iCanDraw? – Using Sketch Recognition and Corrective Feedback to Assist a User in Drawing Human Faces

Daniel Dixon, Manoj Prasad, and Tracy Hammond
Sketch Recognition Lab - Department of Computer Science and Engineering
Texas A&M University - College Station, TX 77843-3112, USA
{dixondm, manoj797, hammond}@tamu.edu

ABSTRACT
When asked to draw, many people are hesitant because they consider themselves unable to draw well. This paper describes the first system for a computer to provide direction and feedback for assisting a user to draw a human face as accurately as possible from an image. Face recognition is first used to model the features of a human face in an image, which the user wishes to replicate. Novel sketch recognition algorithms were developed to use the information provided by the face recognition to evaluate the hand-drawn face. Two design iterations and user studies led to nine design principles for providing such instruction, presenting reference media, giving corrective feedback, and receiving actions from the user. The result is a proof-of-concept application that can guide a person through step-by-step instruction and generated feedback toward producing his/her own sketch of a human face in a reference image.

Author Keywords
Sketch recognition, pen-input computing, assistive and corrective feedback, computer-aided instruction

ACM Classification Keywords
H.5.2 User Interfaces : Interaction Styles

General Terms
Design, Human Factors

INTRODUCTION
Many persons of all ages have adopted the idea that they are unable to draw well, and can be quick to confess this if asked to put pen to paper [16,24]. However, the ability to sketch a freeform drawing of an object is useful for conveying an idea, creating artwork, describing an event, expanding one's perception of the world, or any other of a number of possibilities. Much of the ability to draw comes from how one sees and processes the object being drawn. Joshua Bienko, a professional artist and art professor states, "For me, it is integral that students begin to identify marks that have not been observed accurately. Identifying inaccuracies is the first step toward rendering something realistically." An experienced drawer can see the contours, spaces, and relationships of an object, while the novice drawer produces more simplified representations of the object's features. A classic example is how a novice sketcher will incorrectly draw eyes too high on the head, rather than drawing them in the vertical center of the head where they reside, because the perceptually important features on a face occur on its lower half. "Beginning drawers draw what they KNOW" Bienko says. "I go to great depths in my courses to erase what artists know so that we can develop what we see." Helping the novice draw what s/he sees over what s/he perceives makes drawing assistance a perfect domain for sketch recognition.

Using the human form as a starting domain, this paper presents the first application to use sketch recognition to assist the user in creating a rendition of a human face with the intent of improving that person's ability to draw. The reference image is first processed using face recognition to obtain information about the location and size of the facial features. Each new step helps the user perceive what to draw through written instruction and corresponding adjustments to the reference image. As the user draws for each step, sketch recognition interprets the soon-to-be artist's freeform strokes. If the user requests assistance and/or draws something inaccurately, constructive feedback guides the user with modifications that can be made. Sketch recognition is the automated understanding by a computer of the intent of the user who is drawing. Previously, sketch recognition has focused on less subjective domains, and the task of drawing the human form poses many challenges for sketch recognition. After two design iterations and user

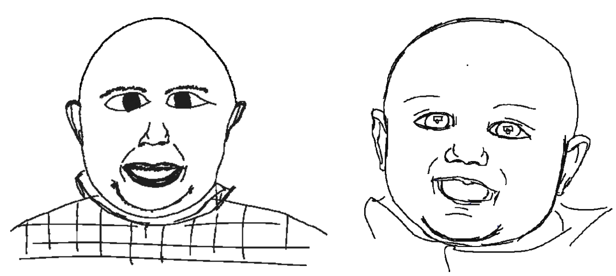


Figure 1 – Contrasting example of two participants perceiving and producing two very different drawings from the same reference image (see image A in Appendix).

studies, nine design principles were developed to determine what to recognize within a drawing, how to best provide helpful assistance, how to correct drawing issues, and how to interact with the user.

THE ART OF DRAWING

Drawing is considered a skill that is learnable at any age and not an innate ability or extraordinary coordination [8]. Every person is able to establish a basic competency. Without the defunct misconceptions that drawing is reserved for only the gifted, providing assistance in drawing within a computer application is not out of the question.

L-Mode vs. R-Mode

One popular theory for how humans visually process a scene involves dividing tasks between the right-brain and left-brain [7] or “R-mode” versus “L-mode” [8]. It states that the two halves of the human mind respond to stimulus differently and that a person is generally predisposed to respond using one half over the other. Those with a perceived inability to draw are responding using their L-mode, when the R-mode is best equipped to process visual data [8]. Aiding the mind in switching to the R-mode is the first step in preparing oneself to draw. As Edwards describes, the R-mode is where “we understand metaphors, we dream, [and] we create new combinations of ideas,” while the L-mode “analyzes, abstracts, counts, [and] plans step-by-step procedures”

Figure 1 shows sketches by two participants working from the same image. The left sketch is a byproduct of the L-mode and marked by symbolic representations for features. For example, an eye is drawn as a basic set of two curves and a black dot; this neither reflects a realistic eye nor the one found in the reference. In contrast, the perception of the R-mode processes the facial features as they are seen at the time and without the need of symbols or words, as in the drawing on the right of Figure 1, resulting in a more realistic representation. To draw realistically, the user must learn to “draw what is seen, not what is perceived.”

While, L-mode and R-mode are considered abstractions for how the brain actually works [5], this analogy works well in art instruction [8]. This is why the authors have chosen to use the terms “L-mode” and “R-mode” as analogical terms rather than the terms “left-brain” and “right-brain”, which claim localization in the brain.

