

XDoG: Advanced Image Stylization with eXtended Difference-of-Gaussians

Holger Winnemöller*
Adobe Systems, Inc.

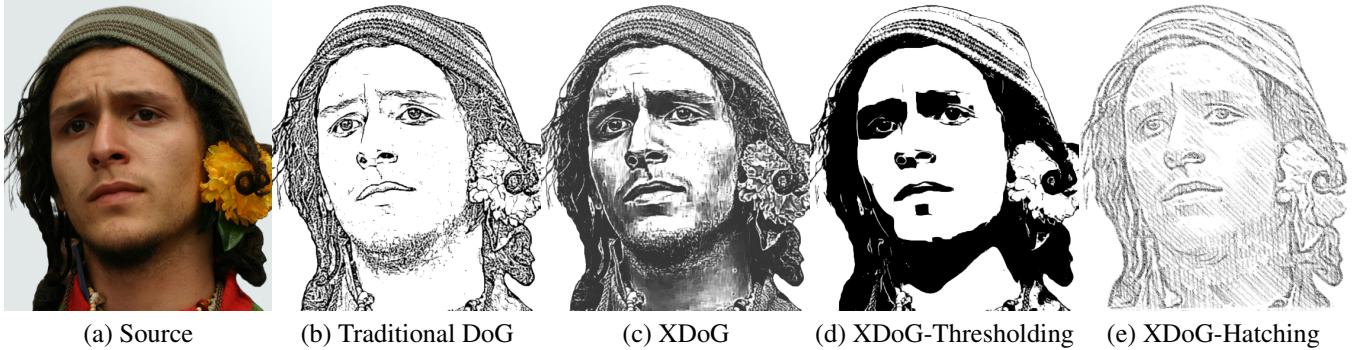


Figure 1: Style variations: All of these images are produced based on slight modifications to a Difference-of-Gaussians operator.
Source © Andrew Calder

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 for stylistic depiction applications. Despite these advances, the technical subtleties and stylistic potential of the DoG operator are often overlooked. This paper reviews the DoG operator, including recent improvements, and offers many new results spanning a variety of styles, including pencil-shading, pastel, hatching, and binary black-and-white images. Additionally, we demonstrate a range of subtle artistic effects, such as ghosting, speed-lines, negative edges, indication, and abstraction, and we explain how all of these are obtained without, or only with slight modifications to an extended DoG formulation. 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.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.3 [Computer Graphics]: Line and Curve Generation

Keywords: Cartoon, Difference-of-Gaussians, Edge Detector, Elements-of-style, Hatching, Negative Edges, Threshold, Sketch.

1 Introduction

The Difference-of-Gaussian (DoG) operator can be used to emphasize or detect edges, when applied to an image. Traditionally,

*e-mail: hwinne@adobe.com

many of the fundamental results in edge detection approaches and operators have been contributed by the computer vision community [Witkin 1983; Canny 1986]. In an effort to decompose and analyze images, edges play an important role in both human and computer vision [Marr and Hildreth 1980; Palmer 1999], and a variety of edge detectors have been proposed for various purposes.

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 [1986] edge detector is arguably the most popular such operator, due to its widespread use in computer vision and image segmentation applications. However, its visual characteristics (lines whose thickness are independent of edge scale) do not lend themselves easily to achieve stylization or abstraction without significant post-processing (Fig. 2c).

The DoG operator, while not as precise at localizing edges throughout scale-space as the Canny operator [Perona and Malik 1991], has been shown to achieve more aesthetically pleasing edges and lines without post-processing, particularly for the synthesis of line drawings and cartoons [Gooch et al. 2004; Winnemöller et al. 2006; Kang et al. 2007; Kyprianidis and Döllner 2008]. In all of these works, the basic operation of the DoG filter is still one of a simple edge detector. Some previous works in thresholding [Gooch et al. 2004; Rosin and Lai 2010] even employ the DoG operator as part of a complex processing pipeline, when an equivalent effect can be achieved with a simple extension to the DoG operator itself (Sec. 4.2). 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 parameter values beyond their traditional edge-detection ranges (Sec.2).

Copyright © 2011 by the Association for Computing Machinery, Inc.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.

NPAR 2011, Vancouver, British Columbia, Canada, August 5 – 7, 2011.
© 2011 ACM 978-1-4503-0907-3/11/0008 \$10.00

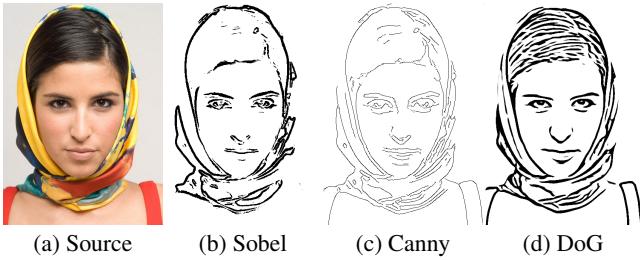


Figure 2: Popular Edge Detectors. *Source ©Maryse Casol*

Effects We demonstrate a number of subtle elements-of-style (effects) with slight or no modifications to an 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 were previously only known from 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 binary black-and-white images. The quality of most of our results are comparable to or better than those of the dedicated systems, yet are easier and faster to compute.

