

PortraitSketch: Face Sketching Assistance for Novices

Jun Xie

Univ. of Washington

Aaron Hertzmann

Adobe Research

Wilmot Li

Adobe Research

Holger Winnemöller

Adobe Research

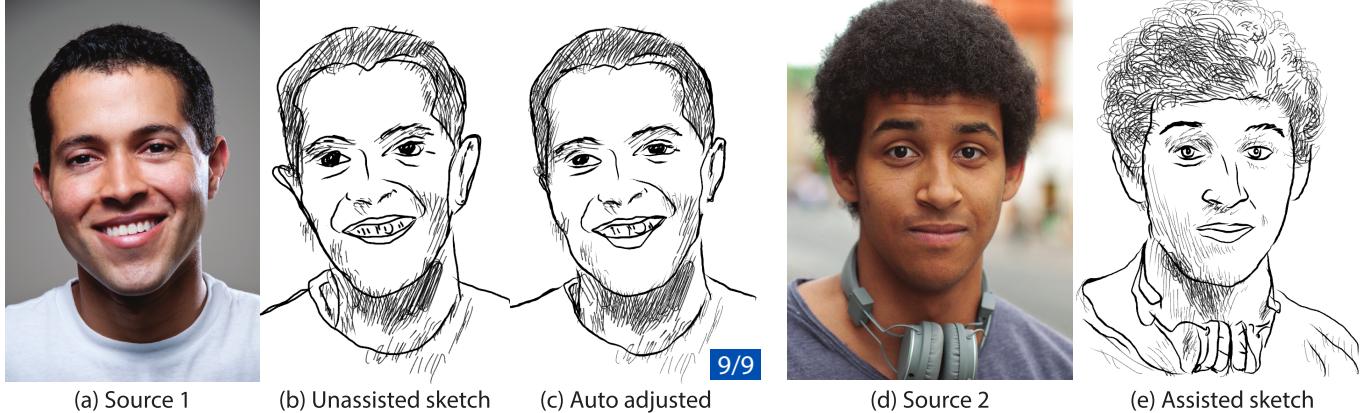


Figure 1: Results generated with PortraitSketch. Starting from image (a), a novice user created sketch (b) using our system without automated assistance. The same user created sketch (e) with automated assistance. For direct comparison, we post-adjusted (b) to generate (c). The assisted results (c and e) exhibit cleaner, more accurate outlines and more evenly distributed shading, compared to the unassisted drawing (b). All nine of our independent judges voted that (c) is a better drawing than (b).
Image credits: (a) and (d): Levy Carneiro Jr and chris zerbos, under a Creative Commons license.

ABSTRACT

We present PortraitSketch, an interactive drawing system that helps novices create pleasing, recognizable face sketches without requiring prior artistic training. As the user traces over a source portrait photograph, PortraitSketch automatically adjusts the geometry and stroke parameters (thickness, opacity, etc.) to improve the aesthetic quality of the sketch. We present algorithms for adjusting both outlines and shading strokes based on important features of the underlying source image. In contrast to automatic stylization systems, PortraitSketch is designed to encourage a sense of ownership and accomplishment in the user. To this end, all adjustments are performed in real-time, and the user ends up directly drawing all strokes on the canvas. The findings from our user study suggest that users prefer drawing with some automatic assistance, thereby producing better drawings, and that assistance does not decrease the perceived level of involvement in the creative process.

Author Keywords

Sketching; Portraits; Novices; Creativity Support Tool

ACM Classification Keywords

H.5.2 HCI: User Interfaces

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or 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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI'14, April 26–May 1, 2014, Toronto, Canada.

Copyright © 2014 ACM ISBN/14/04...\$15.00.

DOI string from ACM form confirmation

INTRODUCTION

Drawing/sketching is arguably one of the most accessible art forms, requiring only a drawing tool, some paper, and the willingness to engage in a creative task. However, many people feel that they lack the ability to draw well and end up creating primitive, iconic sketches [8]. According to many art books, a high-level problem is that non-artists tend to draw “*what they know*,” instead of “*what they see*.” In other words, non-artists draw objects based on their knowledge of the object geometry whereas artists choose which lines to draw based on the pose and lighting of the scene they are trying to depict. Furthermore, non-artists often lack low-level technical skills that can affect drawing quality, such as the ability to produce accurate, clean outlines or shaded regions with even tone. Art books and courses can help novices improve their drawing ability, but most people do not have the time to attend classes and hone their skills through dedicated practice.

The goal of our work is to develop a digital drawing system for novices, which they can use to create better sketches without requiring formal artistic training. There are several important requirements for such a system. Since novices are the target audience, the system should be **easy to learn** and **robust** to input from users with poor drawing skills, while still producing good results. The system should also promote the users’ sense of **ownership** over their drawings. If a user feels he significantly contributed to the creation of a drawing that is of sufficient aesthetic quality, he might perceive the process as a rewarding experience of creative expression. Additionally, every user draws lines slightly differently, which results in a unique style. Ideally, the system should **preserve style**.

There clearly is tension between these requirements. A completely automatic system that generates drawings at the push of a button is easy to learn and robust to unskilled input, but does not preserve a user’s style or encourage ownership over the results. Conversely, a completely manual drawing system preserves style but is not robust. In our view, resolving this tension is the main challenge in designing a successful drawing system for novices. In essence, our goal is a system that assists with the low-level mechanics of drawing, such as ensuring that lines align well with the image, but where all high-level decisions are made by the user.

In this paper, we present PortraitSketch, an interactive drawing system that helps users create drawings of faces.

Faces have clear social importance but are very difficult to draw. In our recruiting survey (59 respondents), 85% said they would like to draw Portraits (vs. Landscape 70%, Abstract 61%, Still Life 56%, Other 3%). The main drawing interaction of PortraitSketch involves users tracing over a transparent version of a source portrait photograph. The system has three key features that address the requirements described above:

Automatic outline adjustment. PortraitSketch automatically adjusts the shape, position, thickness and opacity of outline strokes based on important visual features in the source image, such as strong gradients and facial landmarks. In this, we build on the work of Limpaecher et al. [16], while making a number of improvements.

Automatic shading adjustment. The system adjusts shading strokes to produce more even tones that match the shading variations in the source portrait.

Shading guidance. PortraitSketch provides an interactive shading visualization, suggesting where users might add additional shading strokes to better match the source image.

These features make the system robust to novice input by improving the accuracy and quality of the outline and shading strokes, and by helping users visualize where to add shading. At the same time, the fact that PortraitSketch allows users to add every stroke in the final image themselves keeps them highly involved in the creative process and gives them significant control over the style of the output. Figures 1 and 7 show results generated using our system, both with and without automatic assistance. Note that our system makes very subtle changes, but with a substantial effect on the quality of the resulting drawing. In general, the assisted drawings are more aesthetically pleasing and show a range of interesting stylistic variations.

To evaluate our system, we recruited 24 novices to create drawings using PortraitSketch, with different levels of assistance. We collected self-rating data about the drawing behavior, drawing results, and subjective ownership experienced across the assistance conditions. We also asked independent judges to evaluate the resulting drawings. Our findings suggest that users prefer drawing with some amount of assistance over no assistance, and that assistance does improve the quality of the drawings. There is no evidence that assis-

tance diminishes the perceived involvement of users in the creative task, and visual inspection of the drawings indicates that many unique stylistic qualities of individual users are preserved across all levels of assistance.

