

Decoupling Strokes and High-Level Attributes for Interactive Traditional Drawing

Frédo Durand[†], Victor Ostromoukhov^{†‡},
Mathieu Miller[‡], François Duranleau[‡], and Julie Dorsey[†]

[†]Lab for Computer Science - MIT

[‡]Université de Montréal, LIGUM

Abstract. We present an interactive system, which allows the user to produce drawings in a variety of traditional styles. It takes as input an image and performs semi-automatic tonal modeling. Our system shifts tedious technical aspects to the computer side, while providing the user with freedom on the creative and aesthetic side. The user has low-level control over stroke placement, and high-level control over the tone, smudging and amount of detail. The drawing is rendered in real-time. The basic component is a thresholding model of strokes that can simulate a large class of styles (e.g. pencil, charcoal, engraving). It provides a controllable simulation of the variation of pencil pressure or stroke thickness traditionally used in tonal modeling. We introduce a novel fast equilibration approach for the resulting thresholding structure. The user can specify smudging and control the amount of detail over each part of the drawing.

1 Introduction

The broad appeal of drawing lies in its expressiveness, precision, and simplicity. The versatile role of strokes in conveying form, tone, shape, texture, and style is the source of both the richness and the difficulty of the medium. Our long-term goal is to shift the tedious technical aspects of drawing to the computer side, while providing the user with expressiveness and a new kind of freedom on the creative and aesthetic side. Our intention is not to replace traditional drawing, but to provide a fast and flexible tool to create realistic drawings. We believe our system will benefit both skilled artists due to its editing capabilities, and novices who wish create expressive drawings without tedious technical learning.

We extend the work on interactive image-based non-photorealistic rendering systems [Hae90, Ost99, PB94, SABS94, SWHS97]. As in these works, a reference photograph is used to provide both a visual aid for the user and a reference for the tones of the final image (Fig. 1). Using realistic tones as an input is not as restrictive as one may think. The control of the tone reproduction curve or burning and dodging techniques provide a large freedom [Ada95, Sch99]. Photography, slide projectors, and well before them the *camera obscura* and *camera lucida* have been used by artists as drawing aids for accurate perspective, and sometimes for tones [Kem90, Kos99, Smi87].

In our approach, we decouple the specification of strokes from higher-level attributes such as conveyed tone and precision. An attribute can be varied without affecting the others. The separation of these dimensions of drawing enablesthe user to experiment with a wide range of aesthetic choices.

1.1 Previous work

A variety of systems for non-photorealistic rendering [Elb95, Elb99, LS95, MKT⁺97, ST90, WS94] construct imagery based on a 3D model. In contrast, we use a reference image to specify tones, as pioneered by Haeberli [Hae90].

Similar to our approach, Salisbury et al. [SABS94, SWHS97] generate pen-and-ink drawings starting from grey-scale images. While they capture a rich range of tonal



Fig. 1. Basic features of our interactive drawing system. The user provides a reference photograph (upper-left), edits the tones, draws strokes and specifies a precision map (lower-left) while interactively viewing the drawing (right).

and texture variations, they are limited to a specific drawing style, and in particular, to simple strokes. In their work, darker tones are obtained by adding more strokes, while our thresholding approach achieves this effect by varying the thickness and pressure of strokes. Our system shares many features with theirs, but extends it in two important respects: by providing a richer collection of strokes, and a higher degree of control, particularly in the level of detail that can be achieved. It also enhances interactivity by providing a continuous visual feedback.

Complex simulations of paper-pencil interactions have been proposed by different authors [TFN99, SB99b, SB00]. The results are impressively faithful, but their approach does not yet permit a large variety of stroke styles. However, we believe that these models could be used to generate new stroke textures for our system. Emulating traditional media is only the first step before such techniques can be applied to 3D models (as was done in [SB99a]) or to provide a user with additional features, as we propose in this paper. Our model of strokes is simpler to derive (new stroke style can be acquired using a simple scanner) and better suited for semi-automatic tonal modeling.

Our simulation of strokes is inspired by research on digital engraving [Lei94, PB94, Ost99] and, in particular, builds upon the work by Ostromoukhov [Ost99], where simple engraving strokes are simulated using a halftoning approach known as *thresholding*. Our model of strokes can handle a larger variety of media, and is included in a real-time system providing interactivity both at a high and low level. Finally, due to our new hardware-assisted thresholding, drawings can be rendered in real-time.

1.2 Overview

This paper makes the following specific contributions:

Decoupling strokes and high-level attributes: The user independently controls the placement of strokes, and higher-level properties such as tonal adjustment and preci-

sion.