2 Background

It is well established that *edges* (significant intensity changes within a spatial neighborhood) play an important role in human and computer vision [Koenderink 1984; Marr and Hildreth 1980; Palmer 1999]. As a result, a variety of edge detection operators have been proposed, some of which are illustrated in Figure 2. The Sobel operator (Fig. 2b) computes the gradient magnitude at a pixel by convolving an image with a small kernel. A small kernel size allows for efficient computation, but prevents reliable detection of edges at large scales without further processing (Fig. 2b pre-blurs the source to detect larger scale edges). The Canny [1986] operator (Fig. 2c) is still one of the most sophisticated edge detection operators to date. It filters out frequencies beyond a user-specified threshold, then computes the gradient magnitude of the resulting image (for example, using a Sobel operator), and finally traces edges through the image using hysteresis thresholding. The resulting edges are well-suited for computer vision applications as they remain reasonably stable throughout scale-space and approximate the exact centers of edges with single-pixel-wide contours. However, for stylistic applications Canny edges are often unsuitable without further processing, since edges representing traces or outlines are commonly expected to exhibit a certain amount of width and width-variability. In comparison, the Difference-of-Gaussian operator (Fig. 2d) offers a good compromise between computational efficiency and stylistic versatility, as demonstrated in this paper.

2.1 The DoG operator

History/Motivation Rodieck [1965] was among the first to quantitatively examine the neurophysiology of vision, including detection of features, such as edges. Two decades later, Marr and Hildreth [1980] investigated the problem from a computational point of view. They proposed an edge detection mechanism based on the Laplacian-of-Gaussians, which can be approximated for efficiency as the *Difference-of-Gaussians* (DoG), shown in Figure 3.

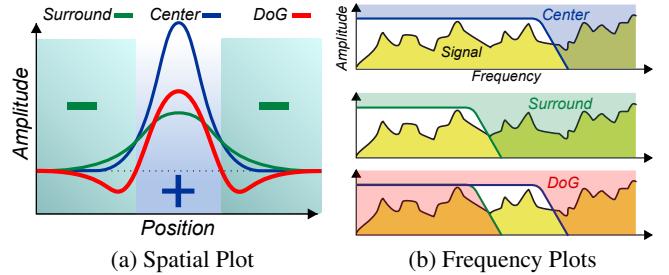


Figure 3: DoG Composition: (a) A wider surround Gaussian is subtracted from a narrower center Gaussian to produce the DoG trace; (b) Two low-pass filters of different cut-off frequency combine to produce a band-pass filter.

Physical Interpretation From a signal-processing viewpoint, a Gaussian filter is a *low-pass* filter, i.e. it allows low spatial frequencies to pass, while attenuating or eliminating high spatial frequencies. The subtraction of two Gaussians thus creates a *band-pass* filter which attenuates all frequencies save those between the cut-off frequencies of the two Gaussians (Fig. 3b). A DoG filter can therefore detect edges whose spatial extent falls within this characteristic frequency band [Pallás-Areny and Webster 1999].

Biological Relevance Young [1987] found that certain retinal cells behaved exactly analogous to the *center-surround* activation mechanism proposed by Marr and Hildreth [1980], as illustrated in Fig. 3(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 [Palmer 1999].

Definition Equation (1) describes a two-dimensional Gaussian function, centered at the origin, where \hat{x} is a two-valued coordinate, and σ is the standard deviation of the distribution. Given this definition, G in Eq. (2), describes the result of convolving an image, I , with the Gaussian function, g . For notational simplicity, the convolution is expressed as the integral over some kernel neighborhood of \hat{x} . It then follows that the classical DoG response, D_0 , of an image, I , can be written as in Equation (3), where k is a factor relating the radii of the standard deviations of the two Gaussian functions. In order for the DoG to approximate the Laplacian of Gaussian, $k = 1.6$ is chosen.

$$g(\hat{x}, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{\|\hat{x}\|^2}{2\sigma^2}} \quad (1)$$

$$G(\hat{x}, \sigma, I) = \frac{1}{2\pi\sigma^2} \int I(x) e^{-\frac{\|\hat{x}-x\|^2}{2\sigma^2}} dx \quad (2)$$

$$D_0(\hat{x}, \sigma, k, I) = G(\hat{x}, \sigma, I) - G(\hat{x}, k \cdot \sigma, I) \quad (3)$$

For readability, the common parameters, \hat{x} (co-ordinate), and I (source image) shall be implied and henceforth omitted.

2.2 Extensions to the standard DoG operator

Variable Thresholding Note, that Eq. (3) does not strictly define an edge detector as it is missing a decision criterion to distinguish

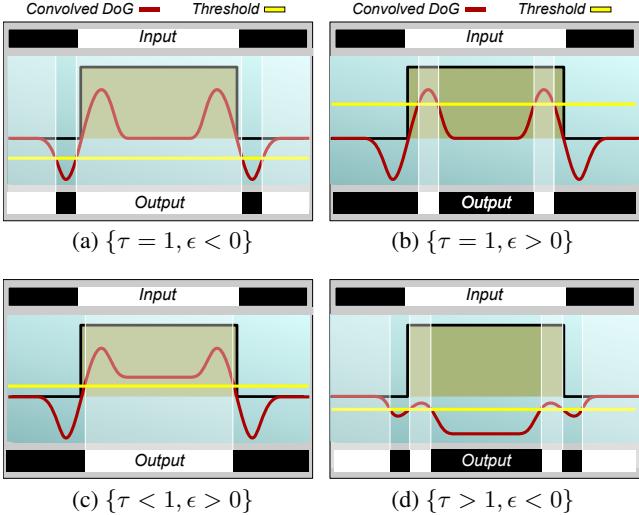


Figure 4: XDoG Parameters: Each subfigure shows (top) an input signal, (middle) a graph of a DoG (red) convolved with the input signal and a threshold level (yellow), and (bottom) the filter response output. (a) Unbiased, low threshold: typical edge detection. (b) Unbiased, high threshold: inverted edge detection. (c) Center-bias, high threshold: Can be used to effect luminance scaling. (d) Surround-bias, low threshold: dual edge detection (positive&negative).