RELATED WORK

There exists extensive work on producing artistic renderings of portraits (see Zhao and Zhu [22] for an overview). However, these techniques are designed to work automatically, while we aim to provide an interactive drawing experience where the user actively engages in the creative process. Our work does share some technical details with recent work by Berger et al. [3], who adopt a data-driven method for generating face images in a range of different styles. Our face feature analysis and edge detection model is similar to theirs, and a measure related to their importance estimation of edges for level-of-detail abstraction is heuristically encoded in our parametric adjustments for facial outlines. Still, their high-level goal remains the automatic synthesis of images rather than interactive assistance. Furthermore, their model does not address shading, which is a key component of our system.

A related topic is sketch refinement or beautification [1, 2, 17]. These systems aggressively modify the user’s drawing to produce results that meet specific critiera and in the process significantly change the style of input strokes, often combining or removing them entirely. Moreover, beautification systems typically do not provide assistance during the drawing process. Our method also modifies strokes, but it does so in an interactive manner that is designed to both assist novice users and preserve their unique drawing style. We note that Zitnick [23] does present a real-time sketch beautification system that aims to preserve style, but the approach focuses on handwriting (or other repetitive symbols) rather than faces.

Some previous work on drawing has focused on teaching traditional skills. Dixon et al. [7] help users learn to draw faces by providing step-by-step instructions, analyzing a user’s performance, and giving feedback about inaccuracies. Cummings et al. [6] apply the same approach to eyes. While such systems help users improve their drawing ability, this process takes time. Our aim is to enable novices to create compelling results at their current skill level. Although users may eventually learn some transferable skills after creating drawings with PortraitSketch, this is not our primary goal. In addition, most teaching systems require manual curation or preparation of the learning content, which makes it difficult scale to new content (e.g., arbitrary source photographs). One exception is the work of Iarussi et al. [12], who analyze a source image and propose construction lines used in traditional drawing to guide the user and help them learn how to perceive and transfer shape and proportions to the page. While we also use image analysis to assist novice users as they draw, we adopt different techniques that are designed to work well with faces.

The most relevant previous efforts are those that focus specifically on real-time assistance for creative drawing tasks. Fernquist et al. [9] allow users to follow tutorials that are automatically generated from artists recorded performances and provide corrective feedback of the tutorial. However, this process still requires manual, expert curation. Lee et al. [14]

compare user strokes to edges in a database of images to suggest possible future lines using a blended-average shadow visualization. Their system is geared toward serendipitous exploration of the database content, whereas our focus is on assisting users in drawing a specific portrait. Flagg et al. [10] propose a multi-projector display to offer visual painting guidance for real-world painting. We would like to offer a simple solution without complex hardware requirements. The DrawAFriend system [16] for tracing portrait photographs is most closely related to PortraitSketch. However, rather than using image analysis and computer vision to assist the user, they rely on crowd-sourcing. While this approach works well for common, popular images (e.g. photos of celebrities), DrawAFriend does not help an individual user draw a specific personal photo. In addition, since the consensus for determining an ideal sketch in DrawAFriend is deduced from many novice peers, it is possible that common novice mistakes (e.g., drawing eyes as simplified almond shapes) may result in sub-optimal drawing corrections. Additionally, we use facial landmarks to ensure that any single outline stroke is only mapped to a single facial feature (e.g., eye or mouth). Finally, we also aim to help users create hatched shading regions. While Salisbury et al. [18] provide high-level tools for creating hatching in user-specified regions, our system allows users to participate more actively in the drawing process by directly drawing their own shading strokes.

SYSTEM

PortraitSketch is designed for ease-of-use. The user simply loads a facial portrait photograph, and starts tracing over the image with a drawing tool. The system is used with a Wacom stylus and tablet, though a mouse or touch interface could be used instead. The tracing is shown on a white background with the portrait faded, and opacity controlled by a slider.

Only two tools are available to the user: The *outline* tool, to depict the shapes of an image, and the *shading* tool, optionally used to apply different types of shading, such as hatching, cross-hatching, scumbling, or “fill.” Using two tools, instead of one, has several advantages. We find that users obtain better results when starting with at least a basic outline of a face before shading it, and using two tools encourages this workflow. Outlines and shading are often drawn in different styles (line thickness etc.) and the two tools can be conceptualized as presets for these styles. Finally, the system can give more targeted assistance if it knows the type of line being drawn.

To help users overcome their artistic and technical challenges, PortraitSketch offers several forms of assistance: (1) *Stroke beautification*, which makes strokes look better; (2) *Geometric corrections*, which improves shape depiction; (3) *Semantic corrections*, which de-emphasizes strokes that a skilled artist would be unlikely to draw; and (4) *Visualization*, which guides users in where to draw next.

To foster a user’s sense of ownership over the results, we adhere to the following guidelines when providing assistance: each stroke shown by the system is drawn by the user; no strokes are generated automatically. Adjustments are kept “small;” strokes may not be dramatically altered, and, even when altered, we aim to preserve the user’s style.

OUTLINE TOOL

The outline tool assists in drawing outlines and contours. The user draws strokes, represented initially as polylines, and the system applies several types of adjustments to the strokes. These adjustments are based on the typical problems with strokes drawn by novices. In particular, strokes may be uncertain and jittery, they may lack dynamics (skillful pressure variations of an artist’s pen), they may represent “incorrect” lines due to semantic bias, and even correct lines may be misplaced due to poor hand-eye-coordination. The latter issue can also arise for trained artists when using a low-fidelity input device, such as a touch-screen (“fat finger problem”).

Beautification

The raw polyline input for each stroke must be resampled and smoothed, to reduce data-size and jitter. To avoid discarding intentional curvature variations, smoothing is performed in a curvature-adapted manner, similar to Zitnick [23], in which we sequentially re-sample the input stroke based both on the amount of curvature and the distance of samples. The resampled points are fitted with cubic B-splines, which are rendered as thick triangle strips, with optional tapering or brush textures for increased realism. This approach is simple to implement, yet supports a variety of styles.

Geometric Adjustment: Outline

The geometric adjustment stage modifies stroke positions to better align with image contours, while still preserving the user’s stroke style. This helps when a user has difficulty following the shape of curves in the reference image.

Guidance Map and Face Part Labeling — Geometric adjustment requires the system to have a best guess as to where strokes should be drawn. Prior to the drawing process, we produce a binary Guidance Map, where black pixels represent suggested locations for strokes (Figure 2). We also identify facial landmark positions, corresponding to facial parts, such as eyes, nose, etc. Additionally, each black pixel in the Guidance Map is labeled with the closest facial part.