Stroke model: We extend a thresholding technique to simulate a large class of styles such as pencil, charcoal, or engraving. Our model supports smudging, widely used in traditional drawing (usually by rubbing a finger or a piece of paper on the drawing).

Equilibration: Thresholding requires a matrix with a flat histogram, which can be obtained by *equilibration*. We present a new rapid probabilistic technique.

Interactive hardware implementation: Our drawing rendering algorithm is hardware-assisted and easy to implement.

2 Decoupling expressiveness and technique

We first place our work in the context of traditional drawing in order to motivate our decoupling of the user interface from image generation.

2.1 Context: the art of drawing

Inspired by different authors [Edw99, Gai96, Hal64, Men76, Spe17], we distinguish the following aspects of drawing: Gestalt, perspective, outlines, tonal modeling, and control of detail. For each category, it is important to make a distinction between the aesthetic choice and the technical achievement of this intent. It is also crucial to understand in what these aspects can be similar and different in drawing and photography.

Gestalt deals with the balance and composition of the image. Since it lies completely on the aesthetic side, our system leaves this aspect to the user. Photography has been widely used as a reference for realistic drawing [Kem90, Kos99, Smi87]. The same is true in our system, where the input image defines the perspective.

Outlines are fundamental visual components of line drawing. Similar to Salisbury et al., [SABS94], we use edge detection as a visual aid. We also offer optional outline snapping in the spirit of magic scissors [MB95]. Outline strokes are excluded from the automatic tonal modeling.

Tonal modeling in photography is surprisingly similar to the comparable process in drawing and painting. The goals are the same: reveal shape and texture, set the mood, and balance the image. Although drawing and photography usually exhibit different tonal styles, darkroom tools [Ada95, Sch99] can be used to give a photograph the tonal style of a drawing. The distinction between the choice of tones and the practical achievement of tones is crucial. Our system manages the achievement of tones, using an adapted thresholding approach. The target tones are specified by the reference image. However, the user is not limited by the initial photograph, as darkroom tools are provided to interactively alter the tones of the reference image to match the user's intent.

An often overlooked aspect of visual arts is the control of the amount of detail. In photography, it is possible to control only depth of field and use filters, while in drawing it is possible to finely control the amount of detail over the whole image, extracting the visual essence, focusing the eye on important areas and establishing the balance between strokes and the depicted subject. Controlling the precision is a challenging and subtle task. An initial solution was proposed by Winkenbach et al. [WS94] using *detail segments*. In our system, a *precision map* locally controls the accuracy of tonal modeling and the amount of spatial detail.

2.2 A proposed decomposition of drawing

When an artist draws a stroke, all five aspects come into play, hence the difficulty for beginners. Our system attempts to decouple these aspects and to provide flexible high-

level editing tools, as well as low-level stroke control. For this, we express image generation as a multidimensional mapping from a user control-space to an imaging control-space. Another benefit of this framework is a simpler implementation.

Central to our analysis is the distinction between medium simulation and user control (Fig. 2). The medium simulation is the “mechanical” imitation of (usually traditional) media. It is driven by what we call the *imaging control space*. In our thresholding-based system, the imaging control space consists of a *threshold matrix*, a *target tone* image and the amount of *smudging*. Image generation involves comparing the threshold matrix against the target tones, and uses the amount of smudging to compute a grey level for each pixel.

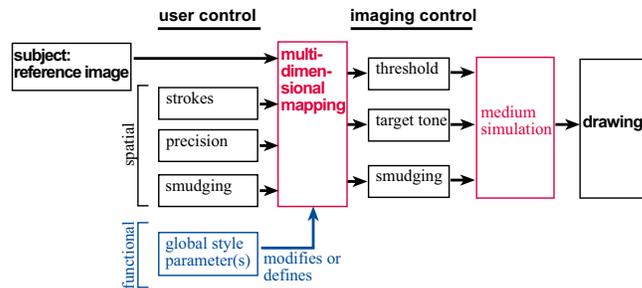


Fig. 2. Drawing generation interface as a multidimensional mapping from user control space to imaging control space. Since the user has direct control over the amount of smudging, it appears both in user and imaging control-spaces, and the sub-mapping is identity.

In contrast, the *user control space* describes the degrees of freedom of the user, which largely defines interface. The image subject is specified by an input photograph. We make a distinction between spatial input (stroke location, amount of precision, smudging), and global or functional input that affect the whole rendering *style*. Our only functional parameter specifies the influence of the precision map: whether it removes details in the target tones, or whether it favors strokes specified by the user vs. target tone accuracy.

Image generation can then be described as a multidimensional mapping from the spatial user control space to the imaging control space. Global user inputs affect or define the mappings. When the user has direct control on imaging parameters, the corresponding mapping is the identity mapping (this is the case for smudging).