Approaches to Drawing Instruction

Among the approaches for teaching drawing, there is no universal approach [3,4,6,8,10,13,14,16,21]. A teacher may choose not to show an example of what to draw, but rather openly define the task to encourage free-form drawing [3]. However, a book or tutorial can guide the student quite well in a step-by-step manner toward a well-defined product. Having students in a class imitate original works may be discouraged [4], though others feel that pupils can initially learn much by reproducing the masters [8, 14]. A distinction exists between encouraging creativity in a student and helping the beginner achieve a level of realism in his/her sketches. This focuses on the latter. When discussing the subject material of iCanDraw?, Bienko shares a similar sentiment: “Although they are most known for their ability to depict the world abstractly, Picasso, Mondrian, and Diego Rivera were each gifted realistic drawers. Knowing the proportions of the face are tools they utilized when crafting abstract masterpieces. When proportions are in place, the artist is free to travel well off the beaten path.”

If the teacher feels something is incorrect with a student's work, some say that the teacher should never make any corrective mark on the student’s work, but use open questioning to direct the student’s thought process [3]. However, such feedback may not always be possible due to limited instructor resource time in an online or beginner class. Without an instructor to provide informative feedback, a student may be uncertain about what to change to make the drawing look “right”. Bienko states, “It is far more important for them to SEE the changes that need to be made then it is for them to actually make the changes.” This paper presents an application that can provide feedback when a human instructor is not available.

Other forms of learning to draw do exist, such as the use of tracing paper or a grid to reproduce an image. The focus of this work, however, remains of those that engage the user’s perception and are in need of corrective feedback.

Drawing Instruction Adopted for this Work

The teaching style used in this work is step-by-step instructions complemented by corrective feedback to assist a user toward creating an accurate rendition of a reference headshot of a human model. Each step provides a written explanation describing the current task. The application provides tools and gestures for drawing, erasing, and marking strokes. Once satisfied with an attempt at a step, the user explicitly asks the application to check his or her work. Vision-based sketch recognition techniques are used to compare the freeform strokes to an underlying example template generated through face recognition on the image. If the application determines that drawing improvements can be made, textual and visual feedback is provided to the user, such that he or she may attempt to either improve the drawing, or accept it as is, whichever is desired by the user. The final drawing is evaluated by comparing a face-recognition generated template of the image to one generated from the drawing itself.

The domain of portrait photography was chosen as a starting point for this research due to the wealth of research around the human form in computers and in art. First, many beginners to drawing want their sketches to achieve a level of realism [8], resulting in works that do not deviate merely to be expressive and are easier to evaluate in an automated way. Additionally, the human head is something seen everyday by every person, yet most novice drawers will make common perception-based mistakes when reproducing it via a sketch, making drawing improvements readily noticeable. Third, step-by-step instructions, corrective feedback, and evaluative metrics can be created from the universal proportions shared by the features on most human faces (e.g. sides of the mouth should fall in line with the pupils). Lastly, face recognition techniques can be used to provide positional data for the facial features of the human model, facilitating sketch recognition.

RELATED WORKS

There are various books and websites [3,4,6,8,10,16] that teach drawing to the willing learner, though these lack immediate corrective feedback. To the best of our knowledge, this is the first application to use sketch recognition as a means toward assistance in drawing.

By preprocessing example imagery, Tsang created a 3D sketching application that provided a suggestive interface for fitting, closing, excluding, and general manipulation of user strokes toward the representation in the examples [19]. Intended as a professional design tool, its end product would be a derivative work and not a rendition of the example imagery.

Bae created an interactive application called "ILoveSketch" that borrowed many sketching affordances for drawing curves in a 3D space [2]. Also created with the professional designer in mind, ILoveSketch allows freedom in sketching but does not understand the sketch, making it unable to provide corrective feedback.

Another work created a heuristics engine that verified a user's sketch for a floor plan, providing text feedback [12]. The heuristics then verified that certain architectural features met their criteria (e.g. "the Emergency Room should be close to the Entrance Lobby"). While this work does provide feedback to improve a user's sketch, this paper presents techniques for such an application to also give visual feedback and on the drawing of human faces, a markedly different type of drawing.

Taele developed a language learning application called Hashigo that taught how to properly write Chinese Kanji by verifying stroke order, stroke direction, and visual correctness [18]. All characters existed as a series of geometric constraints, rigidly defining what the user should draw. This paper differs via perspective drawings and the user's ability to learn universal techniques.

LESSONS FROM THE FIRST ITERATION

The first version of this work sought to teach a user how to draw an iconic cartoon character (Figure 2). Using step-by-step instruction with an example and corrective feedback, users were guided through each piece, from head to ears to eyes, producing their own rendition of the famous character. Many participants found the subject material quite enjoyable to work with during an evaluative study

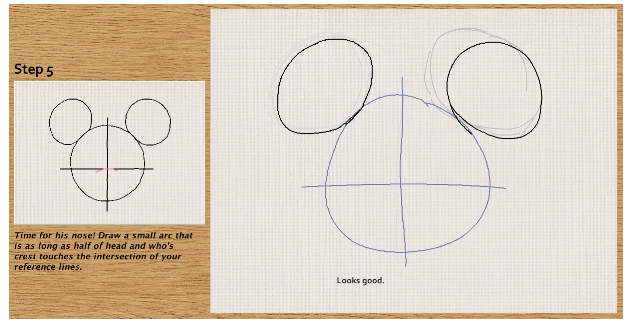


Figure 2 - The user interface of the first version. Instructions and example of the current step are on the left.

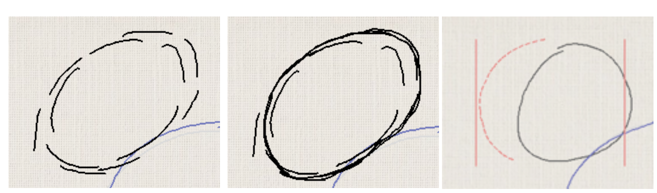
similar to the one described later in this paper. While unsuccessful from a user experience perspective, this first version provided many lessons learned.