Previous findings have shown that many artist-drawn strokes can be explained in terms of image intensity gradients [4]. Motivated by this, we had two artists draw outlines and shading of several faces. We then evaluated various edge-detection filters before settling on the eXtended Difference-of-Gaussians (XDoG) [21] filter, which produces a binary image resembling a line drawing. As shown in Figure 2, the XDoG yields a reasonable approximation not only of the outlines an artist might draw, but also indicates shading regions well. The XDoG filter produces an image curvature (κ) value for each pixel, which we use to compute pixelwise correspondences, since pixels with similar curvatures are more likely to correspond.

Facial landmarks (Figure 2(f)) are located using an existing labeling algorithm [19] that identifies points on the face corresponding to the eyes, eyebrows, nose, mouth and contour. We also manually add eye iris and forehead landmarks, though the labeling algorithm could easily be trained to detect

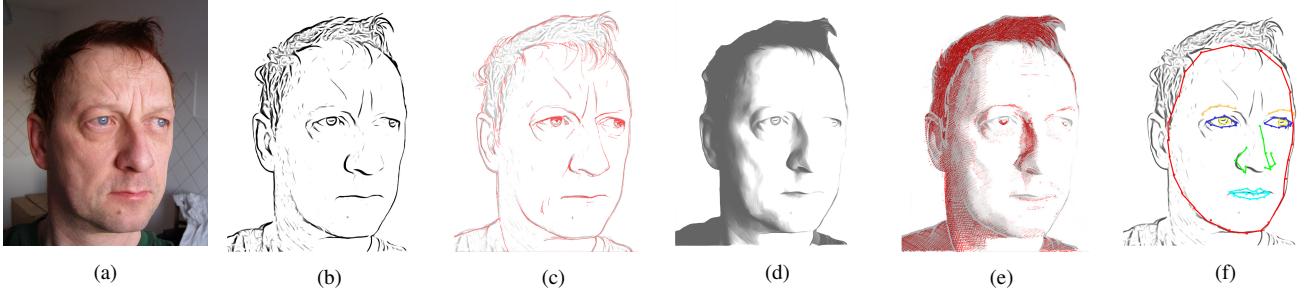


Figure 2: Guidance map. (a) Source portrait. (b) Output of original XDoG algorithm. (c) Overlay of outline sketch by a trained artist over the XDoG response. (d) Output of XDoG algorithm with settings for shading. (e) Overlay of artist’s shading sketch and XDoG response. (f) Facial landmarks. Each landmark is a point on the face. Landmarks are connected by line segments, for visualization (and curvature computation.). *Image credits: Erich Ferdinand, under a Creative Commons license.*

them automatically. The landmarks are connected by pre-determined polylines, and per-landmark curvatures (κ) are computed from the angles of the polyline.

We also precompute a part labeling of each pixel in the Guidance Map. Each black pixel in the Guidance Map is assigned the label of the nearest landmark in (x, y, κ) space. Non-face pixels are labeled as *background*.

Optimization — Given the input stroke and the precomputed maps, our goal is to correct a user’s input stroke toward the closest guidance feature, while retaining as much of the input stroke’s stylistic variations as possible. We formulate an energy function comprised of a Guidance Map energy, face landmark energy, and user input variation:

$$E = \alpha E_{guide} + \beta E_{land} + \gamma E_{var} \quad (1)$$

The weights α, β, γ adjust the contribution of each component. For our study, we experimentally choose their values to be 4, 2, and 1, respectively. The input stroke is described by a list of N points in 2D: $\mathbf{p}_{1:N} = [\mathbf{p}_1, \dots, \mathbf{p}_N]$, and the optimization produces a new list of points $\mathbf{q}_{1:N} = [\mathbf{q}_1, \dots, \mathbf{q}_N]$.

Prior to optimization, the algorithm assigns a face-part label for the stroke, to determine correspondence between the stroke and the Guidance Map. Assuming that users typically draw a stroke for one face feature at a time, we disambiguate conflicting labels of a stroke’s control points by majority voting. Namely, each control point is assigned the nearest label from the nearest landmark. Thus, unlike the DrawAFriend system [16], our system effectively prevents a single stroke from being aligned to multiple facial features (e.g., the eye and eyebrow). This gives N votes, and the majority label is assigned to the stroke. Ties are broken in favor of the label with the smallest average distance to nearest neighbors.

The objective terms are as follows. For alignment with *Guidance Map edges*, we define

$$E_{guide}(\mathbf{q}_{1:N}) = \sum_{i=1}^N \frac{\|\mathbf{q}_i - \mathbf{g}_i\|^2}{\|\mathbf{p}_i - \mathbf{g}_i\|} \quad (2)$$

where \mathbf{g}_i is the position of the closest Guidance Map pixel to \mathbf{p}_i with the same part label as the stroke. The denominator is used to downweight this term for distant pixels that are less likely to represent a correct guidance match.

Similarly, for *face landmark alignment*, we define

$$E_{land}(\mathbf{q}_{1:N}) = \sum_{i=1}^N \frac{\|\mathbf{q}_i - \mathbf{f}_i\|^2}{\|\mathbf{p}_i - \mathbf{f}_i\|} \quad (3)$$

where \mathbf{f}_i is the facial feature closest to \mathbf{p}_i with the same label as the stroke. To preserve *stylistic input variations*, we minimize changes to the curve tangents, similar to [16]:

$$E_{var}(\mathbf{q}_{1:N}) = \sum_{i=2}^n \|(\mathbf{q}_i - \mathbf{q}_{i-1}) - (\mathbf{p}_i - \mathbf{p}_{i-1})\|^2 \quad (4)$$

Subfigures 3(b,c,d) show an example of an outline sketch before and after geometric correction. The optimization in Equation 1 is solved by a least squares solver.

We also provide a level-of-assistance parameter $\lambda \in [0 \dots 1]$. Given the optimized stroke \mathbf{q}^* , the final output stroke is a linear interpolation of the input stroke and the optimized stroke:

$$\mathbf{q}_i^{\text{out}} = \lambda \mathbf{q}_i^* + (1 - \lambda) \mathbf{p}_i \quad (5)$$

Semantic Correction: Outline

Even when tracing, a novice may draw lines that a trained artist would not. This is typically caused by a semantic bias, where knowledge of object properties bias a novice to depict lines that are either weak or merely imagined. To mitigate this problem, our system emphasizes lines that are deemed to be “correctly” drawn, whereas de-emphasizing lines that are drawn “incorrectly”.

For each point on an outline stroke, the algorithm computes the distance from the point to the nearest Guidance Map pixel. This distance is smoothed along the stroke, and then used to adjust the width and opacity of the stroke. Emphasized strokes are wider and/or more opaque while de-emphasis uses the opposite mechanism. Figure 3(e) shows the sketch result after semantic correction.