edge values from non-edge values. Commonly, a simple threshold is used:

$$E(\sigma, k) = \begin{cases} 1, & \text{if } D_0(\sigma, k) > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

That is, edges are detected only where Eq. (3) exhibits values smaller than zero. While this approach provides reasonable edge detection, the resulting binary image is rather limited in terms of stylistic range and aesthetic appeal (compare Figs. 1b and 1c).

To address this limitation, Winnemöller et al. [2006] proposed an extension (XDoG) to the standard formulation of Eq. (4), which may be written as:

$$\begin{aligned} D_X(\sigma, k, \tau) &= G(\sigma) - \tau \cdot G(k \cdot \sigma) \\ E_X(\sigma, k, \tau, \epsilon, \varphi) &= \begin{cases} 1, & \text{if } D_X(\sigma, k, \tau) < \epsilon \\ 1 + \tanh(\varphi \cdot (D_X(\sigma, k, \tau))), & \text{otherwise.} \end{cases} \end{aligned} \quad (5) \quad (6)$$

The additional parameters have the following functions: ϵ shifts the detection threshold, thereby controlling *sensitivity* (albeit on an inverted scale: Smaller values make the edge detection more sensitive, while large values decrease detection sensitivity). The τ parameter changes the relative weighting between the larger and smaller Gaussians, thereby affecting the *tone-mapping* response of the operator. Note that a higher sensitivity detects more edges, resulting in a darker image, thereby also changing the effective tonal mapping. Consequently, parameters ϵ and τ may be adjusted in tandem to achieve a desired result (Fig. 4), such as the *thresholding* in Sec. 4.2. The \tanh function in Eq. (6) creates an adjustable soft ramp between the edge and non-edge values, with parameter φ controlling the *steepness* of this transition. Fig. 17 demonstrates a variety of parameter variations: Image (b) shifts ϵ below zero and,

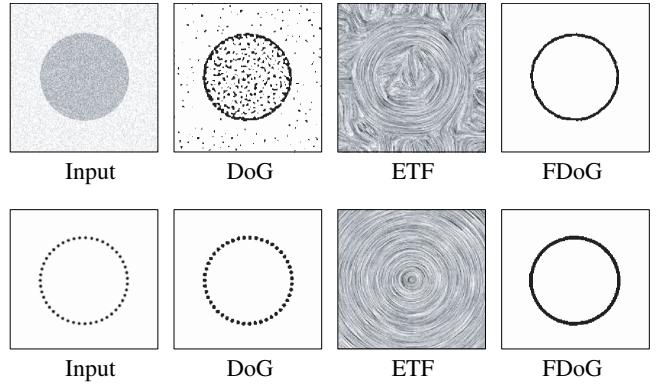


Figure 5: Kang et al.'s [2007] FDoG results: (Top Row) The response of an FDoG filter is less susceptible to noise than an isotropic DoG filter. (Bottom Row) The noisy contour of a circle may be fused by the FDoG operator, thereby increasing coherence.

together with a Gaussian variance of $\sigma = 0$ implements pure tone-mapping. Compare this with images (c) and (d) with increasing σ , and note specifically the increase in local contrast as the edge detector distinguishes edges (black) and negative edges (white, see Sec. 3.4). Figs. 17(e) and (f) demonstrate how ϵ and τ are co-adjusted to retain global contrast, while achieving different line-art appearances. Together, the XDoG parameters ϵ , τ , and φ enable a range of styles and effects, as evidenced in this paper.

Flow alignment Equations (1)-(3) are all 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 (false positives), as in Figure 5. One solution might be to lower the edge sensitivity, ϵ , however, this also lowers the likelihood of detecting larger, significant edges (false negatives). Another solution, proposed in recent works, is to locally adapt the evaluation of the DoG operator to the image content [Kang et al. 2007; Kyprianidis and Döllner 2008; Kang et al. 2009]. Specifically, these approaches no longer compute the DoG operator isotropically, but guide it perpendicular to the direction of image gradients, along the *edge tangent flow* (ETF). These gradients may be estimated using the *Sobel* operator, discussed above. In essence, a flow-aligned DoG (FDoG) is similar to a single-step recursion, in which edges are first weakly estimated using image gradients, followed by another edge detection pass based on these first estimates. The net effects are better noise suppression (fewer spurious edges, Fig. 5, top row) and significantly increased coherence (longer, connected edges, Fig. 5, bottom row).

Variable-threshold FDoG Note that 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. (6) to produce stylistic variations, while we rely on the FDoG implementation of Kyprianidis et al. [2008] for noise suppression and increased coherence.

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*.