This first iteration stuck closely to a user's conceptual model of paper [1,17]. On paper, strokes can be erased, strokes are not beautified as they can be in a sketch recognition application, and strokes can be traced over to emboss or emphasize them. We combined the first two properties to allow for "prep-sketching" and for "beautifying" the drawing by removing strokes that would normally be erased. We thought the third property would be a novel way to cue the functionality of the application.

"Prep-sketching" (Figure 3a) was meant to allow the user to mock up the placement and proportions of a shape using short strokes, adjusting as needed without triggering functionality from the application or committing to the shape. However, the beginning drawer generally does not mock up shapes and those that do will "prep-sketch" in their own unique ways (light strokes, long over-tracing strokes, never lifting the pen, etc.).

When the user liked a shape, s/he would commit to it by over-tracing the portion of the prep-strokes that s/he desired (Figure 3b). Embossing strokes then served as a trigger for functionality, such as corrective feedback or continuation to the next step. For the user, however, over-tracing does not always mean that s/he is done nor was this concept familiar to the beginning drawer. This also required the user to draw the requested piece in an all-or-nothing manner (i.e. draw the whole ear with one stroke to request feedback).

This first version also sought to automatically remove strokes that should be erased, like prep-sketch strokes and



(a) (b) (c)

Figure 3 - (a) "Prep-sketching" for a shape. (b) Embossing, or over-tracing, the shape. (c) Example of feedback initially used.

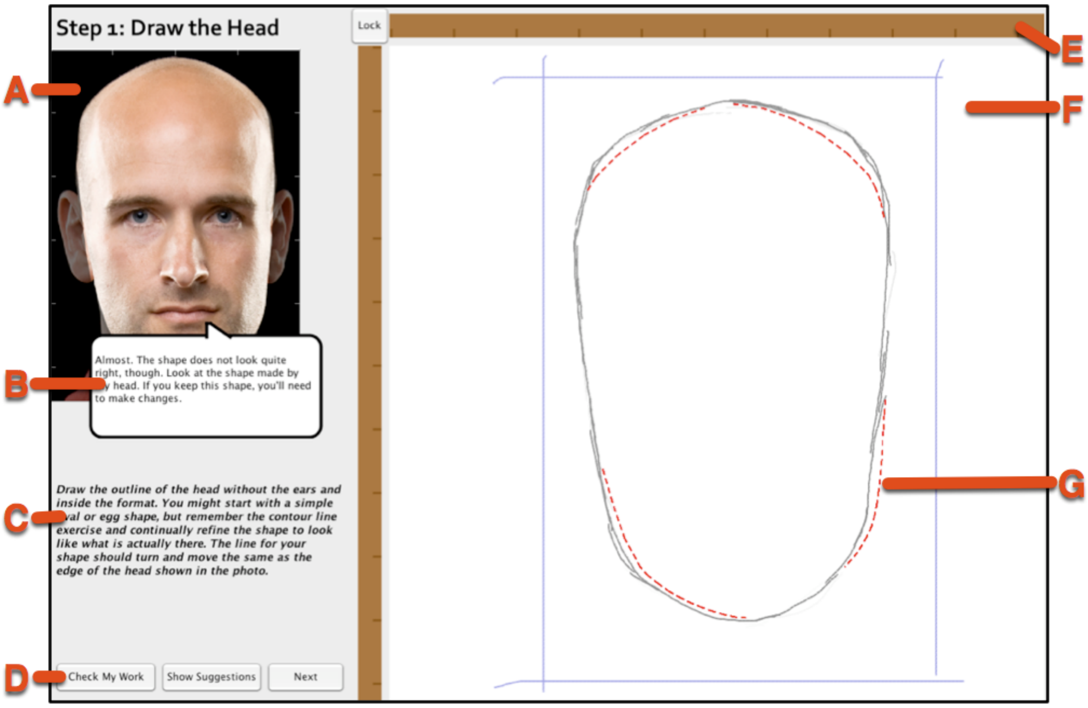


Figure 4 - Screenshot demonstrating many of the interaction techniques. (A) The reference image manipulated to show the head. (B) Text feedback. (C) Step instructions . (D) Options available. (E) Straightedge tool. (F) Drawing area. (G) Visual feedback.

incorrect strokes. However, a common compliant was that the drawing area became very messy given we opted for this feature instead of an explicit eraser function.

Lastly, the approaches used for corrective feedback on this version were ultimately not intuitive, having one form for the proportions and alignment of a shape and another for corrections to the shape itself. For example (Figure 3c), two vertical lines animated in an outward motion on a horizontal axis when a shape should be wider, and the area of difference was displayed in dashed form. The correct shape was also flashed briefly on the screen to help establish a mental image for the user, but did not persist to let him or her simply copy the shape. Simplifying the visual feedback with better text feedback was needed.

With the lessons learned from the first iteration, we went back to the drawing board (pun intended). This second version is an improvement for two reasons: (1) the user experience is improved through many of the nine design principles presented in their latter section and (2) human faces have many advantages for an assistive drawing application as discussed at the end of the second section.

USER INTERFACE

The user interface of the version described in this paper consists of a drawing area, a reference image to draw, and an area to provide instructions and options to the user (Figure 4). Each of these components plays a part when providing corrective feedback to the user.

Drawing Area

The drawing area supports these actions: draw a freehand stroke; draw a stroke with a straight-edge tool; mark a stroke as a reference line with a right angle gesture (Figure 5); erase a stroke with a scribble gesture across it; undo any of these actions with a rollback gesture [2]. Strokes erased by the user adopt a similar nature to strokes erased on a piece of paper by leaving a faint resemblance instead of being instantly removed. This serves as feedback to the user when redrawing to make a correction. If an erased stroke is older than few minutes, it is gradually faded out completely so the drawing area is concurrently cleaned up.