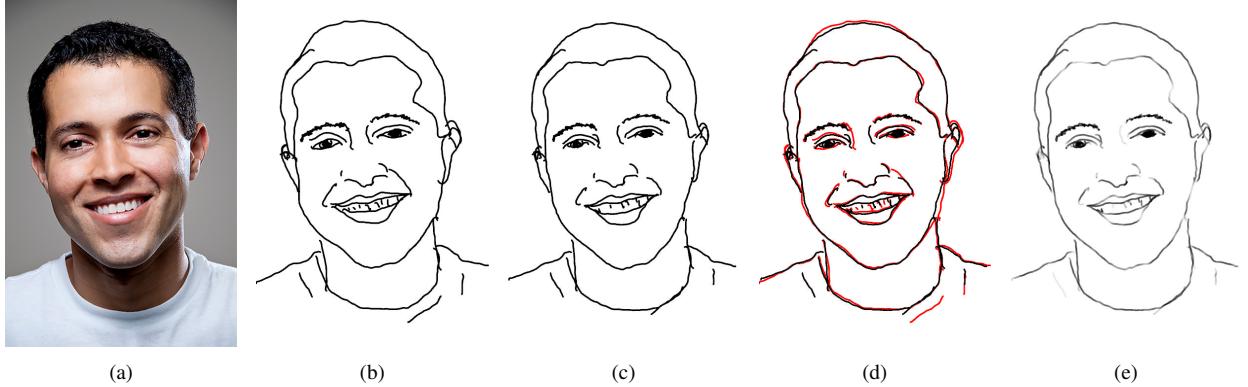


Figure 3: Outline Adjustment Comparison. (a) Source portrait. (b) No assistance. (c) Geometric correction. (d) Overlay of no assistance (red) and geometric correction (black). (e) Semantic correction. Note the geometric correction of the lower eyelid and semantic correction of the teeth and right bridge of the nose. Here, we use opacity for stroke emphasis.

Freehand Strokes

Outline strokes are left unadjusted in 2 situations, as follows.

First, input strokes too far from guidance features are left unadjusted. This case is detecting by checking if

$$\frac{1}{N} \sum_{i=1}^N \max (\|\mathbf{p}_i - \mathbf{g}_i\|, \|\mathbf{p}_i - \mathbf{f}_i\|) \quad (6)$$

passes above a preset threshold. In this case, λ is set to zero.

Second, strokes drawn in “fill” regions are uncorrected. We define *fill* regions as dense XDoG regions whose response is close to the shading response (see below). For example, hair regions often exhibit too many edges to meaningfully align user strokes. We detect such regions by comparison with shading regions and locally set $\lambda = 0$.

SHADING TOOL

Shading gives visual cues about lighting, 3D shape of a depicted object, and can indicate material properties and texture such as hair. Shading is a more advanced skill than outline drawing, and is particularly challenging and tedious for novices. We now describe a novel tool for assisting in the drawing of shading strokes.

Shading Region Detection

The shading tool takes as input a Shading Map, indicating parts of the image requiring shading. This is used in the interactive system both for shading visualization and assistance.

The Shading Map is generated by first defining the total face area (using face feature data) minus eyes and mouth regions, as these should not be shaded. The color histogram of the remaining face pixels typically exhibits two peaks, intuitively corresponding to skin color and shadow. These regions are clustered by fitting a Gaussian Mixture Model with two clusters as $\mathcal{N}(\mu_1, \sigma_1)$ and $\mathcal{N}(\mu_2, \sigma_2)$. XDoG is run twice on the image with two different threshold parameter settings: $\epsilon_1 = \mu_1 - 3\sigma_1$ and $\epsilon_2 = \mu_2 - 3\sigma_2$. The difference of the filter responses defines the shading region. We also run the Mean Shift algorithm [5] to extract shadable skin regions outside the face (e.g., neck, shoulders). The final shading region is shown in Figure 4(a).

Shading Guidance Visualization

The suggested shading regions are overlaid on the reference image in transparent red with opacity corresponding to shading intensity. An optional feedback mode fades the shading guidance according to the amount of shading strokes already applied by the user.

Shading Strokes Correction

As soon as they are drawn, shading strokes are re-sampled and smoothed in the same way outlines are. Then, the system adjusts the spacing of shading strokes to distribute them more evenly to produce an even tone [18].

Our hatching stroke correction uses an energy minimization inspired by the streamline placement method of Turk and Banks [20]. When the user places a new shading stroke, we optimize *all* strokes in a local neighborhood of the new stroke, including the input stroke. Let k index this set of K strokes, so that $\mathbf{P}^{(k)} = [\mathbf{p}_1^{(k)}, \dots, \mathbf{p}_{N_k}^{(k)}]$ is the k -th stroke. Our goal is to produce updated positions $\mathbf{Q}^{(k)}$. We minimize an energy over all control points of all neighboring strokes:

$$E(\mathbf{Q}^{(1:K)}) = E_{tone}(\mathbf{Q}^{(1:K)}) + E_{var}(\mathbf{Q}^{(1:K)}) \quad (7)$$

The first term E_{tone} , encourages strokes to be evenly distributed while matching the target tone. Let S be a binary image of all strokes, and $G * S$ be a Gaussian-blurred version of this image, and let the target (suggested) shading map from preprocessing be T . The energy is:

$$E_{tone}(\mathbf{Q}^{(1:K)}) = \|G * S(\mathbf{Q}^{(1:K)}) - T\|^2 \quad (8)$$

This energy is summed over all pixels in the image, though in implementation only pixels nearby the control points being optimized need to be considered.

As in (Equation 4), E_{var} preserves input stroke tangents:

$$E_{var} = \sum_{k=1}^K \sum_{j=2}^{N_k} w_j \|(\mathbf{q}_j^{(k)} - \mathbf{q}_{j-1}^{(k)}) - (\mathbf{p}_j^{(k)} - \mathbf{p}_{j-1}^{(k)})\|^2 \quad (9)$$

where w downweights strokes that are perpendicular to the input stroke to allow for cross-hatching. In particular, let $\mathbf{d}^{(k)} = \mathbf{q}_{N_k}^{(k)} - \mathbf{q}_1^{(k)}$ be the average direction of a stroke. We

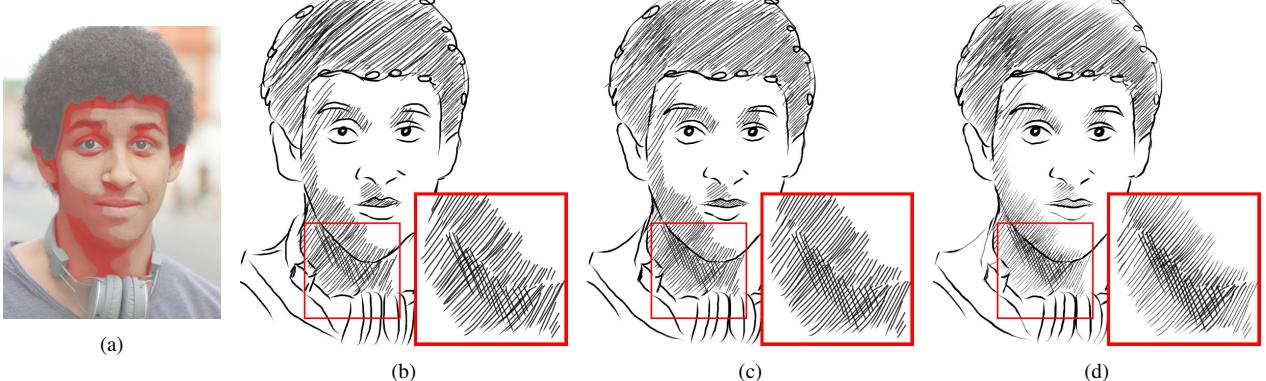


Figure 4: Shading Adjustment Comparison. (a) Source portrait with visual shading guidance. (b) No assistance. (c) Geometric correction. (d) Semantic correction.

set w inversely proportional to $\mathbf{d}^{(k)} \cdot \mathbf{d}^{(\text{in})}$. When the angle between these directions is greater than $> 30^\circ$, the weight is set to zero, which typically corresponds to cross-hatches.