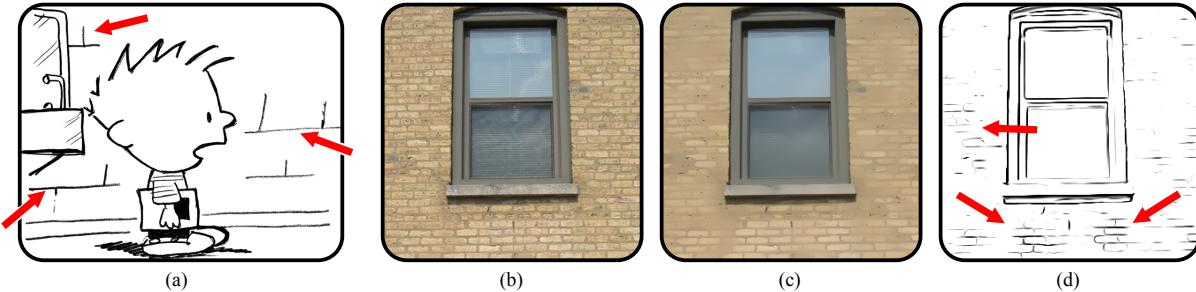


Figure 7: Indication: (a) Cartoonists commonly only indicate unimportant background elements (from Calvin&Hobbes, ©Bill Watterson). (b) A brick wall with window. (c) Bilateral filter result of (b) with simplified texture. (d) Automatic DoG indication of (b) by detecting edges on (c).

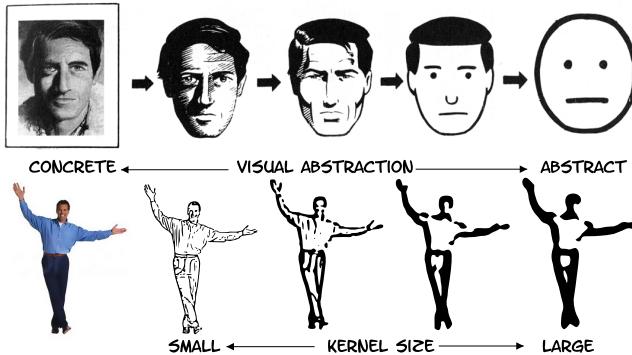


Figure 6: Abstraction: (Top) A successive reduction in level-of-detail abstracts the portrait of a man from a concrete instance to a generic representation (©1993 Scott McCloud). (Bottom) A DoG filter automatically abstracts with increasing kernel size, σ .

There exist dedicated systems to produce most of the effects presented in this Section. The advantage of those systems is 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 minimalist 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 6, 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 [Witkin 1983]. 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 6 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 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 adapt at detecting visual patterns [Palmer 1999], 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 7(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. [2006] 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 [Winnemöller et al. 2006; DeCarlo and Santella 2002]. However, it is worth mentioning that *abstraction* and *indication* remain some of the fundamentally unsolved problems in non-photorealistic rendering.

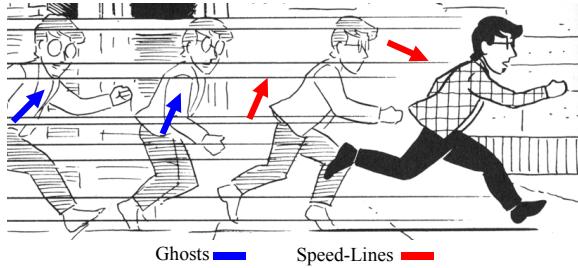


Figure 8: Stylized Motion: Speed and motion are represented in cartoons using speed-lines and ghosting (©1993 Scott McCloud).

ing (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 [1990].

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 8.

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. (2)), except that the blur kernel is not a radially symmetric Gaussian shape, but rather an elongated line in the direction of motion. Convolving an image with such a kernel blurs detail along that line, but not perpendicular to it. Comparing Figs. 9(a) and (b) illustrate this principle. The car is traveling horizontally, resulting in a horizontal motion blur. A horizontal feature, like the hood of the car, is blurred with itself, therefore remaining relatively unperturbed. A vertical feature, like the back of the car, is blurred 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. 8. Such an effect emerges from a simple DoG operator for certain video inputs. Fig. 9(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 edges of the strobed object are detected by the DoG operator and represented as ghosts.

Dedicated systems exist for the generation of cartoon-style motion-effects [Collomosse et al. 2005; Kim and Essa 2005; Linz et al. 2010], which allow for more control over the effect parameters. However, these systems require either 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

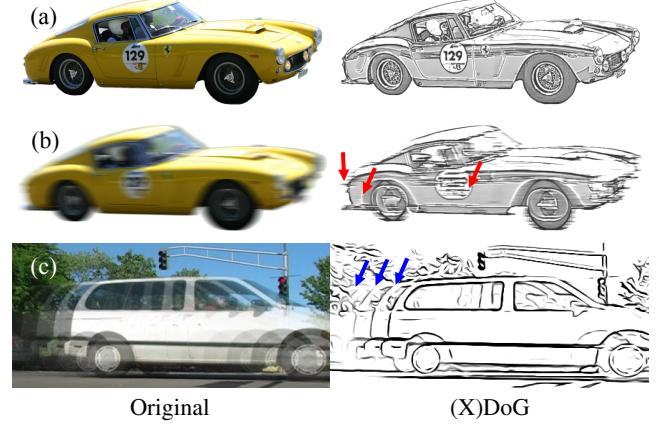


Figure 9: Speed-lines: (a) Images of a stationary car, and corresponding DoG image; (b) Moving car with motion-blur, and DoG image with speed-lines; (c) Moving car with strobing, and DoG image with ghosting.