2.3 Algorithm overview

The final output of our system is a pixel image, but the image generation data structures are a set of strokes and pixel maps representing target tone, precision and smudging. Unlike purely image-based systems (such as Corel Painter), the final resolution of the image need not be fixed, since the drawing is constantly regenerated from the strokes. The target tone, precision and smudging maps are scaled using the hardware. This permits device-independent interactive editing and device-specific final rendering (in the spirit of [SALS96]).

Our algorithm is summarized in Fig. 3 and will be detailed in the next sections. The strokes are used to render a raw threshold matrix. This raw matrix needs equilibration to compensate for the non-flat histogram resulting from irregular stroke placement. The amount of equilibration is affected by the precision map.

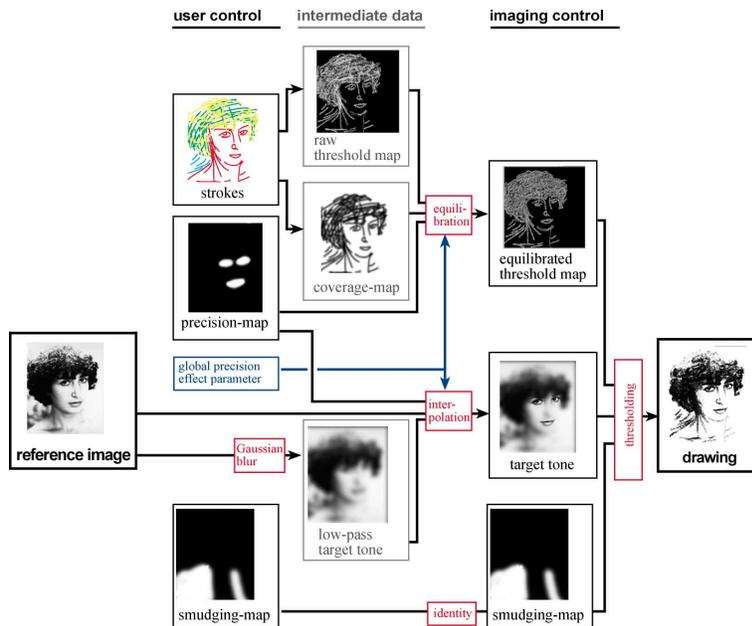


Fig. 3. Flowchart of our system. Drawing rendering is expressed as a set of 3-dimensional sub-mappings. Non-tonal strokes are not represented; they “bypass” the target tone.

The target tone image used for thresholding is a filtered version of the input image, where the amount of detail is specified by the precision map. As aforementioned, smudging is a direct parameter specified by the user.

All these mappings are implemented using the graphics hardware and the *pixel textures* [HWSE99, BM00]. This feature permits the use of the color buffer as texture coordinates for 3D texture mapping. This permits efficient mapping from a 3-dimensional space to a 3-dimensional space. Tri-linear interpolation is performed. This makes the implementation of our technique easy and fast. We render the relevant space into each channel, set the look-up-table corresponding to the desired mapping, and make a buffer-copy that performs the look-up [HWSE99, BM00].

3 Strokes and thresholding

3.1 Thresholding

In digital halftoning, tone is obtained by comparing at each pixel a threshold matrix $T(x,y)$ against the target tone of the input image $I(x,y)$. Depending on the result, the pixel is declared black or white [Uli87]. If S is the step function ($S(x) = 0$ if $x < 0$ and $S(x) = 1$ if $x \geq 0$), thresholding can be expressed as:

$$Thresh(T(x,y), I(x,y)) = S(T(x,y) - I(x,y)) \quad (1)$$

If the histogram of the threshold matrix T is flat (all values are equally represented), faithful tone reproduction is obtained. However, the irregular placement of strokes does not result in a flat histogram. For faithful tonal rendering, an equilibration has to be performed, as will be presented in Section 4.

In our system, strokes are simulated using local threshold structures, which we call *threshold textures*. The initial principle of our rendering is simple. To render the drawing, we render all the strokes using their threshold texture, resulting in a global threshold matrix $T(x, y)$, which is then compared to the target tones $I(x, y)$ to obtain a black and white drawing. This is the key in decoupling stroke style from tonal value: the threshold texture defines the stroke style, while tone is obtained by thresholding with respect to the target tone (Fig. 4).