The first iteration of this work found that users can have difficulty drawing a straight line over an extended length, which can effect recognition and the placement of strokes on subsequent steps. To assist with this, a user can drag a straightedge tool into the drawing area to create straight horizontal and vertical lines (Figure 4), mimicking the affordance of a pencil against a ruler. Lines drawn this way are automatically marked as reference lines by a blue color,

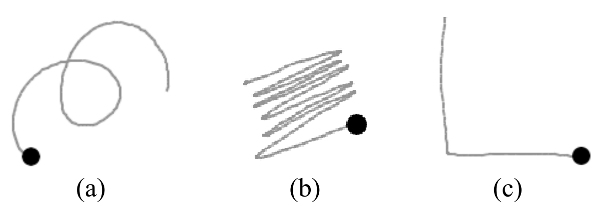


Figure 5 - Supported gestures. Rollback for undo. Scribble for erase. Right angle to mark reference line.

distinguishing them from the user's strokes intended for the final sketch. Many steps ask the user to draw reference lines for use in constructing proportions on the drawing. The user can also explicitly mark the last non-reference stroke drawn as a reference line with a right angle gesture (Figure 5c), also changing the stroke to blue. This grid-like work of reference lines made by the user can also be locked to prevent an accidental erase or hidden to see the final drawing without them.

Drawing Instructions

To lead the user, we wrote generic, step-by-step instructions centered around portrait photography (Table 1). These were made after a literature review [3,6,8,10,13] and included questions for the user to self-check the drawing. There were eights steps in all, some made of sub-steps for setting up reference lines and then drawing the piece.

From Step 2: "Follow the inside corner of an eye straight down on the photo and you should cross the edge of a nostril. Use this to put two short reference marks on your drawing for the width of the nose."
From Step 3: "Using the reference lines, draw the eyes just like you see in the photo. What bends and moves does the curve of an eyelid make? What shape is made by the white space around the pupil? How big are the eyes in comparison to other features? Are the eyes symmetric? Feel free to practice drawing an eye in the space around your drawing."

Table 1 – Part of the instructions for setting up and drawing the eyes in Steps 2 and 3.

Reference Image

The reference image is manipulated for each step to help the user see what to draw (Figure 6). When the step calls for a reference line on the drawing, a corresponding red reference line is shown on the image. By default, these reference lines collectively build on the image and persist with each step. The user can tap the image to see only the last references line(s) asked for, tap again to remove all lines and view the image, or tap once more to return all lines. When the step calls for drawing a facial feature, that feature is highlighted on the image by dimming all others.

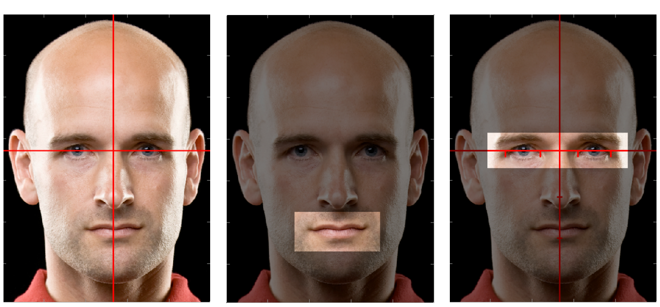


Figure 6 - Possible view states of the reference image, showing reference lines and/or highlighting facial features to draw. Tapping the image toggles the reference lines.

This adjusted view assists the user in focusing on the shape, appearance, and contours of the facial feature, presenting it as it is not normally encountered and making the user less likely to produce a mere symbolic representation of the feature. These manipulations assist in the R-mode shift and are applicable to all types of imagery, not just portrait photography.

Explicit Options Available

The user has three explicit options for initiating assistance from the application (bottom left of Figure 4). "Check My Work" performs sketch recognition on the user's drawing and verifies that what was drawn meets what was asked for by the current step. When initiated, corrective feedback is provided as text informing the user of what is wrong, or the user is told that the attempt is correct and s/he can continue. A first version sought to infer when the user was done with a task by an implied over-tracing his/her work, but this contributed to a poor user experience. After the initial "Check My Work" request, "Show Suggestions" becomes available when the user's attempt is incorrect and the application has visual guidelines to show. Otherwise, this option is not available. Visual guidelines are not automatically shown on a "Check My Work" request, allowing the user to make corrections him/herself based on the text. "Next" allows the user to advance to the subsequent step anytime after an initial use of "Check My Work". The first version explicitly avoided the use of buttons altogether and prevented step advancement on an incorrect shape, but would sequentially loosen thresholds until the user drew as asked. However, when creating subjective content such as human forms or likewise, it is beneficial to the user to permit him/her to disagree with the application's suggestions. The latter two options are returned to an unavailable state at the start of each step.

Corrective Feedback

Of the two forms of corrective feedback, text remarks and visual guidelines, the former appear whenever the user asks the application to check the work and appear as a comment bubble from the model (Figure 4). If the user's attempt passes all constraints against the example template, the application replies with a complimentary remark. However, if the constraints fail for the user's attempt, the application will provide text with what is wrong (e.g. "The nose needs to come down more. …"). If the text contains ambiguity in terms of which strokes are being evaluated, the strokes referred to will be temporarily changed to orange; e.g., the text may state, "My left eye is two small," and the strokes for the left eye of the sketched head will become orange temporarily. The user may elect to gain additional information from the computer by pressing "Show Suggestions." For a correction to a reference line asked for by the instructions, this visual guideline would be a red line in the proper location for the reference line on the user's drawing. For a facial feature, the application will generate the proper shape in red on the user's drawing by either using a piece of the underlying template made from the photo or via a copy of the user's strokes that satisfies the constraints, or “Show Suggestions” is not available. Because it is desirable to leave only user-created strokes on the drawing area, visual guidelines are removed after the user has drawn a single stroke and do not leave a mark on the drawing area. Like the step-by-step instructions, the text feedback is specific to portrait photography, yet the visual feedback is not limited to this one domain.