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

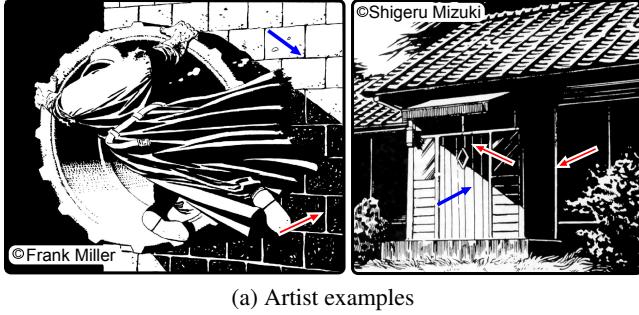
Some artists, such as Frank Miller or Shigeru Mizuki (Fig. 10a), have mastered the depiction of images with just two tones (black and white), lending their artwork a dramatic, stark look. Given such a limited palette these artists use inversion techniques to depict scene detail in dark image regions. Fig. 10(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 [2007] demonstrated how such inversion techniques can be applied to better illustrate 3D models. Similarly, Lee et al. [2007] added bright highlight lines as additional shape cues to 3D models. For images, Rosin and Lai [2010] 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 positive and negative edges 'out-of-the-box' (Figs. 10 b,c). The reason for this becomes evident when re-examining Figs. 3(a) and 4(d): Because the DoG profile exhibits both minima and maxima, we can produce dual-edge detection if we adjust τ and ϵ appropriately. This effect is shown in Figs. 17(c,d), where black edges are surrounded by white edges. This dual-nature of XDoG edges is best visible with soft-thresholding ($\varphi \in \{1 \dots 10\}$). For step-like thresholding (Sec. 4.2), positive edges are only visible in light regions, while negative edges are only visible against a dark background.

4 Styles

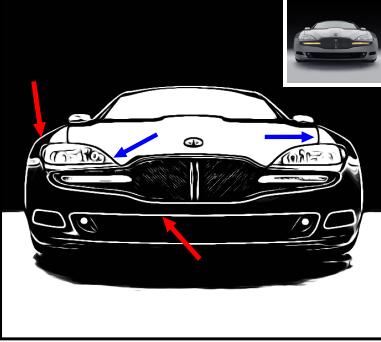
In this Section, we demonstrate several visual styles that can be achieved with the XDoG operator. We start off with demonstrating a few parameter variations which produce *natural-media* appearances in Sec. 4.1. Then, we focus on a subset of this parameter space that implements *thresholding* in Sec. 4.2. Finally, we combine several thresholding results to achieve *hatching* in Sec. 4.3.



(a) Artist examples



(b) XDoG



(c) XDoG

Figure 10: 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).

4.1 Natural Media

The various XDoG parameters may be adjusted to allow for a previously unreported range of natural-media-style appearances. The *pencil-shading* look of Fig. 1(c) 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 $\epsilon < 0$, $\tau < 1$, and $\varphi \approx 1$. The *charcoal* appearance of Fig. 11(c) is due to a much larger spatial support ($\sigma \approx 7$), creating broad strokes. The *pastel* style in Figs. 11(b) and 18(c,d) can be achieved with an intermediate $\sigma \approx 2$. The latter two styles employ the flow-based smoothing of Kyrianiidis et al. [2008], with an identical flow-field computation. Here, the ETF is only minimally smoothed and the FDoG integration uses a relatively large kernel along the flow-direction, resulting in noticeable turbulence and noise along the image edges, which appears as dry brush or charcoal on canvas, depending on the spatial support and local contrast. We achieve the colored pastel look in Fig. 18(d) by modulating the natural media appearance of Fig. 18(c) with source image colors, which are weighted by the negative of Fig. 18(c).

4.2 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. 10a).



(b)



(a)



(c)

Figure 11: Variations: Parameter variations of the XDoG operator produce a range of natural media looks. (a) Source; (b) Pastel; (c) Charcoal. (Source © Flickr user tibchris)

Related Work Various recent works have re-examined the problem of thresholding. Gooch et al. [2004] 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 [2008] 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 [2008] 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

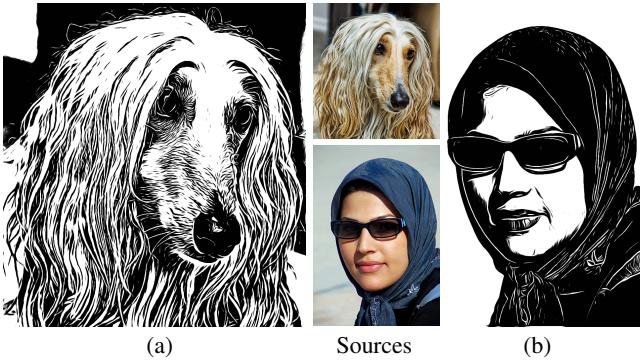


Figure 12: Thresholding: (a) High-detail Thresholding; (b) Large flow distortion and negative edges result in Woodcut appearance.

and Lai [2010] 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 [2007] 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 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, we demonstrate that similar results 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. 12 are obtained by simply applying the XDoG operator (Eq. (6)) with appropriate parameter values. Setting $\tau < 1$ changes the XDoG operation from pure edge-detection to tone-mapped edge-detection, while setting $\epsilon < 0$ co-adjusts the local contrast, as desired. Setting $\varphi \gg 1$ effectively changes the *tanh* soft-ramp in Eq. 6 to a step-function. Note the use of negative edges in Fig. 12(a,b) and the *woodcut* appearance of Fig. 12(b), achieved by prominence of negative edges and extreme smoothing of the FDoG’s ETF field. More threshold results are shown in Figs. 1(d), 10(b,c), and 18(a). Note also the direct comparison with Xu and Kaplan [2008], Figs. 14, 15, and Mould and Grant [2008], Fig. 16.

