# VoiceDraw: A Hands-Free Voice-Driven Drawing Application for People with Motor Impairments

Susumu Harada
Computer Science and Engineering
Box 352350
University of Washington
Seattle, WA 98195 USA
harada@cs.washington.edu

Jacob O. Wobbrock
The Information School
Box 352840
University of Washington
Seattle, WA 98195 USA
wobbrock@u.washington.edu

James A. Landay
Computer Science and Engineering
Box 352350
University of Washington
Seattle, WA 98195 USA
landay@cs.washington.edu

## ABSTRACT
We present VoiceDraw, a voice-driven drawing application for people with motor impairments that provides a way to generate free-form drawings without needing manual interaction. VoiceDraw was designed and built to investigate the potential of the human voice as a modality to bring fluid, continuous direct manipulation interaction to users who lack the use of their hands. VoiceDraw also allows us to study the issues surrounding the design of a user interface optimized for non-speech voice-based interaction. We describe the features of the VoiceDraw application, our design process, including our user-centered design sessions with a "voice painter," and offer lessons learned that could inform future voice-based design efforts. In particular, we offer insights for mapping human voice to continuous control.

## Categories and Subject Descriptors
H.5.2 [**Information interfaces and presentation**]: User Interfaces – *Voice I/O.*

## General Terms
Design, Human Factors.

## Keywords
Voice-based user interfaces, speech recognition, drawing, painting, computer art, continuous input, motor impairments.

## 1. INTRODUCTION
Creative self-expression and artistic endeavors can play a vital role in enhancing people's quality of life, including those with various types of disabilities [22]. Despite the challenges that motor impairments pose to an individual's ability to manipulate physical art mediums such as paint brushes or drawing pencils, numerous people have overcome disabilities through creative re-adaptation of existing tools. For example, the Association of Mouth and Foot Painting Artists of the World (AMFPA) [1] comprises artists with various disabilities affecting the use of their hands who create artwork using their mouths or feet.


*ASSETS* '07, October 15-17, 2007, Tempe, Arizona, USA.

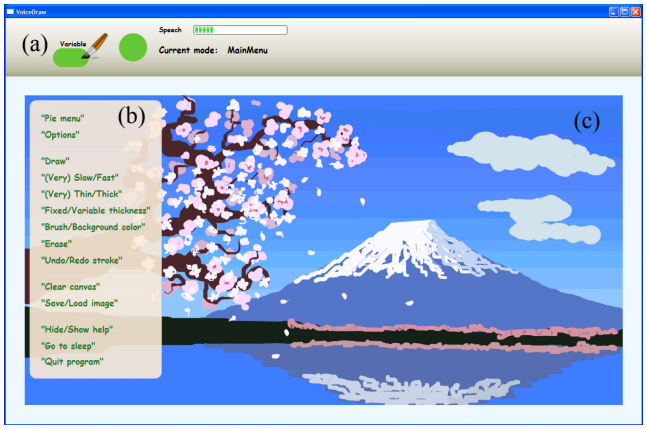




**Figure 1:** A screenshot of the VoiceDraw application[1] showing **(a)** the status bar, **(b)** help overlay, and **(c)** canvas area. The first author created this painting using only his voice in about 2.5 hours.

However, for those with moderate to severe motor impairments, manipulation of physical tools may be difficult or impossible. Even those with some ability to manipulate physical artistic media may find the process arduous enough to be a barrier to engaging in creative activity.

Computer applications hold promise for enabling such individuals with limited motor abilities to engage in creative activities with reduced overhead of manipulating physical tools. Painting programs on a computer can simulate physical brush strokes or even provide artistic effects not possible in the physical domain.

A challenge that limits the realization of this potential is that many of today's computer applications remain inaccessible to people with motor impairments [2]. One of the main reasons for this is that modern graphical user interfaces (GUIs) assume that users have the ability to move a mouse cursor, especially in painting and drawing applications. In fact, many users with motor impairments have difficulty moving a mouse cursor or performing a variety of other continuous control tasks [12]. To address this issue, eye-tracking [10] and head-tracking [13] have been investigated as methods for continuous control, but these have notable drawbacks, including the need for high-end equipment and high cost, configuration, and maintenance. Prior work shows that these factors can be significant barriers to adoption and retention of assistive devices [5][18].

[1] A video demonstration of the VoiceDraw system can be obtained from http://ssli.ee.washington.edu/vj/voicedraw/.

As an alternative, we investigated the use of non-speech vocalizations as a method for providing users with motor impairments the fluid control essential for expressive interaction. In contrast to speech recognition, non-speech vocalizations refer to vocal sounds that do not correspond to any words or phrases in a language. For example, pitch, volume, and vowel quality are all continuous features of voice that may be manipulated by users in control tasks. To situate the study of these issues, we developed *VoiceDraw* (Figure 1), a fluid drawing application that allows users with motor impairments to create artwork by using the non-speech properties of their voice.

In this paper, we describe the design, implementation, and evaluation of VoiceDraw. Our process included interviews, field investigations, and user-centered design sessions over a period of two weeks with a self-described "electronic voice painter" who formerly used Dragon Dictate and Microsoft Paint to produce his own artwork. Based on our sessions with this artist, we made numerous iterative design refinements. Ultimately, the contributions of this work include the VoiceDraw system, new voice-based interaction techniques (e.g., vocal marking menus, continuous undo), and some lessons learned, which can be used to inform future designs for voice-based user interfaces.