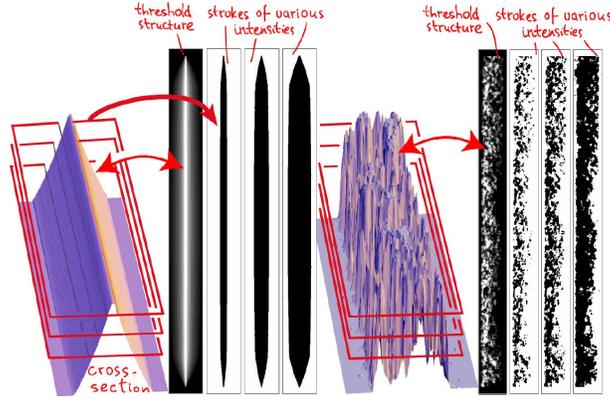


Fig. 4. Typical stroke threshold structure for etching (left) and pencil (right). We show various tonal effects conveyed by single stroke structures. The key idea is to truncate the threshold structure at a different height depending on the target tone.

When multiple strokes overlap, their merging must receive particular attention. We use the maximum blending mode, which effectively imitates the overlap of multiple traditional strokes [Ost99].

3.2 Antialiasing

Thresholding was developed for tonal rendering on high resolution devices that can only produce black dots [Uli87]. In contrast, monitors have a lower resolution but a larger gamut. We propose to modify the step function S in equation 1 to introduce smoothness, using an S-shaped function. Special care must be paid to symmetry in order to maintain tonal fidelity. The initial idea is to use a clamped sine function, with width $2w$. Using t for $T(x, y)$ and i for $I(x, y)$:

$$s(t, i, w) = \begin{cases} 1, & t - i < -w \\ 0.5 + 0.5 \sin\left(-\frac{\pi(t-i)}{2w}\right), & -w \leq t - i \leq w \\ 0, & w < t - i \end{cases} \quad (2)$$

This works fine for middle tone, but introduces bias for low and high tones, because the clamping to black or white obviates symmetry. We thus limit the width of the s function to the distance of the tone x to black or white $d(x) = \min(1 - x, x)$ and use:

$$s'(t, i, w) = s(t, i, \max(w, d(i))) \quad (3)$$

By default we use $w = 0.05$, but we will see in Section 5.3 that varying this value mimics the effect of smudging.

3.3 Stroke threshold texture

A *threshold texture* is essentially a threshold structure as described above. It has a flat histogram and is represented as a grey-level texture map. Threshold textures can be either acquired by scanning real strokes (in the spirit of e.g. [VB00]), or modeled directly. We scanned real strokes for pencil and charcoal, and modeled etching-style strokes, and then applied a simple histogram flattening algorithm.

The relevance of this model is not obvious. Clues can be found in the physical process of paper-graphite interaction [SB00]. The grey values correspond to the number of graphite particles per area, and can be seen as the probability that particles are deposited. Thresholding together with our antialiasing technique corresponds to simulating macro-graphite particles. A more phenomenological macroscopic validation is the similarity between pencil drawing and lithography, and the similarity between the thresholding height-field and engraving [Ost99].

3.4 Individual strokes and rendering

Individual strokes are described by a skeleton, a reference to a stroke threshold texture, and a target tone for strokes excluded from automatic tonal modeling. A stroke skeleton is described by a poly-Bézier curve, along which the threshold texture is warped, as described below. The threshold textures are randomly shifted along the curve to avoid uniformity and cross-stroke correlation.

To compute the raw global threshold matrix $T(x, y)$, we transform strokes according to the Bézier skeleton using a triangle-strip band around the curve and texture mapping [GDCV98]. When rendering strokes, we use the OpenGL 1.2 blending mode MAX.

4 Equilibration

The equilibration of threshold structures is usually performed using an iterative trial-and-error approach [Ost99, OH99]. However, this is too slow for our purpose. We therefore introduce a new technique that is both direct and fast. It is not as accurate as previous techniques, but it proves sufficient for drawing. Since the histograms of our strokes are flat, we only need to compensate for overlapping strokes. We base our technique on a probability analysis of the histogram of overlapping strokes merged using the *max* mode.

4.1 Probabilistic analysis

We model the histogram of a stroke as a probability density function $p(x) = 1$ for $x \in [0, 1]$. The cumulative density function (corresponding to the cumulative histogram) is $P[X \leq x] = \int_0^x p(\xi) d\xi = x$. The main assumption is that there is no correlation between strokes. This is reasonable given the texture of our strokes and their irregular placement.

The probability and cumulative density of two overlapping strokes p_1 and p_2 are:

$$\begin{aligned} p_{\max(12)}(x) &= p_1(x) * P_2[X \leq x] + p_2(x) * P_1[X \leq x] = 2x \\ P_{\max(12)}[X \leq x] &= x^2 \end{aligned}$$

This formula can be generalized to the histogram of n overlapping strokes:

$$p_{\max(i=1..n)}(x) = n x^{n-1}, \quad P_{\max(i=1..n)}[X \leq x] = x^n$$

Experimental measurements fit this formula well (Fig. 5).