IMPLEMENTATION

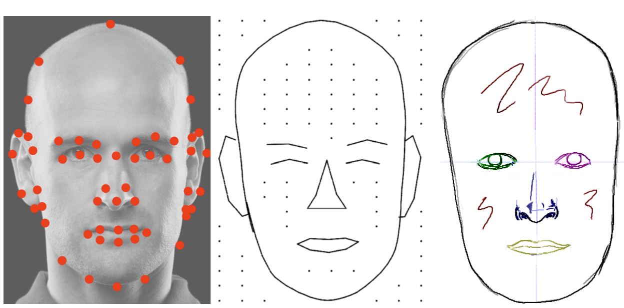
Pre-processing Reference Imagery

To provide corrective feedback on the user's drawing, the application must have an underlying template to which it can compare. The reference image undergoes pre-processing through a face recognition library that provides 40 data points for the facial features [11]. Those facial features include: corners and pupils of each eye; contour of each eyebrow; tip, bridge, and wings of nose; contour of the mouth; contour of the head. To represent the ideal case, any point that deviated too far from its corresponding feature was manually moved and data points were manually inserted for the contours of the ears, resulting in 53 points total (Figure 7a). A B-spline is fit to the head contour to produce a smoother representation.

Setting the Example Template

In a first version, many users ended up with a drawing cut off by the edge of the drawing area because correct proportions where not set initially. To assist the user and establish good practice, the first step of the second version has the user create a drawing boundary (e.g. the tall box in Figure 4). This boundary is checked to have the same aspect ratio as the reference image and, once set, the example template is centered and scaled to this size. After the user has successfully drawn the head, the template is permanently set to the size and scale of the drawn head.

Additionally, because users may want space to practice a shape, strokes drawn outside of the boundary are ignored by the sketch recognition. Likewise, an “ignore space” grid is superimposed on the template (Figure 7b) to define areas that should be ignored by the sketch recognition, such as strokes drawn on the cheeks or the forehead. The “ignore space” is computed by using an unseen ten column by twenty row grid on the format. A point is assigned to the “ignore space” grid if it is not within a static threshold distance of the data points from the face recognition. The static threshold chosen allows for an ample buffer between existing data points and any spaces to ignore, and is roughly the width between the eyes of the model (a measurement used when establishing the face)[8]. Thus, the example template consists of a set of data points each labeled with the facial feature or ignore space it belongs to.



(a) (b) (c)

Figure 7 - (a) Visualization of data points generated for example template. (b) The underlying example template after adding “ignore space” data points, a B-spline, and lines. (c) Color-coded visualization of user’s drawing when classified.

Processing of a Stroke

A “stroke” drawn by the user is a collection of data points drawn in-between pen-down and pen-up on a tablet screen. To assist in sketch recognition, each stroke is re-sampled so that the data points are equally distant from one another along the stroke. Gestures are recognized using [15].

Allowing Freedom to Draw

Though the application guides the user step-by-step, s/he may still wish to modify work done on a prior step. For example, the user may decide to erase and redraw an eye after noticing its relationship to an eyebrow drawn later. To allow the user to draw anywhere but so the application can select the strokes desired when checking constraints, the application reclassifies all strokes of the user’s sketch on a “Check My Work” request through a k-Nearest-Neighbor (kNN) classifier ($k = 3$) using the data points of the underlying template [9,20]. To classify a stroke, its endpoints and center sample point are matched to the closest data points on the template via Euclidean distance and then placed into its appropriate “pool” of similar strokes (visualization in Figure 7c). For example, if each selected point of a stroke is matched with a data point for the head contour from the template, it is put into the pool of strokes for the head. An odd number is used for the points to classify a stroke to prevent equal votes between the possible classifications, unless all three receive different classifications leading the classifier to default to that of the first end point. All strokes drawn outside the example template are immediately classified as ignore strokes. Of importance is that this technique is only used on the steps that it can support. For latter steps where no recognition data points are known or cannot be inferred from the facial recognition data (i.e. ears, neck), the classifier is not used, existing strokes assume their prior classifications from previous steps, and new strokes are analyzed based on their positions to the facial feature of that step.

Determining Correctness of the Sketch

Given classifications and pools of all strokes, the pool for the facial feature of the current step must then pass through constraints against its corresponding piece in the example template. For reference lines, the location and end points of

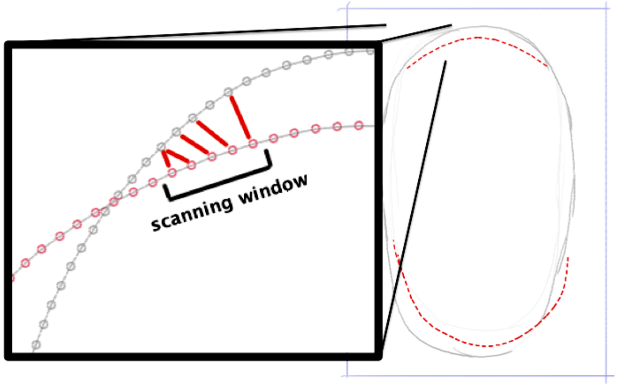


Figure 8 – Visualization of the simple scanning window algorithm used for comparing a user's drawing (grey) to the underlying template piece (red).

the user's line are checked for coincidence at an 80% confidence level or greater with the corresponding data points on the example template. For example, the eye reference line (a horizontal line used to place the eyes) asked for in the second step is checked against the data points the pupils of the eyes. Probabilities for confidences are calculated using an univariate Gaussian distribution.