The level of assistance is adjusted by the λ parameter as in Eq. 5. We solve the above optimization as follows. The algorithm computes the blur map $G * S$. Then, each control point is separately updated by a few gradient descent steps, and the process repeats. Image gradients are computed by finite differences. The algorithm runs about 10-30 iterations to convergence, requiring about 1-2 ms per iteration.

Multi-stroke Support

Fine hatching can be tedious and difficult. To make it easier, the hatching tool draws multiple parallel stroke for each user stroke (similar to [13]). The number of strokes reduces locally in areas with little shading to allow for finer shading control. The multi-stroke feature naturally creates evenly spaced lines, thereby enhancing overall stroke distribution.

Semantic Correction: Shading

Semantic adjustment is used to de-emphasize strokes in regions that need no shading. We use the previous shading map T and the stroke map S as guidance sources. We first lower the dynamic range of T to prevent excessive variations in shading appearance. For each new shading stroke \mathbf{Q} , the shading error is computed at each control point as:

$$\epsilon_{\text{shading}}(\mathbf{q}_i) = k \cdot \lambda(T(\mathbf{q}_i) - \min(G * S(x_i), \delta_s)) \quad (10)$$

where k is a constant and the level-of-assistance λ is used to modulate this correction. The shading density parameter δ_s can be adjusted via the UI. As in the outline tool, this value is used to de-emphasize strokes by making them smaller and more transparent when the value of ϵ is large. As more shading strokes are added to a region, new strokes become thinner and lighter, to prevent the region from darkening too much.

Figure 4 shows a shading sketch before and after adding assistance. In the zoomed-in window, shading strokes are placed in a more regular pattern compared to the original input. After the semantic adjustment, the shading also has a better interpretation of the face's structure and luminance variations.

EVALUATION

To evaluate how well PortraitSketch achieves our stated goals, we consider the following three questions:

1. Do novices prefer the assisted behavior of PortraitSketch over unassisted drawing?
2. Do novices create better drawings with assistance?
3. Do novices feel ownership over their assisted drawings?

The remainder of this section describes the methodology of the user study we designed to help answer these questions.

Methodology

Our evaluation consists of two phases. First, we asked novice participants to create drawings using four versions of PortraitSketch with different levels of assistance. After completing the drawings, subjects answered several questions about the drawing behavior, drawing results and their perceived level of involvement in the creative process across the four conditions. In the second phase, we asked independent judges to evaluate the novice drawings.

Novice Drawing Study

For our novice drawing study, we recruited 24 participants who self-rated their drawing skills at 2 or lower (on a scale from 1–Novice to 5–Expert), who draw infrequently (monthly or less), and are interested in drawing more. To evaluate how well our system works for different drawing styles, we also asked potential participants to rate their level of interest in creating drawings in one of three target styles (Figure 5). We picked styles that resemble many example portrait sketches that we found online and have visual characteristics that are distinct from one another (e.g., width, length

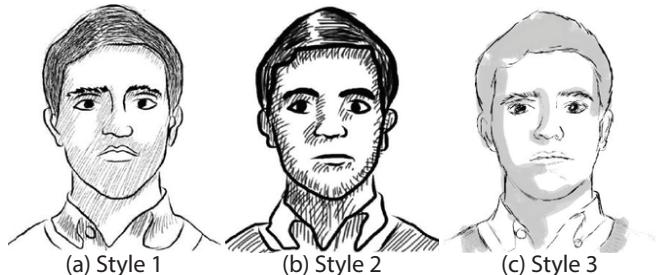


Figure 5: Target styles for user study.

and “cleanness” of outlines, type of shading, etc.) Based on their responses, we assigned participants to draw in one of the three target styles such that each style had eight participants and all subjects were assigned their (co-)favourite style.

For the study itself, we used a within-subjects design, where each subject created drawings of four different source images using four different levels of assistance: *None* ($\lambda = 0$), *Low* ($\lambda = \frac{1}{3}$), *Med* ($\lambda = \frac{2}{3}$), and *High* ($\lambda = 1$). We modified the UI to show an example sketch in the target style (as a reference image) to the right of the drawing canvas (please refer to the submission video). For the source images, we selected an equal number of male and female portrait photos with (approximately) fronto-parallel viewing angles. Three of our source images are from the DrawAFriend dataset of celebrity photos [16]. For each target style, we counterbalanced the order of assistance conditions and the pairings between conditions and source images based on a Greco-Latin Square design that accounts for both immediate ordering effects and condition-stimulus pairing effects [15].

At the beginning of each study session, we played a brief video introducing the key features of the PortraitSketch UI. We then allowed users to practice with the unassisted version of PortraitSketch for up to ten minutes to get comfortable with the setup (a Wacom tablet and stylus in front of a 24" display) and to clarify any questions about the UI. After the practice session, we asked subjects to create four drawings. We explained that some system parameters affecting drawing behavior would change between each drawing, but we deliberately said nothing about the nature of the changes. This was done to alert subjects to behavior changes without biasing their experience. For each task, we asked users to do their best to draw the given source image in the target style in eight minutes or less.

After completing the tasks, subjects filled out an exit questionnaire. To determine whether users could identify the different types of drawing assistance, we asked subjects to explain what differences they noticed (if any) about the drawing behavior. We also asked subjects to rate how much they liked the drawing behaviors and their actual drawing results across the four conditions on a scale from 1 (strongly dislike) to 5 (strongly like) and to explain these ratings by describing what they liked and disliked about the behavior and results. Finally, to assess ownership, we asked users to rate on a scale from 1 (strongly disagree) to 5 (strongly agree) the following two statements: “I created the image by myself,” and “The system helped me with creating the image.”

Independent Judges

To gain more insight into the quality of the assisted versus unassisted drawings, we asked independent judges to evaluate the drawings. We recruited two experts, who are both professional designers with extensive formal training and experience in drawing. We also recruited seven non-artists to act as peer judges. We asked the experts to give absolute rankings for each set of four drawings created by the same user on a scale from 1 (Worst) to 4 (Best) based on two different criteria: 1) the artistic quality of the drawings and 2) the artistic quality of the drawings *as a likeness of the subject of the por-*

trait. While we expected these two rankings to be correlated, we felt there might be some cases where one drawing was artistically superior but a worse likeness and vice versa. To help keep the two rankings independent, we first asked the judges to rank all 24 sets of images using the first criterion, before adding the source images and asking the judges to rank all 24 sets of images again based on the second criterion.