A flat histogram is then obtained by multiplying each value x by the corresponding cumulative density $P_{\max(i=1..n)}(X \leq x)$ [Bov00]. Equilibration can thus be performed by applying a power function depending on the number of overlapping strokes.

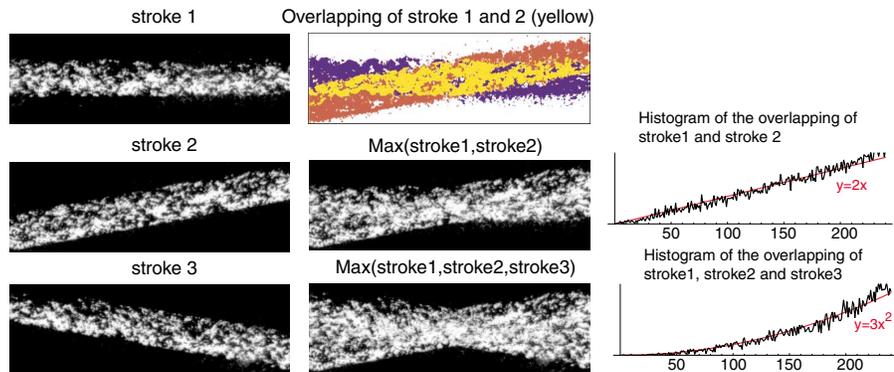


Fig. 5. Histogram of overlapping strokes under the *max* mode.

4.2 Hardware-assisted equilibration

For each pixel, we compute the number of overlapping strokes in a *coverage map* and perform the corresponding monomial mapping. The coverage map is built like the raw threshold matrix by rendering the stroke triangle strips, but using the coverage texture instead of the the threshold texture, and an additive blending mode.

A Gaussian low-pass filter is usually used to simulate the spatial integration of the visual system [Ost99, OH99, Uli87]. If we consider only straight strokes, performing a Gaussian blur in the global matrix is similar to using a Gaussian-filtered texture, which can be pre-computed. We make a linearization and neglect the effect of warping. We use coverage textures that are pre-filtered with a Gaussian (cutoff frequency is 30 cycles per degree [LO86]), and warp them like the threshold textures. We map a coverage of 1.0 to a frame-buffer value of 16, which results in a maximum number of strokes handled by pixel of 16, while providing enough dynamic for the Gaussian (quantization to 16 levels). Equilibration is performed using a look-up table and the pixel-textures hardware (Fig. 6(a)).

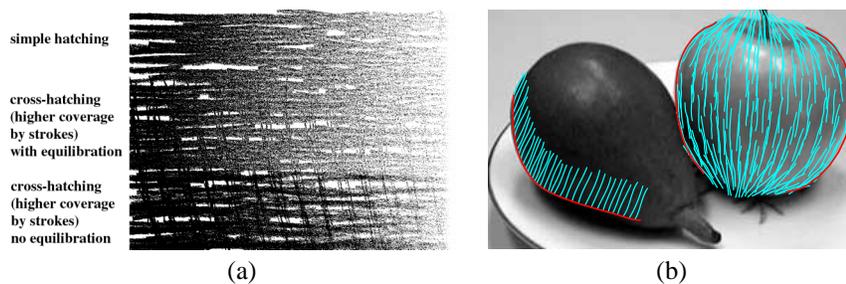


Fig. 6. (a) Effect of equilibration for a target tone image consisting of a grey-level ramp. (b) Hatching tool using one or two carriers (in red).

5 User control

We now discuss in more detail how the user can draw strokes and how he controls the *precision* and *smudging*.

5.1 Stroke drawing interface

The user draws free-hand strokes using the mouse or a tablet. We use a Bézier-fitting method enforcing G^1 continuity for poly-Bézier curves [Sch88]. Strokes are organized into layers to facilitate editing or selection of groups of strokes. We propose rudimentary hatching tools based on the interpolation between one or two carrier strokes and random perturbations (Fig. 6(b)).

5.2 Precision

As discussed in Section 2.1, controlling the amount of detail is a key aspect of drawing. The mastery and subtlety of this skill are unfortunately elusive and remains the prerogative of great masters. Our system however proposes a simple model of precision, controlled spatially through the precision map. We developed two mechanisms, and their respective influence is controlled through two global parameters, which can be seen as a crude attempt to introduce a notion of *style*.

The precision and smudging maps are controlled with the same interface. The user modifies them using an airbrush tool, either directly on the drawing or on the map.

