6.1. Limitations

Like any image processing filter, the XDoG operator has its limitations. For some parameter values in the thresholding range (Sec. 4.1) noise may be severely amplified. This becomes particularly noticeable in otherwise homogeneous regions in the source image (Fig. 19a). While the source image looks “clean” at first glance, any image may contain shot noise or other noise artifacts that are highlighted by the DoG response. Fig. 19(b) was additionally tweaked to demonstrate another issue: The ETF computation on the source image picks up JPEG blocking artifacts. As a result, the spurious noise is integrated around JPEG block boundaries to produce the flowy, tiled structure apparent in Fig. 19(b). However, a slight shift in parameter

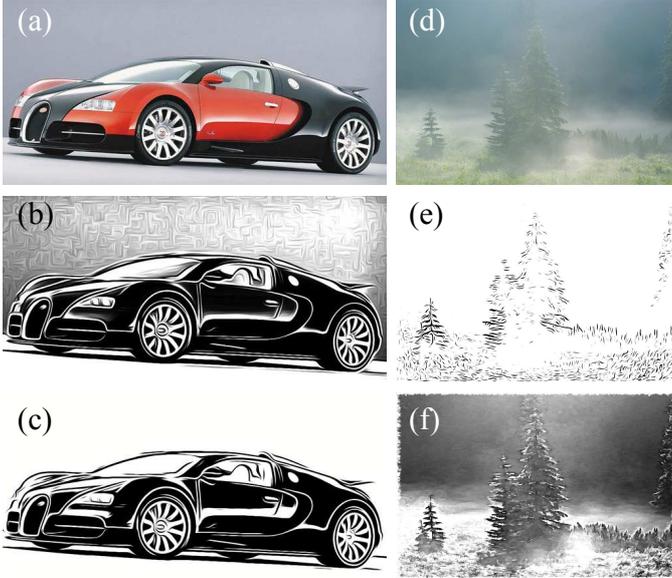


Figure 19: *Limitations*: (a) Source image (©Flickr:InertiaCreeps); (b) Noise artifacts; (c) Slightly modified settings compared to (b); (d) Low contrast source image (©Adam Baker); (e) Edge detection result; (f) Pastel result.

values, increasing φ and lowering the white-value ε , eradicates the noise emphasis to produce a clean Fig. 19(c).

Another challenge for any edge detector are low-contrast images, such as Fig. 19(d). Computing the flow-DoG of Fig. 19(d) results in the visually disappointing Fig. 19(e). However, using a flow-XDoG with *pastel/dry-brush* settings (Sec. 5.2) we obtain an arguably much more pleasing result (Fig. 19f). There are two important points to note here. First, Fig. 19(f) looks good due to the emphasized fog, which accentuates the interplay of light with the fog. The experienced photographer, painter, or NPR practitioner will immediately recognize that contrast-enhancement is an age-old trick-of-the-trade to increase the dynamic appeal of an image. As it turns out, the XDoG operator contains two fundamental mechanisms (unsharp masking and tone mapping) that are particularly well suited to achieve such desirable contrast emphasis. Second, it is fair to say that not all the styles discussed in this paper will produce adequate results for all input images (e.g. Figs. 19d,e). However, given the large range of styles that may be produced with the few parameters of the XDoG, and the noise-suppression and coherence-increasing characteristics of the flow formulation, we have found that an interesting or visually pleasing result can be obtained for most input images.

6.2. Future Work & Conclusion

Looking forward, we are interested in investigating the role that operators like the XDoG may play in the inception and comprehension of *effects* (elements-of-style). Line drawings are effective because humans use edge-detection to decompose their visual world [2]. Evidence suggests that speed-lines and ghosting are not merely artistic fancy, but may be traced back to the physiology of human vision [40]. Similarly, it is possible that *negative edges* are not just stylistic necessity to depict detail in dark image regions, but may be tightly linked

to early vision processes [18]. Such connections between human vision and the effectiveness of artistic styles have been proposed by other authors. Zeki, a distinguished neurologist, states “[Artists] are exploiting the characteristics of the parallel processing-perceptual systems of the brain to create their works, sometimes even restricting themselves largely or wholly to one system, as in kinetic art” ([41], pg. 80). Specifically, Zeki and Lamb found that various types of kinetic art are ideal stimuli for the motion sensitive cells in area V5 of the visual cortex [42]. In another experiment, Zeki and Marini [43] showed that fauvist paintings, which often divorce shapes from their naturally assumed colors, excite quite distinct neurological pathways from representational art where objects appear in normal color. Ramachandran and Hirstein explore similar concepts [44].