4.3 Hatching

We can leverage the results of multiple threshold operations in conjunction with scanned hatching textures, to achieve a convincing hatching rendition of an image, as in Fig. 13. Our approach is based on the concept of tonal art maps, where layers of strokes add up to achieve a desired tone [Winkenbach and Salesin 1994; Praun et al. 2001]. First, we compute a standard DoG edge image (Fig. 13(a), top). Then, we set $\epsilon < 0$, $\tau < 1$, $\varphi \gg 1$, as above. To obtain multiple threshold results, we keep ϵ and φ constant and merely adjust τ . 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. 13(b). For hatching with global directions, the textures may be pre-computed using texture synthesis [Praun et al. 2000; Cohen et al. 2003]. For hatching with local directions, local regions are defined by segmenting the source (colored) image (e.g. using Graph-cuts) and then masking regions that overlap with the

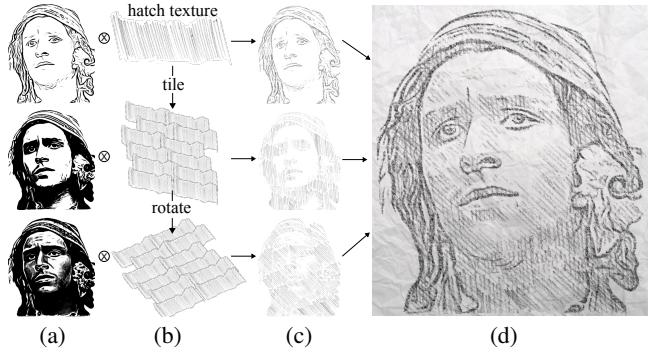


Figure 13: Hatching: (a) Threshold variations; (b) Tiled and rotated hatching textures; (c) Textures in (b) masked by (a); (d) Final result by multiplying all (c)’s onto a paper texture.

threshold results. Each region is then tiled independently, as above, with a local hatching direction.

The results in Fig. 13(a) may be used as masks for the hatching textures in Fig. 13(b), and multiplied together to compute the final image. For added effect, the hatching output can be composited onto a paper texture, as in Fig. 13(d). While simple to implement and efficient to compute, the hatching style retains many of the XDoG features (clean edges, tone control with 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. 18(b).

5 Conclusion

The DoG operator has been employed in a variety of applications, ranging from computer vision [Witkin 1983] to stylistic rendering [Gooch et al. 2004; Winnemöller et al. 2006; Kang et al. 2007]. However, its use has always been one of a straight-forward edge detection operator. To achieve more complex styles and effects, researchers have resorted to building complex, multi-stage systems, often including expensive optimization schemes [Mould and Grant 2008; Xu and Kaplan 2008; Rosin and Lai 2010]. In this paper, we have demonstrated that many of these styles and effects can be achieved more directly and computationally efficient with slight variations of an extended DoG formulation. We hope that our theoretical review of the DoG operator and its recent extensions help to broaden the community’s understanding and interest in the operator’s potential beyond mere edge-detection.

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 [Marr and Hildreth 1980]. Evidence suggests that speed-lines and ghosting are not merely artistic fancy, but may be traced back to the physiology of human vision [Kim and Francis 1998]. 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 [Young 1987]. In general, it might be fruitful to start with simple formulations (such as the XDoG) and investigate, and possibly expand them, to simulate more complex stylistic appearances. This could help us to understand the motivation and ubiquity of certain effects, and it could aid us in identifying more satisfying solutions to the problems of *indication* and *abstraction*.

Appendix A: Efficient Implementation

Note that the Gaussian operator, G , in Eq. (2) 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_0 , Eq. (3). 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 $D_0 = I_{G1_y} - I_{G2_y}$.

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 co-ordinate and computing I_{G1_x} and I_{G2_x} simultaneously. If the number of channels in the 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