The ranking tasks described above require judges to evaluate drawings from four different source images. For more direct comparisons, we took each *None* (no assistance) drawing created by a novice participant and generated an auto adjusted version *AutoMed* by playing back the recorded strokes in PortraitSketch with the *Med* assistance settings. We chose this assistance level because it typically produces noticeable differences, and as reported below, the *Med* condition received the highest mean drawing behavior rating. These auto adjusted drawings represent a worst-case version of the assisted output from PortraitSketch since users tends to adjust their drawing behavior based on visual feedback from the system during real drawing sessions. After generating the assisted images, we asked both our expert and peer judges to select the better result from each pair of *None* and *AutoMed* drawings. (We added peer judges for this evaluation as it required less expertise to compare.)

For all ranking tasks, we randomized the order in which the image quadruples/pairs were presented, and for each individual ranking task, we also randomized the layout of the drawings, which were arranged in a horizontal row.

RESULTS

To summarize the data from our study, we present boxplots that show the median and interquartile ranges for each set of ratings/rankings, with upper and lower quartiles computed using the method of Freund and Perles [11]. For brevity, we aggregate ratings across the three target styles.

Do novices enjoy the assisted drawing behavior?

Figure 6a summarizes the drawing behavior ratings for all 24 participants. Both the *Low* and *Med* conditions are rated more favorably than the *None* condition. Interestingly, the *High* condition received very similar ratings to the *None* condition. Looking more closely at the explanations of these ratings from the exit questionnaire, the relative lack of enthusiasm for *High* was likely due to the perception that the automated assistance was overly aggressive in this condition, which several participants noted. This highlights the trade-off between user control and assistance; even when the *High* setting produced the best results, some users were frustrated by the system not following their input.

Do novices create better drawings with assistance?

Overall, the drawing result ratings increase with increasing levels of assistance (Figure 6b), which suggests that participants generally preferred the drawings they created with more assistance. The rankings from expert judges also show a preference for the assisted drawings over the *None* condition (Figures 6e–f). Finally, the comparisons between the *None* and *AutoMed* (auto adjusted) image pairs produced the clearest results; judges preferred the assisted drawings 70% of the

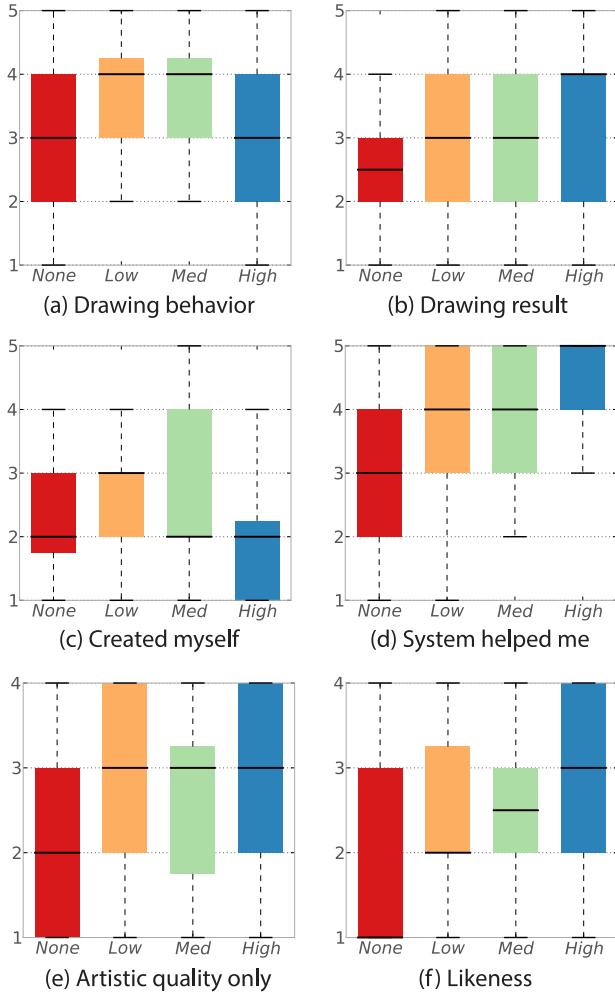


Figure 6: Results from user study. Boxplots a-d show aggregated self-rating results from our exit questionnaire (1: strongly dislike ... 5: strongly like). Boxplots e-f show expert ranking results (1: Worst ... 4: Best). Overall, the results suggest that users prefer drawing with some assistance over no assistance (a) and that assistance results in better drawings (b, e, f). The responses also suggest that while users clearly notice the system helping more at higher assistance levels (d), this may not impact their feelings of involvement in the creative process (c). The thick black line in each colored box indicates the median.

time, with only five of the 24 *None* drawings receiving more than four out of nine possible votes. In general, both the outlines and shading contribute to the improved quality of the assisted drawings. As shown in Figure 7, more accurate outlines tend to make the sketch more similar to the source image, and more even shading increases the technical quality of the result.

Do novices feel ownership over assisted drawings?

While this is a difficult question to answer conclusively, the ratings for the statements “*I created the image by myself*” and “*The system helped me with creating the image*” pro-

vide some insight. For the first statement, there is no obvious trend across the four conditions. *None* and *Low* have a similar distribution of ratings, with *Med* rated somewhat higher and *High* somewhat lower (Figure 6c). However, the ratings for the second statement clearly increase with increasing assistance (Figure 6d). Thus, while participants notice the increasing levels of assistance, this does not seem to diminish their sense of involvement in the creative process, at least up until the *Med* condition. We consider this a desirable result.

Moreover, the ability of the system to preserve characteristics of individual users’ drawing style may also contribute to a sense of ownership. While we did not evaluate style preservation explicitly in our study, almost all of the drawings from different users exhibit clear stylistic differences (even the drawings in the same target style). For example, Figure 8 shows four sets of drawings in two different target styles with various levels of assistance. In the top row, subject 129 uses very loose, messy outlines and shading, while subject 136 has a much tighter, more controlled style with lots of structured, parallel shading lines. In the bottom row, subject 144 uses fairly clean outlines and focuses on details around the eyes, while subject 138 draws with long, wavy outlines and relatively little detail, which gives the drawings a more abstract look. These types of differences suggest that, even with assistance, PortraitSketch users are still able to create drawings in their own unique styles.

Free-form responses

We asked users to describe the *drawing behavior* changes they noticed. Only 2 users noticed no difference and most users correctly identified *assistance* as the behavior change. We analyzed the given descriptions and grouped them into three connotation categories: Positive (e.g. help, assistance), Neutral (e.g. snapping, correction), and Negative (e.g. restrictive). If a description used words from multiple categories, we counted the less favorable one. 5 users made positive mention, 8 users were neutral, and 3 users made negative comments about the assistance. Several users noticed behavior changes that were not real, and might reflect a subjective learning experience. This evaluation mirrors the quantitative data in that most users seemed comfortable with the given assistance, or even appreciated it.

For *positive feedback*, 8 users specifically mentioned liking the *assistance*; 5 users thought that *tracing* made the task fun and manageable; 6 users mentioned the *shading visualization*; and 9 users commented on the usefulness of adjusting the *opacity* of the reference image.

Suggested *system improvements* include: An eraser tool to delete arbitrary lines (12 users); manual adjustment of outline and shading thickness (11 users); additional shading styles (3 users); and ability to adjust assistance (3 users).