Given the significant stylistic arsenal at the disposal of the NPR community, we believe that it will be enlightening to re-examine various artistic styles from a perceptual point-of-view. Instead of focusing purely on art-direction and expressiveness, we can attempt to devise minimal filters and operators capable of qualitatively simulating the desired style, and then conduct experiments to investigate how their parameter space influences the effectiveness of visual communication tasks. In limited form, such approaches have been undertaken for Cartoons and Caricatures [45, 6, 7], but we believe that to address some of NPR’s open problems, such investigations should be expanded to include styles focusing on color, motion (temporal coherence), indication, and abstraction.

We hope that our theoretical review of the DoG operator, its recent extensions, and our reparameterization for stylization purposes help to broaden the community’s understanding and interest in the operator’s potential beyond mere edge-detection.

Appendix A. Parameter Settings

The majority of the results shown in the paper are generated using the flow-based XDoG filter. That filter is created by combining the DoG extensions of Sections 2.5 and 2.6, resulting in a single filter with six parameters. While a basic DoG filter has only a single parameter, σ , extending the basic DoG to an FDoG, as discussed in Section 2.6 implies replacing σ with three separate parameters, σ_c , σ_e , and σ_m . Extending the DoG to create the XDoG introduces an additional three parameters, p , φ and ε , which are discussed in Section 2.5. Finally, as discussed in Section 4.3, many of our results have been post-processed using an additional line integral convolution pass, in such cases, the width of the post-processing blur becomes a seventh parameter, σ_a .

Examples of the exact parameters used to generate many of the results in the figures are given in Table A.1.

In addition to the seven formal parameters of the filter, there are also several implementation details that can significantly effect the output. The first is the choice of colorspace. Our implementation is based on CIE Lab. Input RGB images are assumed to be linearized, so no gamma correction is done prior to colorspace conversion. All three Lab channels are used when calculating the structure tensor terms, while the luminance channel

Table A.1: Parameters used in the examples

Result	Fig.	σ_c	σ_e	σ_m	p	φ	ϵ	σ_a
	2(g)	2.28	1.4	4.4	21.7	0.017	79.5	1.0
	6	2.45	1.0	6.0	18.0	0.60	82.2	NA
	13(b)	2.97	1.4	13.2	18.2	10.3	73.1	1.95
	14(a)	3.76	1.4	2.20	15.7	0.49	78.3	2.4
	15(a)	5.84	0.8	3.2	120	.083	72.6	.75
	18(b)	0.10	2.0	20	40	0.01	100	7.2
	18(c)	0.10	6.8	20	70	0.01	80.0	0.6
	19(b)	4.16	1.4	12	22	0.09	88.0	4.0
	19(c)	4.16	1.4	12	22	3.42	79.0	4.0

is used to define the grayscale source image. The contrast adjustments implied by the Lab conversion have a significant impact on shape and structure of the edges in the results. Also, because the range of luminance values is $[0, 100]$, ϵ values should be expected to fall in a similar range

Finally, the implementation of Gaussian convolution also has an impact on the output. When approximating flow-aligned convolutions (needed for the gradient and edge tangent aligned filters), we sample the Gaussian's response at all integer locations less than $2\times$ the corresponding standard deviation. However, when performing the two component passes of the structure tensor blur, we extend the sampling to all pixels within $2.45\sigma_c$.