Stroke saliency vs. tonal fidelity. The first precision mechanism controls the respective importance of stroke saliency and tonal fidelity. More salient strokes focus the attention on the 2D quality of the drawing, while a more faithful tone reproduction emphasizes the depicted subject.

Our system uses precision to vary the amount of equilibration. A precision of 1 corresponds to a complete equilibration and faithful tone reproduction, while a precision of 0 corresponds to no equilibration at all and more visible strokes. This mechanism is implemented by modifying the equilibration mapping presented in Section 4. The global importance of this mechanism is controlled by the weight $w_{saliency}$. Because we use the *max* merging mode, a lower precision also results in darker tones in areas covered by more than one stroke. This means that the output tone is then more ruled by the number of strokes than by the target tone.

Spatial detail. The second mechanism controls the amount of spatial detail in the target tone map. It is common artistic practice to use uniform tones for regions of little importance, and fine spatial details for regions of focus. This feature could be implemented using a non-linear blurring filter with a spatially variant Gaussian kernel size. Unfortunately this solution is too slow for real-time implementation. We therefore use a linear interpolation between the fully detailed target tone image and a low-pass version computed using a Gaussian filter (in the spirit of e.g. [KVHS00]). In practice we set the variance of the Gaussian to 5% of the image size.

5.3 Smudging

Traditional smudging softens a drawing by spreading graphite particles. In contrast to the technique by Sousa et al. [SB99b], our implementation of smudging is purely phenomenological. It is an extension of the antialiasing technique described in Section 3.2. It consists in varying the width of the s' function used for thresholding.

The user has direct control on smudging via the smudging map. It is used for imaging during the thresholding phase. The target tone is stored in the red channel, the equilibrated threshold map in the green channel, and the smudging map in the blue channel. Using the function s' defined in Eq. 3, the look-up-table is then $LUT_{thresh} = s'(r, g, b)$. Fig. 7(a) demonstrates the effect of smudging. The use of a spatially-varying blurring filter would result in a similar result, but at a higher cost.

6 Results

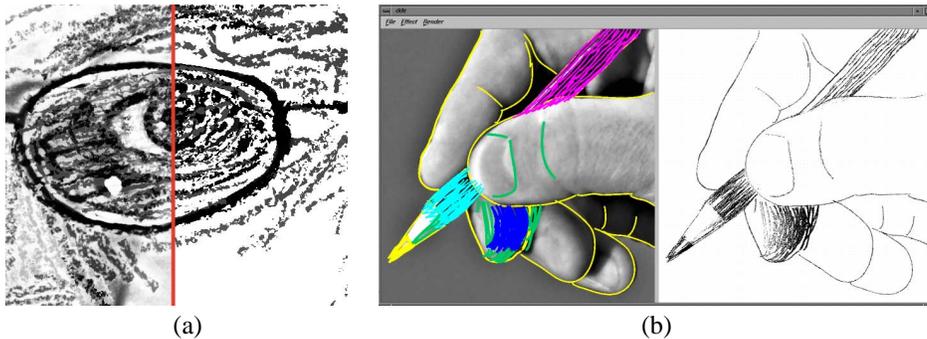


Fig. 7. (a) With (left) and without (right) smudging. (b) Snapshot of an interactive session.

Our system was implemented on Silicon Graphics workstations using the OpenGL graphics API. Fig. 7(b) presents a snapshot of an interactive session, where the drawing is refreshed at about 20Hz on an SGI Octane2 R14K.

Fig. 8(a) imitates red chalk or soft pastel. Color and background paper were added as a post-process. Fig. 8(b) uses etching strokes. Interpolation between carriers was used to generate the large number of strokes. Note the importance of outline strokes. Fig. 1 demonstrates the rich degree of economy that can be obtained, since only a dozen strokes are used for the facial features. Precision is high around the eyes and mouth.

All the examples shown in this paper were generated by a user who is not a trained illustrator. Similar or better results could be obtained by skilled artists using traditional media or a product such as *Corel Painter*. However, this is the goal of this project to make this degree of expressiveness available to a non-artist.

7 Conclusions and future work

We have presented an interactive drawing system based on a thresholding model of strokes. It permits semi-automatic tonal modeling and decouples the technical aspects of drawing from expressiveness. We have developed a fast and direct equilibration technique that is based on a probabilistic model of threshold textures. We have introduced different high-level controls, including a precision map that controls both the amount of spatial detail and the saliency of strokes. Our system provides both interactivity and high-quality output.

An important issue of future work is the introduction of higher-level textures as pioneered by Winkenbach et al. [WS94] and Salisbury et al. [SABS94] for pen-and-ink style. More elaborate hatching tools should also be provided, either using orientation textures [SWHS97] or line-integral-convolutions [TB96].