Interesting feedback from a *general comments* question include: (146) “[...] I found when I tried to use the shading system I was more focused on the guide than the image itself which had some bad results.”; (132) “i can create wonderful sketch with this software”; (120) “Fun factor [...] I think that will help in freehand drawing”; (110) “It was surpris-

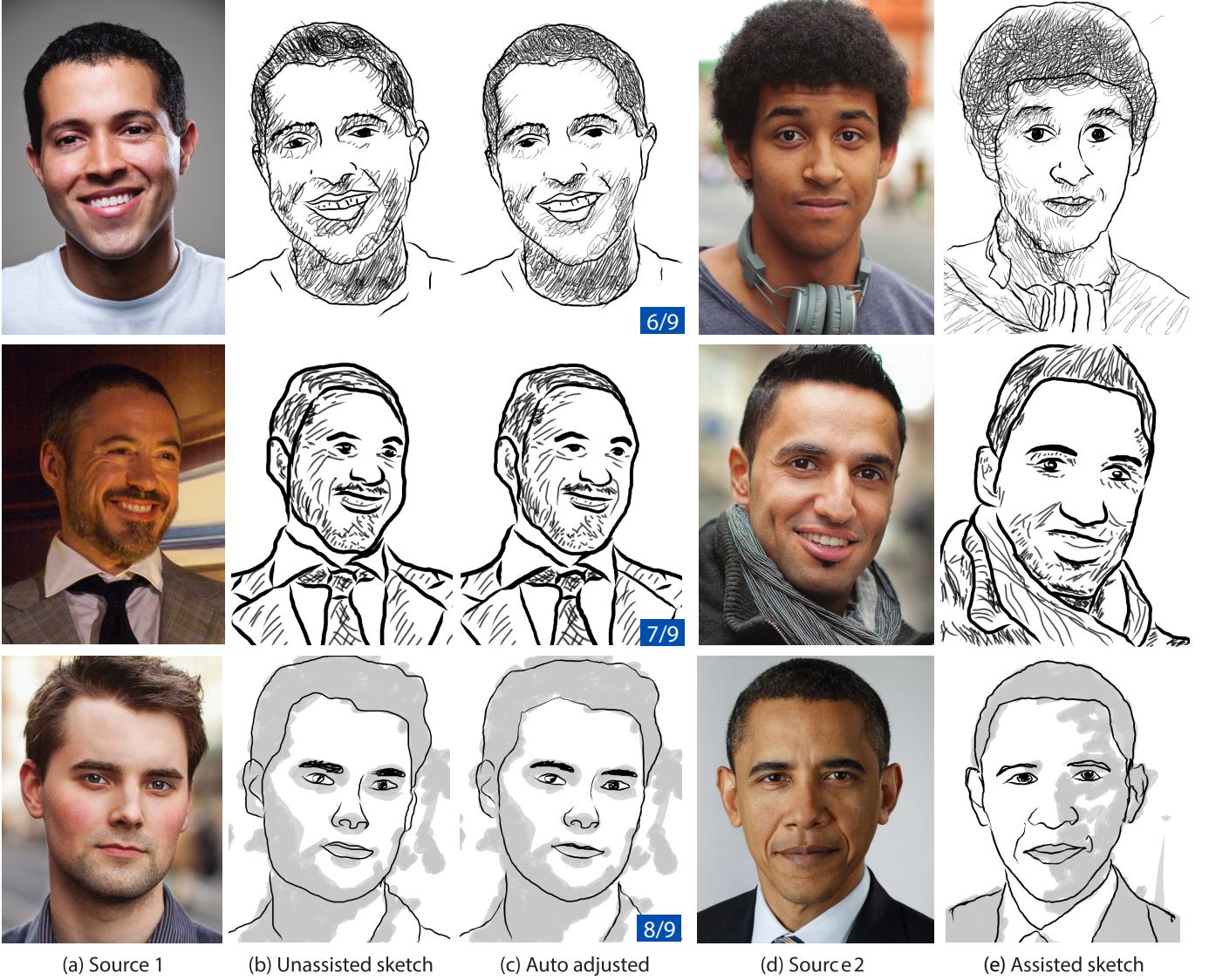


Figure 7: Some results from user study. We show unassisted (b), auto adjusted (c), and assisted (e) drawings by three different novice users (organized by row) in our study. PortraitSketch’s automatic adjustments produce more accurate outlines (compare the face contours and eyes in columns (b) and (c)) and more even shading (e.g., the hair and neck in top row drawings). While some differences are subtle, several small adjustments often produce noticeable improvements in the overall drawing. The blue numbers in column (c) indicate the fraction of independent judges who preferred the auto adjusted images over the unassisted drawings in column (b). All three users produced the assisted drawing (e) *before* the unassisted one (b) in their study sessions. *Image credits:* (a) top to bottom: Levy Carneiro Jr, Edgar Meritano and chris zerbes. (d) top to bottom: chris zerbes, chris zerbes and Pete Souza. All are under a Creative Commons license.

ingly fast and easy to use and understand. I also feel like I could quickly draw something, if tasked”; (166) “I lack creativity, so I like that there was a ‘tracing’ aspect to help me draw”; (127) “Overall, I was very impressed when I saw the outcome. Can’t wait to try out more drawings, it’s addictive :-); (138) “[An exercise] that gave me hope that maybe I actually could produce something worth looking at! :); (121) “It gave me hope that I could actually learn to draw”.

DISCUSSION & FUTURE WORK

Our system assists novices in creating sketches by making subtle adjustments to technical qualities of their drawings. Two situations may lead to sub-optimal assistance conditions: 1) Facial feature detection and XDoG edge extraction may fail (due to low contrast, facial occlusion, etc.) 2) The user creates drawings with little correspondence to the source image. Our study gives preliminary evidence that this kind of assistance enhances the resulting drawings, enhances user’s enjoyment and pride in their work, while maintaining the sense of ownership they would feel in a conventional interface. The