Appendix B. Efficient Implementation

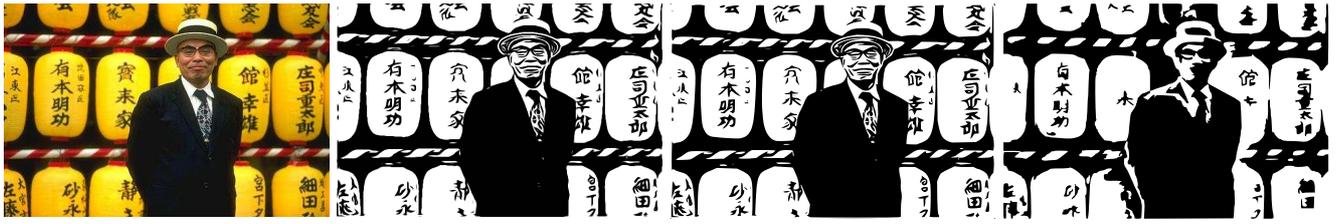
Note that the Gaussian operator, G , in Eq. (1) is linearly separable and therefore efficiently implemented as two successive one-dimensional Gaussian operators (one applied horizontally, one vertically). The same separability, however, does not hold for the DoG operator, $D_{\sigma,k}$, Eq. (2), or its extensions. One can still benefit from the separability of the Gaussians, at the expense of additional storage, as follows. Given an input image, I , compute I_{G1_x} and I_{G2_x} as the responses of a one-dimensional Gaussian with standard deviations σ_1 and σ_2 , respectively. Then, compute I_{G1_y} and I_{G2_y} from I_{G1_x} and I_{G2_x} by applying the relevant one-dimensional Gaussian operators in the perpendicular directions, before finally computing $D_G = I_{G1_y} - I_{G2_y}$.

For GPU implementations, where texture access is often more expensive than a small number of computations, the cost of texture lookups for the first pass can be ameliorated by accessing image values $I(x)$ only once for each coordinate and computing I_{G1_x} and I_{G2_x} simultaneously. If the number of channels in the input image is small enough (typically < 3), the result of the first pass can be written into multiple channels of an output image, and the same approach described above may be applied for the second pass, effectively enabling the computation of the entire DoG operator in just two 1-D convolution passes.

References

- [1] Koenderink, J.J.. What does the occluding contour tell us about solid shape? *Perception* 1984;13:321–330.
- [2] Marr, D., Hildreth, E.C.. Theory of edge detection. *Proc Royal Soc London, Bio Sci* 1980;207:187–217.
- [3] Palmer, S.E.. *Vision Science: Photons to Phenomenology*. The MIT Press; 1999.
- [4] Witkin, A.P.. Scale-space filtering. In: 8th Int. Joint Conference on Artificial Intelligence. Karlsruhe, Germany; 1983, p. 1019–1022.
- [5] Canny, J.F.. A computational approach to edge detection. *IEEE Trans on Pattern Analysis and Machine Intelligence* 1986;8:769–798.
- [6] Gooch, B., Reinhard, E., Gooch, A.. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans Graph* 2004;23(1):27–44.
- [7] Winnemöller, H., Olsen, S.C., Gooch, B.. Real-time video abstraction. *ACM Trans Graph* 2006;25(3):1221–1226.
- [8] Kang, H., Lee, S., Chui, C.K.. Coherent line drawing. In: *NPAR '07*. ACM. ISBN 978-1-59593-624-0; 2007, p. 43–50.
- [9] Kyprianidis, J.E., Döllner, J.. Image abstraction by structure adaptive filtering. In: *Proc. EG UK Theory and Practice of Computer Graphics*. 2008, p. 51–58.
- [10] Rosin, P.L., Lai, Y.K.. Towards artistic minimal rendering. In: *NPAR '10*. ACM. ISBN 978-1-4503-0125-1; 2010, p. 119–127.
- [11] Winnemöller, H.. XDoG: Advanced Image Stylization with eXtended Difference-of-Gaussians. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*. *NPAR '11*; New York, NY, USA: ACM. ISBN 978-1-4503-0907-3; 2011, p. 147–156.
- [12] Pratt, W.K.. *Digital Image Processing*. John Wiley & Sons, Inc.; 3 ed.; 2001.
- [13] Orzan, A., Bousseau, A., Barla, P., Thollot, J.. Structure-preserving manipulation of photographs. In: *NPAR '07*. ACM; 2007, p. 103–110.
- [14] DeCarlo, D., Santella, A.. Stylization and abstraction of photographs. *ACM Trans Graph* 2002;21(3):769–776.
- [15] Lindeberg, T.. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of Applied Statistics* 1994;21(1):225–270.
- [16] Pallás-Areny, R., Webster, J.G.. *Analog Signal Processing*. Wiley-IEEE; 1999.
- [17] Rodieck, R.W.. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research* 1965;5(11):583–601.
- [18] Young, R.A.. The gaussian derivative model for spatial vision: I. retinal mechanisms. *Spatial Vision* 1987;2:273–293.
- [19] Langford, M.J.. *Advanced Photography: A Grammar of Techniques*. London: Focal Press; 1974.
- [20] Kang, H., Lee, S., Chui, C.. Flow-Based Image Abstraction. *IEEE Trans on Visualization and Computer Graphics* 2009;15(1):62–76.
- [21] Brox, T., van den Boomgaard, R., Lauze, F., van de Weijer, J., Weickert, J., Mrázek, P., et al. Adaptive Structure Tensors and their Applications. In: *Visualization and Processing of Tensor Fields*. Springer; 2006, p. 17–47.
- [22] Kyprianidis, J.E., Döllner, J.. Real-Time Image Abstraction by Directed Filtering. In: Engel, W., editor. *ShaderX7 - Advanced Rendering Techniques*. Charles River Media; 2009, p. 285–302.