- CANNY, J. F. 1986. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 8, 769–798.
- COHEN, M. F., SHADE, J., HILLER, S., AND DEUSSEN, O. 2003. Wang tiles for image and texture generation. *ACM Trans. Graph.* 22 (July), 287–294.
- COLLOMOSSE, J., ROWNTREE, D., AND HALL, P. 2005. Rendering cartoon-style motion cues in post-production video. *Graphical Models* 67, 6, 549 – 564.
- DECARLO, D., AND RUSINKIEWICZ, S. 2007. Highlight lines for conveying shape. In *NPAR '07*, ACM, 63–70.
- DECARLO, D., AND SANTELLA, A. 2002. Stylization and abstraction of photographs. *ACM Trans. Graph.* 21, 3, 769–776.
- GOOCH, B., REINHARD, E., AND GOOCH, A. 2004. Human facial illustrations: Creation and psychophysical evaluation. *ACM Trans. Graph.* 23, 1, 27–44.
- KANG, H., LEE, S., AND CHUI, C. K. 2007. Coherent line drawing. In *NPAR '07*, ACM, 43–50.
- KANG, H., LEE, S., AND CHUI, C. 2009. Flow-Based Image Abstraction. *IEEE Trans. on Visualization and Computer Graphics* 15, 1, 62–76.
- KIM, B., AND ESSA, I. 2005. Video-based nonphotorealistic and expressive illustration of motion. In *Proc. of Computer Graphics International (CGI 05)*, 32–35.
- KIM, H., AND FRANCIS, G. 1998. A computational and perceptual account of motion lines. *Perception* 27, 785–797.
- KOENDERINK, J. J. 1984. What does the occluding contour tell us about solid shape? *Perception* 13, 321–330.
- KYPRIANIDIS, J. E., AND DÖLLNER, J. 2008. Image abstraction by structure adaptive filtering. In *Proc. EG UK Theory and Practice of Computer Graphics*, 51–58.
- LINZ, C., LIPSKI, C., ROGGE, L., THEOBALT, C., AND MAGNOR, M. 2010. Space-time visual effects as a post-production process. In *Proceedings of the 1st international workshop on 3D video processing*, ACM, 3DVP '10, 1–6.
- MALIK, J., AND PERONA, P. 1990. Preattentive Texture Discrimination with Early Vision Mechanisms. *Journal of the Optical Society of America A* 7, 5, 923–932.
- MARR, D., AND HILDRETH, E. C. 1980. Theory of edge detection. *Proc. Royal Soc. London, Bio. Sci.* 207, 187–217.
- MOULD, D., AND GRANT, K. 2008. Stylized black and white images from photographs. In *NPAR '08*, ACM, 49–58.
- PALLÁS-ARENY, R., AND WEBSTER, J. G. 1999. *Analog Signal Processing*. Wiley-IEEE.
- PALMER, S. E. 1999. *Vision Science: Photons to Phenomenology*. The MIT Press.
- PERONA, P., AND MALIK, J. 1991. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12, 7, 629–639.
- PRAUN, E., FINKELESTEIN, A., AND HOPPE, H. 2000. Lapped textures. In *Proc. of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, 465–470.
- PRAUN, E., HOPPE, H., WEBB, M., AND FINKELESTEIN, A. 2001. Real-time hatching. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, SIGGRAPH '01, 581–586.
- RODIECK, R. W. 1965. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research* 5, 11, 583–601.
- ROSIN, P. L., AND LAI, Y.-K. 2010. Towards artistic minimal rendering. In *NPAR '10*, ACM, 119–127.
- WINKENBACH, G., AND SALESIN, D. H. 1994. Computer-generated pen-and-ink illustration. In *Proc. of ACM SIGGRAPH 94*, 91–100.
- WINNEMÖLLER, H., OLSEN, S. C., AND GOOCH, B. 2006. Real-time video abstraction. *ACM Trans. Graph.* 25, 3, 1221–1226.
- WITKIN, A. P. 1983. Scale-space filtering. In *8th Int. Joint Conference on Artificial Intelligence*, 1019–1022.
- XU, J., AND KAPLAN, C. S. 2008. Artistic thresholding. In *NPAR '08*, ACM, 39–47.
- YOUNG, R. A. 1987. The gaussian derivative model for spatial vision: I. retinal mechanisms. *Spatial Vision* 2, 273–293.
- YUNJIN, L., LEE, M., SEUNGYONG, L., AND F., H. J. 2007. Line drawings via abstracted shading. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, ACM, 18.

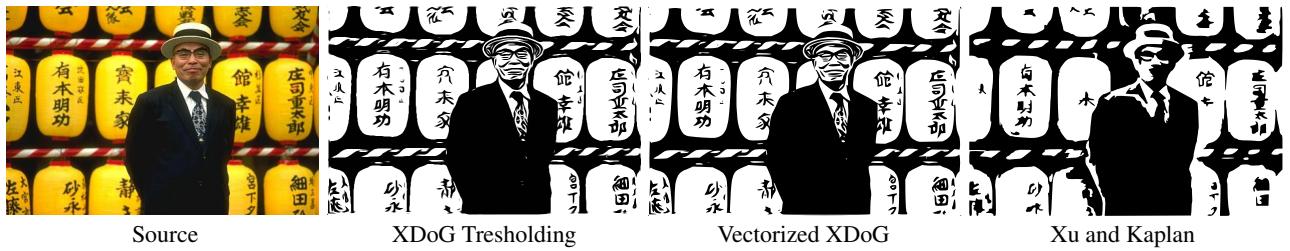


Figure 14: Comparison with Artistic Thresholding (Xu and Kaplan [2008])