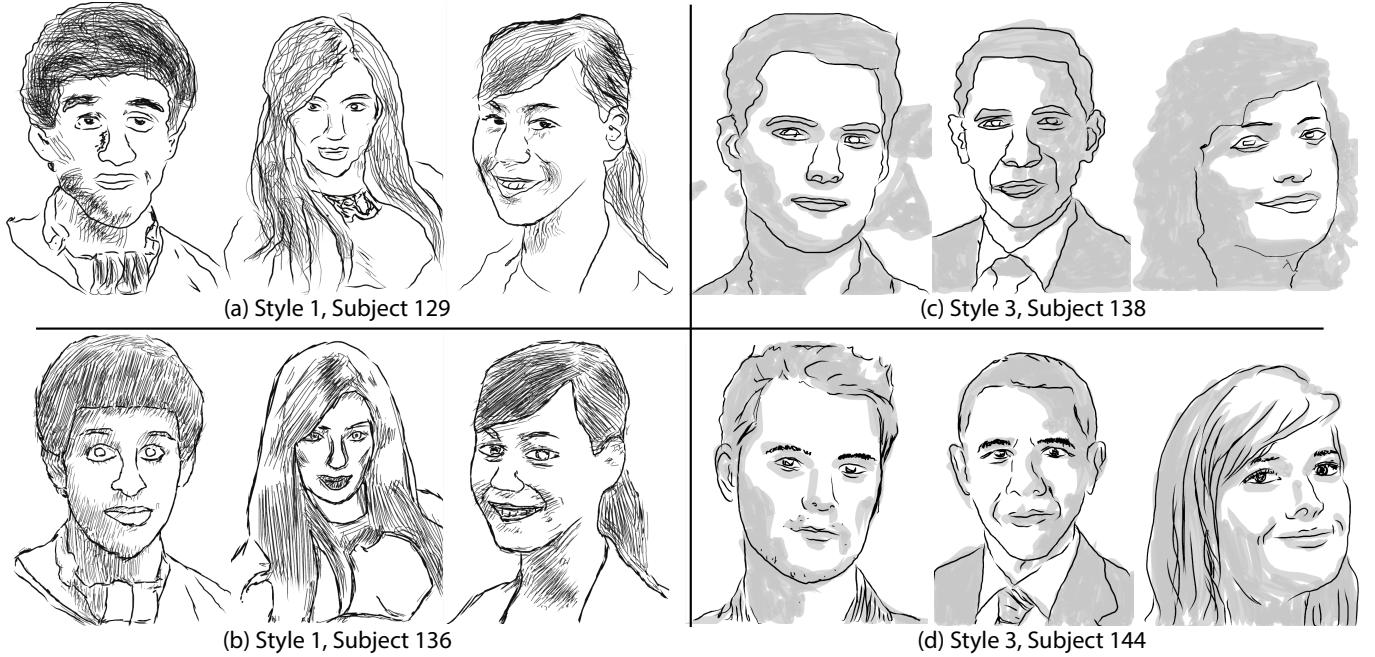


Figure 8: Preserving stylistic differences. Each set of three drawings was created by one of our study participants using three different levels of assistance. The drawings by each user exhibit unique characteristics that distinguish them from the drawings by other users. These stylistic differences are present even when comparing drawings in the same target style; for example, (a–b) are in target style 1, and (c–d) are in target style 3.

behavior and drawing result ratings suggest that there may be a sweet spot around the *Med* level of assistance, which provides enough help to noticeably improve drawing quality but does not modify user outlines and shading too aggressively. We chose not to explain the nature of the assistance to participants, but we observed that some users who figured out our system produced better results.

We observed several potential sources of noise in our study, including unfamiliarity with the Wacom tablet, learning effects, and variation between the different types of source images. Although we counterbalanced for each of these effects, they did reduce the observed effect sizes.

Many novices are interested in creative expression but lack the confidence and skills to partake in art projects. Indeed, several study participants initially apologized for their lack of drawing skills, but, at the end, were thrilled with their results. One user even captured her drawings with her phone to immediately share with her friends. We think that additional guidance for difficult drawing elements, such as hair or clothing could be useful. While we focus on novices, even professional artists, who quickly have to produce a rough sketch, might benefit from image-based drawing assistance. We see the encouraging results from our user study as evidence that building creative assistance tools for novices is an important goal and we intend to apply our assistance approach to other artistic domains in the future.

REFERENCES

1. Arvo, J., and Novins, K. Fluid Sketches: Continuous Recognition and Morphing of Simple Hand-drawn Shapes. In *Proc. UIST* (2000), 73–80.
2. Barla, P., Thollot, J., and Sillion, F. Geometric Clustering for Line Drawing Simplification. In *Proc. EGSR* (2005).
3. Berger, I., Shamir, A., Mahler, M., Carter, E., and Hodgins, J. Style and Abstraction in Portrait Sketching. *ACM Trans. Graph.* 32, 4 (July 2013).
4. Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H. S., Finkelstein, A., Funkhouser, T., and Rusinkiewicz, S. Where Do People Draw Lines? *ACM Trans. Graph.* 27, 3 (Aug. 2008).
5. Comaniciu, D., and Meer, P. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Trans. PAMI* 24, 5 (May 2002), 603–619.
6. Cummings, D., Vides, F., and Hammond, T. I Don’t Believe My Eyes!: Geometric Sketch Recognition for a Computer Art Tutorial. In *Proc. SBIM* (2012).
7. Dixon, D., Prasad, M., and Hammond, T. iCanDraw: Using Sketch Recognition and Corrective Feedback to Assist a User in Drawing Human Faces. In *Proc. CHI* (2010).
8. Eitz, M., Hays, J., and Alexa, M. How Do Humans Sketch Objects? *ACM Trans. Graph.* 31, 4 (July 2012).

9. Fernquist, J., Grossman, T., and Fitzmaurice, G. Sketch-sketch Revolution: An Engaging Tutorial System for Guided Sketching and Application Learning. In *Proc. UIST* (2011).
10. Flagg, M., and Rehg, J. M. Projector-guided painting. In *Proc. UIST* (2006).
11. Freund, J., and Perles, B. A new look at quartiles of ungrouped data. *American Stat* 41 (1987), 200–203.
12. Iarussi, E., Bousseau, A., and Tsandilas, T. The Drawing Assistant: Automated Drawing Guidance and Feedback from Photographs. In *Proc. UIST* (2013).
13. Kazi, R. H., Igarashi, T., Zhao, S., and Davis, R. Vignette: Interactive texture design and manipulation with freeform gestures for pen-and-ink illustration. In *Proc. CHI* (2012).
14. Lee, Y. J., Zitnick, C. L., and Cohen, M. F. ShadowDraw: Real-time User Guidance for Freehand Drawing. *ACM Trans. Graph.* 30, 4 (July 2011).
15. Lewis, J. R. Pairs of latin squares to counterbalance sequential effects and pairing of conditions and stimuli. In *Proc. Human Factors and Ergonomics* (1989).
16. Limpaecher, A., Feltman, N., Treuille, A., and Cohen, M. Real-time Drawing Assistance Through Crowdsourcing. *ACM Trans. Graph.* 32, 4 (July 2013).
17. Pusch, R., Samavati, F., Nasri, A., and Wyvill, B. Improving the sketch-based interface. *The Visual Computer* 23, 9-11 (2007), 955–962.
18. Salisbury, M. P., Anderson, S. E., Barzel, R., and Salesin, D. H. Interactive pen-and-ink illustration. In *Proc. SIGGRAPH* (1994).
19. Saragih, J. M., Lucey, S., and Cohn, J. F. Deformable Model Fitting by Regularized Landmark Mean-Shift. *Int. J. Comput. Vision* 91, 2 (Jan. 2011), 200–215.
20. Turk, G., and Banks, D. Image-guided Streamline Placement. In *Proc. SIGGRAPH* (1996), 453–460.
21. Winnemöller, H., Kyprianidis, J. E., and Olsen, S. C. XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization . *Computers & Graphics* 36, 6 (2012), 740–753.
22. Zhao, M., and Zhu, S.-C. Artistic Rendering of Portraits. In *Image and Video-Based Artistic Stylisation*, P. Rosin and J. Collomosse, Eds. Springer, London, 2013.
23. Zitnick, C. L. Handwriting Beautification Using Token Means. *ACM Trans. Graph.* 32, 4 (July 2013).