- [23] Malik, J., Perona, P. Preattentive Texture Discrimination with Early Vision Mechanisms. *Journal of the Optical Society of America A* 1990;7(5):923–932.
- [24] Collomosse, J., Rowntree, D., Hall, P. Rendering cartoon-style motion cues in post-production video. *Graphical Models* 2005;67(6):549 – 564.
- [25] Kim, B., Essa, I. Video-based nonphotorealistic and expressive illustration of motion. In: *Proc. of Computer Graphics International (CGI 05)*. 2005, p. 32–35.
- [26] Linz, C., Lipski, C., Rogge, L., Theobalt, C., Magnor, M.. Space-time visual effects as a post-production process. In: *Proceedings of the 1st international workshop on 3D video processing. 3DVP '10*; ACM. ISBN 978-1-4503-0159-6; 2010, p. 1–6.
- [27] DeCarlo, D., Rusinkiewicz, S.. Highlight lines for conveying shape. In: *NPAR '07*. ACM. ISBN 978-1-59593-624-0; 2007, p. 63–70.
- [28] Yunjin, L., Markosian, L., Lee, S., Hughes, J.F.. Line drawings via abstracted shading. In: *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*. ACM; 2007, p. 18.
- [29] Mould, D., Grant, K.. Stylized black and white images from photographs. In: *NPAR '08*. ACM. ISBN 978-1-60558-150-7; 2008, p. 49–58.
- [30] Xu, J., Kaplan, C.S.. Artistic thresholding. In: *NPAR '08*. ACM. ISBN 978-1-60558-150-7; 2008, p. 39–47.
- [31] Ostromoukhov, V.. Digital facial engraving. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques. SIGGRAPH '99*; New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-48560-5; 1999, p. 417–424.
- [32] Mizuno, S., Okada, M., Toriwaki, J.. An interactive designing system with virtual sculpting and virtual woodcut printing. *Computer Graphics Forum* 1999;18(3):183–194.
- [33] Mello, V.B., Jung, C.R., Walter, M.. Virtual woodcuts from images. In: *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia. GRAPHITE '07*; New York, NY, USA: ACM. ISBN 978-1-59593-912-8; 2007, p. 103–109.
- [34] Yang, L., Sander, P.V., Lawrence, J., Hoppe, H.. Antialiasing recovery. *ACM Trans Graph* 2011;30:22:1–22:9.
- [35] Winnemöller, H.. *Perceptually-motivated Non-Photorealistic Graphics*. Ph.D. thesis; Northwestern University; 2006.
- [36] Winkenbach, G., Salesin, D.H.. Computer-generated pen-and-ink illustration. In: *Proc. of ACM SIGGRAPH 94*. ISBN 0-89791-667-0; 1994, p. 91–100.
- [37] Praun, E., Hoppe, H., Webb, M., Finkelstein, A.. Real-time hatching. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques. SIGGRAPH '01*; ACM. ISBN 1-58113-374-X; 2001, p. 581–586.
- [38] Praun, E., Finkelstein, A., Hoppe, H.. Lapped textures. In: *Proc. of the 27th annual conference on Computer graphics and interactive techniques. SIGGRAPH '00*. ISBN 1-58113-208-5; 2000, p. 465–470.
- [39] Cohen, M.F., Shade, J., Hiller, S., Deussen, O.. Wang tiles for image and texture generation. *ACM Trans Graph* 2003;22:287–294.
- [40] Kim, H., Francis, G.. A computational and perceptual account of motion lines. *Perception* 1998;27:785–797.
- [41] Zeki, S.. Art and the brain. *Journal of Consciousness Studies* 1999;6(6–7):76–96.
- [42] Zeki, S., Lamb, M.. The neurology of kinetic art. *Brain* 1994;117:607–636.
- [43] Zeki, S., Marini, M.. Three cortical stages of colour processing in the human brain. *Brain* 1998;121:1669–1685.
- [44] Ramachandran, V.S., Hirstein, W.. The science of art. *Journal of Consciousness Studies* 1999;6(6–7):15–51.
- [45] Ryan, T.A., Schwartz, C.B.. Speed of perception as a function of mode of representation. *American Journal of Psychology* 1956;69(1):60–69.