For a facial feature, the proportions and alignment of the user's shape are checked against the template piece by comparing the bounding boxes to determine if the two are the same using an 80% confidence level. This constraint helps the user get his/her shape for the current step inline with the example template while still allowing variance.

Next, for the features with ample data points (i.e. the head), the user's shape and template piece pass through elastic matching to determine the differences (Figure 8). For each re-sampled point, the algorithm uses a window of neighboring points on each side (w = 2) of the point as it moves along all the re-sampled points of the user's shape and finds any regions that vary greatly [22]. A value is given to a window instance by matching each of its points to its closest point on the template piece and then summing the Euclidean distances. When the algorithm has finished scanning, the average and standard deviation for all windows is calculated. If a window's distance is greater than one standard deviation from the average, it varies enough to need correction and is returned as part of the visual guidelines shown to the user. This constraint is only satisfied when the standard deviation is less than 60% of the average. The goal of this simple algorithm was to be invariant to direction, start and end points, and over-tracing so that the user is not required to draw in a particular manner. For facial features without sufficient data points on the template, the user's shape is reused and fitted to the constraint or no visual feedback is provided.

DESIGN PRINCIPLES GAINED

As one result of both development cycles and user studies, nine design principles for assisting the act of drawing using sketch recognition were developed:

1. In order to provide an accurate error measure of the drawing via sketch recognition, it is vital that any master template used for comparison be correct and thus a formal process should exist to determine any error in the template itself. In our original implementation, the template was hand-drawn and contained an undiscovered error until well into its user study. In our current implementation the template is generated through face recognition then examined but hand-tweaked as necessary, better ensuring that the user is not falsely corrected by the sketch recognition process. Using additional image analysis on the reference photo (e.g. edge detection) to gain information for sketch recognition is advantageous. Face recognition is only one approach. If using a hand-drawn template, other solutions could including having the creator draw the template multiple times for comparison or have a peer review the template before use.
2. To guide the user's perception of the object to draw, the application should help the user process and "see" the object. This is the R-mode shift where a person is better equipped to perceive the edges, spaces, and relationships that make up a scene. Aspects of the application should be complimentary to this shift (e.g. asking questions to the user about the object, presenting reference imagery in different ways such as highlighting pieces or flipping it upside down).
3. Feedback should be supplied only when explicitly asked for, otherwise it may interrupt the creative thought process. The first version attempted to automatically infer when a user had completed a task and when feedback should be supplied. However, the completion of the shape requested in the instructions does not always mean the completion of the task for the user. Drawing is an iterative process, and users frequently draw, pause, and re-draw, after personal evaluation.
4. Corrective feedback must be clear and constant, showing what is being asked for and why it is being asked. The feedback should also be directed towards the correct completion of the task, rather than be relative to the user's current attempt of the task. In the first version, feedback was altered based on the user's previous attempt instead of blankly relating to the evaluation template. This caused confusion since, as the user made multiple attempts, the feedback appeared to change and direct the user to alternative outcomes.
5. Previous or "erased" strokes should be temporarily visible as a form of corrective feedback. These strokes help prevent a repeat of the same mistake. This is very similar to how strokes erased by a pencil on a piece of paper will still persist faintly and can serve as a reference for a previous attempt.
6. Users must be allowed to continue sketching no matter what the application's suggestions are for the user's most recent strokes. The first version required users to perfect their drawing up to an acceptable point before moving on. Many insisted that their sketch was 'good enough' and preferred to

move to the next step. The current application as a 'Next' option for use at any time no matter what has been drawn.

7. In addition to being constant across user's attempts on a piece (see #4), corrective feedback must be adaptive to mature pieces of a sketch. As the previous principle states, users can feel that their sketch is 'good enough.' The underlying template may need to conform as the user draws and commits to pieces of the sketch. For example, the scale and location of the template are modified after the head-contour is accepted.
8. A large and properly primed drawing area is advantageous to the user creating subjective content given that proportions are one of the first things the beginner struggles with. Many users of the first version failed to gauge proper sizes while laying down earlier pieces, thereby causing latter pieces to run off the page. The current version introduces a drawing boundary as an opening step and hosts a larger drawing area.
9. A drawing application needs to be mindful of artistic affordances, such as light/dark values, pencil types, crosshatching, symmetry, etc. The user's perception of a sketch is not limited to just lines and some features are only brought out through shading, which sketch recognition will need to interpret and handle.

QUALITATIVE EVALUATION

Five participants tested the application on a Wacom Cintiq 21UX monitor in a two-session user study (see Appendix). The study consisted of performing a blind contour drawing exercise as a warm-up for each session and then creating drawings of human models either freehand or with assistance. All participants were male (M1-M5) and were selected based on their "confession" to not being able to draw. After a demonstration of the tools and gestures, each participant was asked to perform these exercises:

1) Freehand drawing of a male baby to produce a drawing that was exemplary of his drawing ability.
2) Assisted drawing of a different male baby with step-by-step instruction and feedback.
3) Freehand drawing of the first male baby again in order to see what differences would be made within a sitting.
4) Assisted drawing of an adult male.
5) Freehand drawing of an adult male.

The purpose of this study was to determine how the beginning drawer would acclimate to assistive drawing computer application. Most participants expressed feeling overwhelmed given the task to draw a human figure. The authors expected this and viewed this perceived difficulty of the task as validation of the task's use as a test bed that would show the most improvement in the participants. All participants drew the same sequence of images for comparative purposes, though a larger study might rotate the image order. When asked what was wrong with their drawings after the first exercise, most participants could point out incorrect features. M2 expressed that he did not know how to convey what he wanted in the sketch.