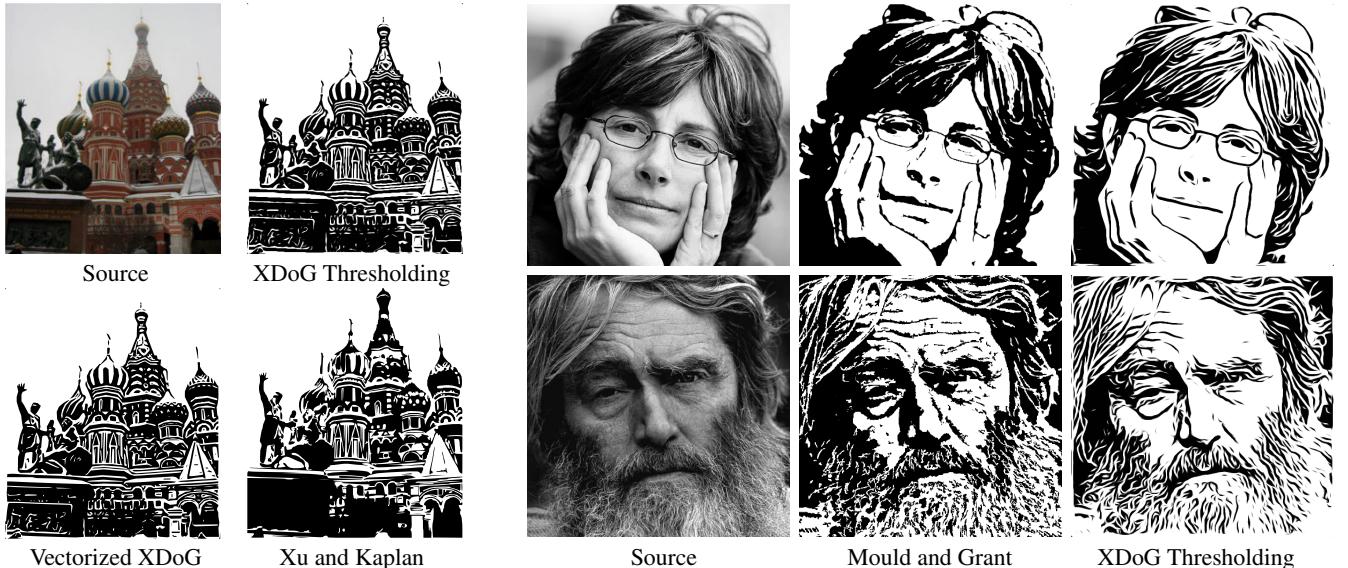


Figure 15: Comparison with Xu and Kaplan [2008]

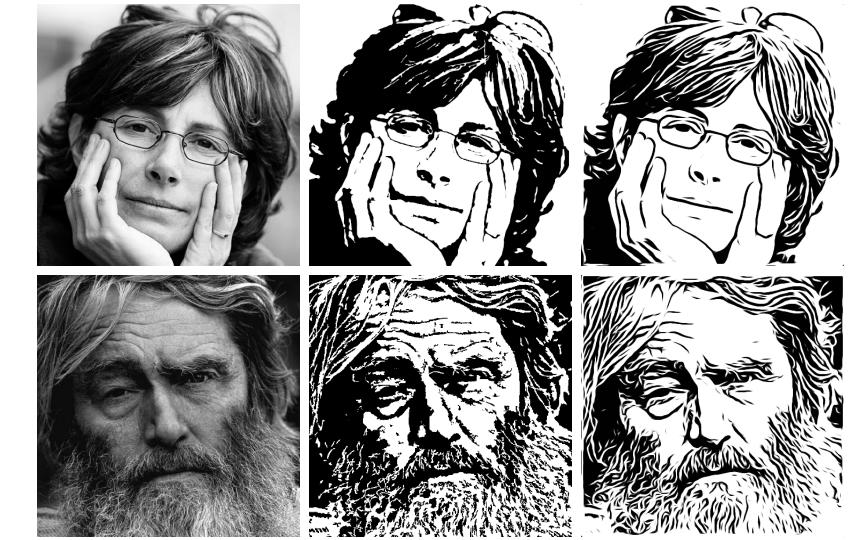


Figure 16: Comparison with Mould and Grant [2008]

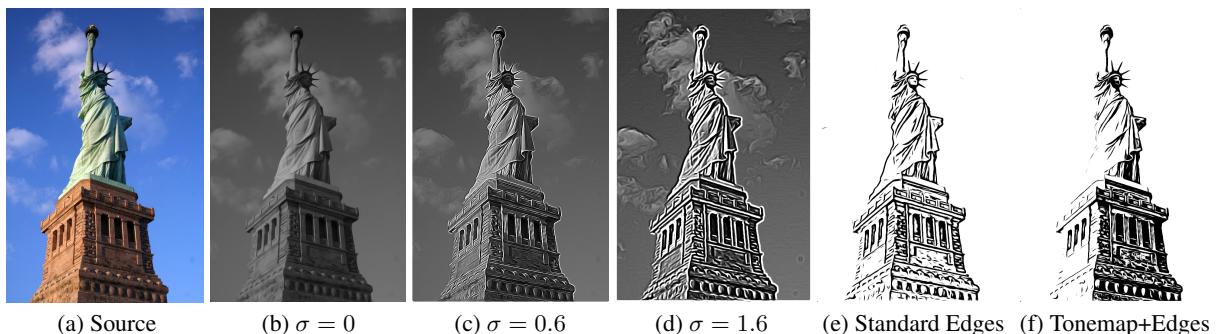


Figure 17: Parameters: (b) $\sigma = 0 \rightarrow$ pure tone-mapping; (c,d) $\sigma > 0$ increases local contrast; (e) $\epsilon = 0, \tau = 0.998$; (f) $\epsilon = -5, \tau = 0.914$



Figure 18: Various Additional Results. Please zoom in to see fine details.