The use of a photograph restricts the drawing to a realistic perspective. Introducing warping or “caricature” tools could remove this limitation.

The introduction of color is an exciting but delicate topic; we believe that colors cannot be treated simply like tones. A better treatment of outline strokes is a challenging task as well. The complex interaction between outline and tonal strokes accounts for the elusive quality obtained by skilled artists.

The system can also be adapted for automatic drawing from 3D models. Using the discussion in Section 2.2 as a starting point, mappings from 3D properties and user control space could be defined.

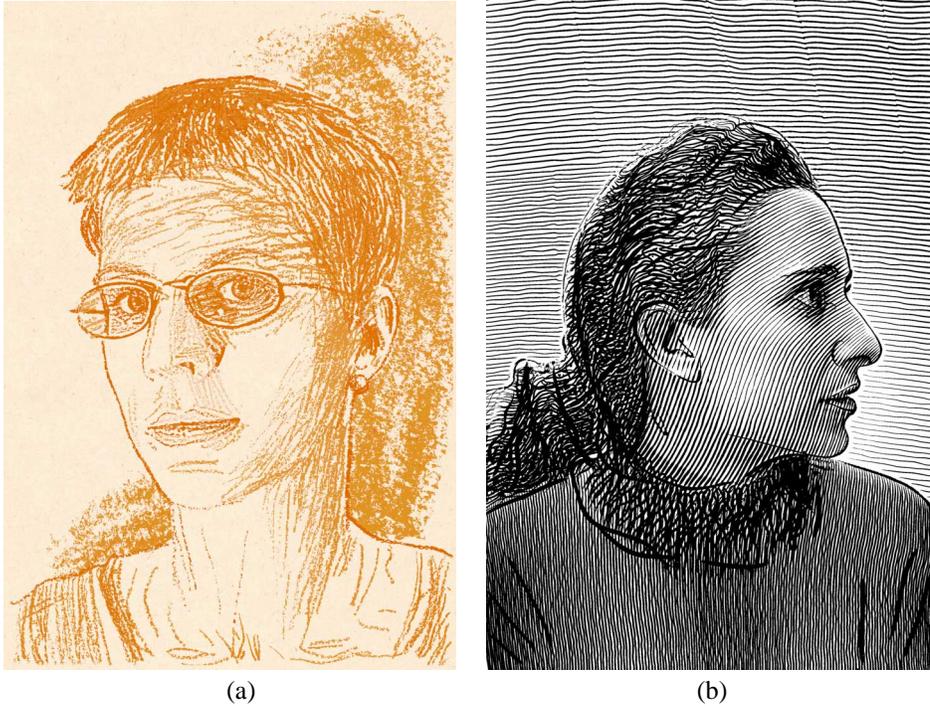


Fig. 8. (a) Sanguine style (red chalk). (b) Engraving style.

Acknowledgments

We wish to thank Edouard Basilia for the photograph used in Fig. 1, and our models, Sylvie Fresco, Kari Anne Kjølås, and Manue Brun. Many thanks to Adam Finkelstein, Justin Legakis, Joëlle Thollot, Agata Opalach and Osama Tolba for helpful discussions and suggestions about the paper. This work was supported by an NSF CISE Research Infrastructure Award (EIA-9802220) and a gift from Pixar Animation Studios. Mathieu Miller was supported by Abvent and François Duranleau by an FCAR grant (B1-03C-68966).

References

- Ada95. A. Adams. *The Negative+The Print*. Little Brown and Co, 1995.
- BM00. D. Blythe and T. McReynolds. Advanced graphics programming techniques using OpenGL. SIGGRAPH course notes, 2000.
- Bov00. A. Bovik, editor. *Handbook of Image and Video Processing*. Academic Press, 2000.
- Edw99. B. Edwards. *The New Drawing on the Right Side of the Brain*. J P Tarcher, 1999.
- Elb95. G. Elber. Line art rendering via a coverage of isoparametric curves. *IEEE Trans. on Visualization and Computer Graphics*, 1(3), September 1995.
- Elb99. G. Elber. Interactive line art rendering of free form surfaces. In *Eurographics*, 1999.
- Gai96. A. Gair, editor. *Artist's Manual: A Complete Guide to Painting and Drawing Materials and Techniques*. Chronicle Books, 1996.
- GDCV98. J. Gomes, L. Darsa, B. Costa, and L. Velho. *Warping And Morphing Of Graphical Objects*. Morgan Kaufman, 1998.