After the second exercise, all participants expressed that they found the step-by-step approach useful. Three of the five participants commented that they did not have much prior knowledge of the symmetry and proportions found in the human face. M3 liked how the dimming of facial features on the reference image helped him concentrate on drawing the desired one. All participants had some difficulty using the gesture for marking reference lines. M2 also had difficulty adjusting to the straightedge tool. The gradual fade of erased strokes was subtle enough that many did notice their removal. The feedback provided by the application also took adjustment on the part of the participant. If the participant found the suggestion useful, he would make the correction and move on. However, some suggestions were perceived as incorrect when they were not, or the application could not account for a participant's conceptual model of how it should work (e.g. M2 created a grid in the drawing area to help him scale up the picture while drawing it). Generally, the participant would respond with two or three additional attempts but then opt to advance to the next step if the application still did not accept his work. There were instances where the application did misinterpret the participant's shape and provide incorrect feedback, causing the participant to simply advance the step.

For the third exercise, a common sentiment was that the new drawing was more symmetric, proportions were better, and it looked more like a child. All could still find incorrect parts when asked. M2 expressed that his "drawing was better because of the [corrective feedback]."

For the latter two exercises, each participant did an assisted and a freehand drawing of a male adult. The common sentiment was that latter drawings had better symmetry and proportions and were more identifiable. Besides M3, another common sentiment was that the reference image of the baby was actually harder to draw than that of an adult. M3 and M4 verbally expressed more confidence to draw, and M3 said that the last freehand exercise was easier after doing the assisted exercises.

One concept that had to be conveyed to users was that all strokes on the screen are evaluated in some form, so they were encouraged to erase any stragglers that were not actually in use to improve the feedback. Users also learned to only do what was asked for by the step to improve feedback. For example, adding lots of shading and extra detail to a mouth when the step only asked to establish it's contours and size affected the feedback. The last step of the exercise allows them free reign to add details to any features after they have used the tool to establish the face.

When evaluating the application and the participant's drawings, artist Joshua Bienko stated, "iCanDraw? goes a long way toward developing…seeing skills. This is notable for instance, in the improvements made on the position of the nose (which can be thought of as the 'anchor' of the face) [by] Male #2…Noses and mouths are among the most difficult features to draw, let alone to place. When drawing the face, it is common to see the most beautifully rendered eye placed somewhere on the forehead, or a wonderfully drawn nose placed…between the eyes. Drawers then react to these incorrect observations and the drawing goes from looking close to, well, not even close! iCanDraw? eradicates many of these errors by exposing them.” Bienko also suggested using black and white reference imagery due to the perceptual tricks that color can play on the eye, and to include visual diagrams in lieu of heavy text instructions to cut down on unintended interpretations. “I think iCanDraw? can be utilized to give students the tools with which an instructor can build on” Bienko concluded.

QUANTITATIVE EVALUATION

To quantify the variance that a drawing contains from its reference source, face recognition software is again used under the ideal case (i.e. manual corrections and insertions) to generate a template from the drawing itself. The new template T_d is then compared with the template T_p generated from the reference photograph. A summation of the distance found between the corresponding pairs of points of each template leads to an overall error for the user’s drawing. Similar to the approach used by [23], two corresponding pairs of points (p_i^d, p_j^p) between the templates are compared via:

$$e\left(p_i^d, p_j^p\right) = \sqrt{e_\theta^{\ 2}(p_i^d, p_j^p) + e_{xy}^{\ 2}(p_i^d, p_j^p) + e_\ell^{\ 2}(p_i^d, p_j^p)}$$

$$e_\theta(p_i^d, p_j^p) = \left|\theta_i - \theta_j\right| \qquad e_\ell^{\ 2}(p_i^d, p_j^p) = \left|\ell_i - \ell_j\right|$$

$$e_{xy}(p_i^d, p_j^p) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

The orientations of each pair of points are represented by θ_i and θ_j, (x_1, y_1) and (x_2, y_2) are the locations of each pair’s calculated midpoint, and lastly ℓ_i and ℓ_j are the lengths each pair. With P as the number of corresponding pairs for a facial feature f, the overall error is then calculated as:

$$e(T_d, T_p) = \sum_{f \in T} \frac{1}{P} \sum_{(i,j) \in P} e\left(p_i^d, p_j^p\right)$$

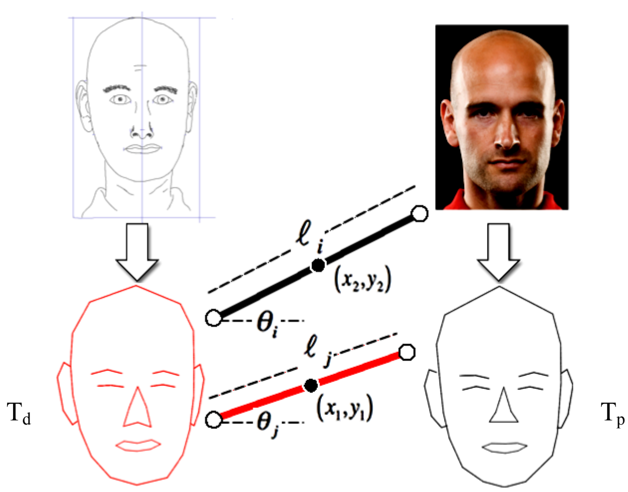


Figure 9 - Visualization of how templates from the drawing and the reference image are compared.