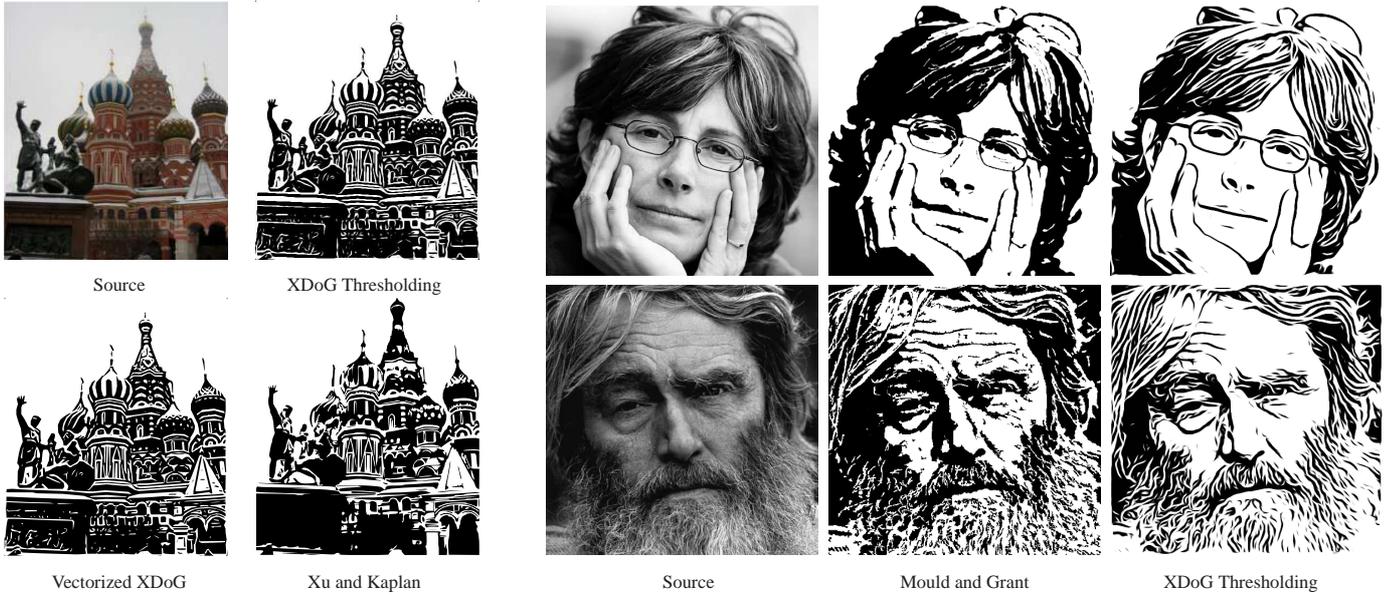
Source

XDoG Thresholding

Vectorized XDoG

Xu and Kaplan

Figure 20: Comparison with *Artistic Thresholding* (Xu and Kaplan [30])



Source

XDoG Thresholding

Vectorized XDoG

Xu and Kaplan

Source

Mould and Grant

XDoG Thresholding

Figure 21: Comparison with *Xu and Kaplan* [30]

Figure 22: Comparison with *Mould and Grant* [29]



(a) Source

(b) $\sigma = 0$

(c) $\sigma = 0.6$

(d) $\sigma = 1.6$

(e) $\epsilon = 0.07$

(f) $\epsilon = 0.26$

Figure 23: *Parameters*: (b) $\sigma = 0 \rightarrow$ pure tone-mapping; (c,d) $\sigma > 0$ increases local contrast; (e,f) variations in ϵ



(a) High-detail thresholding

(b) Felt-tip hatching

(c) Pastel

(d) Colored pastel

Figure 24: Various Additional Results. Please zoom in to see fine details.