- Hae90. P. Haeberli. Paint by numbers: Abstract image representations. In *SIGGRAPH*, 1990.
- Hal64. R. Hale. *Drawing lessons from the great masters*. Watson Guptill, 1964.
- HWSE99. W. Heidrich, R. Westermann, H.-P. Seidel, and T. Ertl. Applications of pixel textures in visualization and realistic image synthesis. In *ACM Symp. on Interactive 3D Graphics*, 1999.
- Kem90. M. Kemp. *The Science of Art*. Yale Univ. Pr., 1990.
- Kos99. D. Kosinski. *The Artist and the Camera: Degas to Picasso*. Yale Univ. Pr., 1999.
- KVHS00. J. Kautz, P. Vazquez, W. Heidrich, and HP. Seidel. A Unified Approach to Prefiltered Environment Maps. In *Eurographics Rendering Workshop*, 2000.
- Lei94. W. Leister. Computer generated copper plates. *Computer Graphics Forum*, 13(1), 1994.
- LO86. J. Thomas L. Olzak. Seeing spatial patterns. In Boff, Kaufman, and Thomas, editors, *Handbook of Perception and Human Performance*. Wiley and Sons, 1986.
- LS95. J. Lansdown and S. Schofield. Expressive rendering: A review of nonphotorealistic techniques. *IEEE Computer Graphics and Applications*, 15(3), May 1995.
- MB95. E. Mortensen and W. Barrett. Intelligent scissors for image composition. In *SIGGRAPH*, 1995.
- Men76. D. Mendelowitz. *A Guide to Drawing*. Holt, Rinehart and Winston, 1976.
- MKT⁺97. L. Markosian, M. Kowalski, S. Trychin, L. Bourdev, D. Goldstein, and J. Hughes. Real-time nonphotorealistic rendering. In *Proc. of SIGGRAPH*, 1997.
- OH99. V. Ostromoukhov and R. Hersch. Multi-color and artistic dithering. In *Proc of SIGGRAPH*, 1999.
- Ost99. V. Ostromoukhov. Digital facial engraving. In *Proc. SIGGRAPH*, 1999.
- PB94. Y. Pnueli and A. Bruckstein. DigiDürer – a digital engraving system. *The Visual Computer*, 10(5), 1994.
- SABS94. M. Salisbury, S. Anderson, R. Barzel, and D. Salesin. Interactive pen-and-ink illustration. In *Proc. of SIGGRAPH*, 1994.
- SALS96. M. Salisbury, C. Anderson, D. Lischinski, and D. Salesin. Scale-dependent reproduction of pen-and-ink illustrations. In *Proc. of SIGGRAPH*, 1996.
- SB99a. M. Sousa and J. Buchanan. Computer-generated graphite pencil rendering of 3D polygonal models. In *Proc. of Eurographics*, 1999.
- SB99b. M. Sousa and J. Buchanan. Observational model of blenders and erasers in computer-generated pencil rendering. In *Graphics Interface*, 1999.
- SB00. M. Sousa and J. Buchanan. Observational models of graphite pencil materials. *Computer Graphics Forum*, 19(1), March 2000.
- Sch88. PJ Schneider. Phoenix: An interactive curve design system based on the automatic fitting of hand-sketched curves. Master's thesis, University of Washington, 1988.
- Sch99. G. Schaub. *The Digital Darkroom: Black-And-White Techniques Using Photoshop*. Silver Pixel Pr., 1999.
- Smi87. R. Smith. *The Artist's Handbook*. Knopf, 1987.
- Spe17. H. Speed. *The Practice and Science of Drawing*. Dover, 1917.
- ST90. T. Saito and T. Takahashi. Comprehensible rendering of 3-D shapes. In *Proc. of SIGGRAPH*, 1990.
- SWHS97. M. Salisbury, M. Wong, J. Hughes, and D. Salesin. Orientable textures for image-based pen-and-ink illustration. In *Proc. of SIGGRAPH*, 1997.
- TB96. G. Turk and D. Banks. Image-guided streamline placement. In *SIGGRAPH*, 1996.
- TFN99. S. Takagi, I. Fujishiro, and M. Nakajima. Volumetric modeling of artistic techniques in colored pencil drawing. In *SIGGRAPH 99: Conference abstracts and applications*, 1999.
- Uli87. R. Ulichney, editor. *Digital Halftoning*. MIT Press, 1987.
- VB00. O. Veryovka and J. Buchanan. Texture-based dither matrices. *Computer Graphics Forum*, 19(1), March 2000.
- WS94. G. Winkenbach and D. Salesin. Computer-generated pen-and-ink illustration. In *Proc. of SIGGRAPH*, 1994.



Fig. 9. Sanguine style (red chalk).