Figure 9 is a visualization of comparing a face recognition template from a user’s drawing to one from a reference photo. To align the templates, T_d is scaled to have a matching height and moved to have the same bounding box center point as T_p. The lower the error, the more proportionally correct the sketch was drawn. The errors calculated from template matching for each drawing of the user study are shown in the Appendix, however this approach was not implemented at the time of the study. All error counts shown in the Appendix for the user study generally reflect what the authors expected. In every case, the error on a participant’s drawing for an assisted exercise is lower than when the user is unassisted. When comparing the participants to a control group who had the same instructions and changing reference image yet the corrective feedback was turned off, the average error was roughly a factor of two lower for the assisted participants and their standard deviation was significantly lower. Given the brevity of this user study, the authors admit it cannot prove the validity of the application as a teaching tool but can confirm its stance as a proof of concept for an end-to-end system.

CONCLUSION

This paper presents the first system for using computer-aided instruction to assist a student in learning to draw human faces. This system uses face and sketch recognition to understand the reference photograph of a human model and a user’s drawing of it. The subject of the human head was chosen because it has many advantages. First, it’s a drawing task enjoyable to achieve. Second, improvement is quickly noticed as many make common mistakes with drawing the human head. Third, a body of knowledge already exists for drawing the human head and face. Fourth, a wealth of image-based techniques can be built upon to implement effective sketch recognition. The final application built under this system uses principles obtained from art literature in concert with principles gained from user interactions with the versions of the application. The iterations of development, lessons learned, and evaluation techniques were also recounted and user experiences shared.

REFERENCES

1. Alvarado, C. “Sketch recognition user interfaces: guidelines for design and development.” In AAAI 2004 Symposium on Making Pen-Based Interaction Intelligent and Natural (Oct 2004), 8-14.
2. Bae, S.-H., Balakrishnan, R., and Singh, K. “ILoveSketch: as-natural-as-possible sketching system for creating 3d curve models.” In Proc. of 21st Annual ACM Symposium

on User Interface Software and Technology, (Oct 2008), 151-160.

3. Bartel, M. "Teaching creativity (2008)." http://www.goshen.edu/~marvinpb/arted/tc.html.[Accessed Jan 2009].
4. Bartel, M. "How to teach drawing to children." 2007. http://www.goshen.edu/art/ed/draw.html. [Accessed: Jan 2009].
5. Bruer, J.T. "In search of...brain-based education." 1999. http://www.pdkintl.org/kappan/kbru9905.htm.[Accessed 1/2009].
6. Clark, G., and Zimmerman, E. Teaching Talented Art Students - Principles and Practices. Teachers College Press, New York, NY, USA, 2004.
7. Corballis, M.C., and Beale, I.L. The Psychology of Left and Right. Erlbaum, Hillsdale, NJ, USA, 1967.
8. Edwards, B. The New Drawing on the Right Side of the Brain. Penguin Group, Inc., New York, NY, USA, 1999.
9. Hammond, T., and Davis, R. "LADDER: a language to describe drawing, display, and editing in sketch recognition." In Proc. of 2003 International Joint Conference on Artificial Intelligence (IJCAI), (August 2003), 461-467.
10. Hoddinott, B. "DrawSpace.com." 2009. http://www.drawspace.com. [Accessed: Mar 2009].
11. Luxand Inc., Luxand FaceSDK, version 1.81, 2009.
12. Oh, Y., Do, E.Y-L., and Gross, M.D. "Critiquing freehand sketching - a computational tool for design evaluation." In Visual and Spatial Reasoning in Design (VR), Cambridge, MA, USA, (2004), 105-120.
13. Orde, B.J. "Drawing as visual-perceptual and spatial ability training." In Proc. of National Convention of the Association for Educational Communications and Technology, Albuquerque, New Mexico, USA, (1997), 271-278.
14. Pariser, D.A.. "Two methods of teaching drawing skills." In Studies in Art Education, 20, 3 (1979), 30-42.
15. Paulson, B., and Hammond, T. "PaleoSketch: accurate primitive sketch recognition and beautification." In Proc. of 13th International Conference on Intelligent User Interfaces (IUI) (Jan 2008), 1-10.
16. Roam, D. The Back of the Napkin – Solving Problems and Selling Ideas with Pictures. Penguin Group, Inc., New York, NY, USA, 2008.
17. Sezgin, T.M., Stahovich, T., and Davis, R. "Sketch based interfaces: early processing for sketch understanding." In ACM SIGGRAPH 2007 Courses, August 2007, No. 37.
18. Taele, P., and Hammond, T. "Hashigo: a next-generation sketch interactive system for Japanese kanji." In Twenty-First Innovative Applications Artificial Intelligence Conference (IAAI), (Jul 2009).
19. Tsang, S., Balakrishnan, R., Singh, K., and Ranjan, A. "A suggestive interface for image guided 3D sketching." In Proc. of SIGCHI Conference on Human Factors in Computing Systems, (Apr 2004), 591-598.
20. Watt, S.M., and Xie, X. "Prototype pruning by feature extraction for handwritten mathematical symbol recognition." In Proc. of Maple Conference 2005, Maplesoft, (2005), 423-437.
21. Willats, J. "How children learn to draw realistic pictures." In The Quarterly Journal of Experimental Psychology. 29, 3 (Aug 1977), 367-382.
22. Wolin, A., Eoff, B., and Hammond, T. "ShortStraw: a simple and effective corner finder for polylines." In Eurographics 5th Annual Workshop on Sketch-Based Interfaces and Modeling, (2008), 33-40.
23. Xu, Z., and Luo, J. "Face recognition by expression-driven sketch graph matching." In Proc. 18th International Conference on Pattern Recognition (ICPR), (2006), 1119-1122.
24. Yang, M.C., and Cham, J.G. "An analysis of sketching skill and its role in early stage engineering design." In Journal of Mechanical Design 129, 5 (May 2007), 476-482.

APPENDIX