# 2. RELATED WORK

## 2.1 Drawing without Hands

The closest works to VoiceDraw are art installations and exhibits. Levin et al. [14] present interactive art installations in which the sounds generated by the participants are captured by the system and rendered as an artistic projection onto a publicly visible surface. Although the system controls the appearance of the generated image based on the phonetic features of the incoming audio, it does not provide users with a level of control comparable to traditional painting.

An exhibit called the VoicePainter [21] in New York allows children to make vocalizations into a microphone while the system populates a digital canvas with random pre-stored patterns in response to sound. Based on the descriptions in the press release, it does not appear that the features of the voice are correlated with the patterns selected.

Beyond voice, EyeDraw [10] enables children with severe cerebral palsy to draw using their eyes and an eye tracker. The latest version supports the ability to add straight lines, rectangles, ellipses, and predefined icons, but no free-form strokes as we do in VoiceDraw. The creators of EyeDraw note the limitations of eye-based interaction in supporting "free-eye" drawing due to the inability of the eye to move in slow continuous movements, and the overloading of drawing and viewing with movements of the eye (i.e., the Midas Touch problem).

In exploring the combination of a head tracker and speech recognizer for supporting hands-free pointer control, Malkewitz [15] compares a simple circular figure painted with his system to that made by the mouse. However, the paper does not provide any further discussion regarding the design issues raised.

## 2.2 Voice-Based Pointer Control

There have been voice-based pointer control systems that attempt to provide the continuous control needed by applications such as drawing programs. Igarashi et al. [11] proposed the use of non-speech vocal parameters (such as volume, pitch, and vowel sounds) as a means for achieving direct manipulation via voice. Subsequent systems have used voice to control the mouse pointer [4][6][16][17][19], but none offers direct manipulation for fluidly controlling the mouse's direction and speed. For a comparative discussion of different voice-based pointer control techniques, the reader is directed elsewhere [9].

VoiceDraw incorporates the Vocal Joystick engine [3] for 2-D control of the paintbrush and discrete sound recognition used for mode switching. The Vocal Joystick engine continuously processes a user's vocalizations and recognizes the corresponding vowel sounds. The pointer moves in the direction of vowel sounds based on the mapping shown in Figure 2. This occurs for as long as the user continues vocalizing, at a speed proportional to the loudness of the vocalization. The system also recognizes discrete sounds, such as "ck" and "ch", which can be mapped to input events such as mouse-click and toggle.

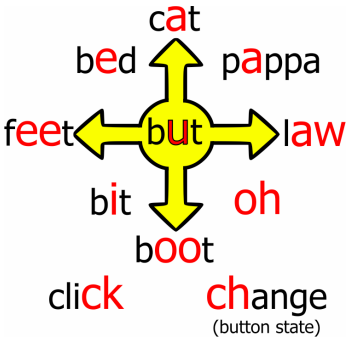



**Figure 2:** Directional mapping of vowel sounds (in bold red) used in VoiceDraw. The middle vowel is a neutral sound not mapped to any direction. The two non-vowel sounds on the bottom are the discrete sounds "ck" and "ch".

# 3. FIELD INVESTIGATIONS

## 3.1 Interview with a "Voice Painter"

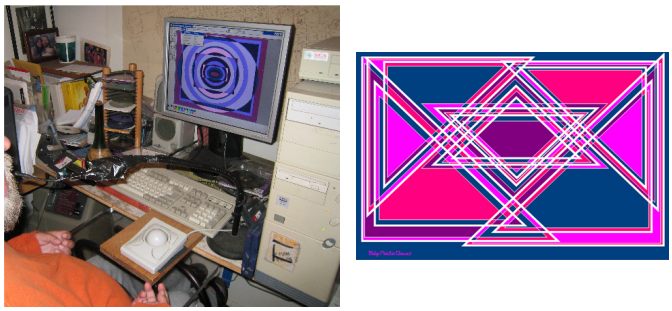
Despite the absence of tools that enable drawing on a computer using voice, we were able to find an artist who refers to himself as an "electronic voice painter." In order to gain an understanding of the current challenges he faces with his existing tools, we conducted a series of user-centered design sessions, starting with a situated semi-structured interview.

### *3.1.1 Background*

Philip Chavez[2] is about 60 years old and has been creating art on his computer using his voice for over 15 years. He has had a spinal cord injury at the C4-C5 level for about 30 years, and has limited movement in his shoulder. As a result, he has no dexterity in his elbow or wrist, and no sensation in his hands. His speech is unimpaired, but he does have limited lung capacity, which restricts his ability to produce long uninterrupted vocalizations.

His inspiration to create art on his computer came after a period of successive tragic events, leading him to seek an outlet for his emotions. He experimented with Microsoft Paint, a basic painting program that ships with Windows 95 and later. With Paint, he used Dragon Dictate speech recognition software to create his first "voice art" piece. Since then, he has produced hundreds of pieces

[2] The artist has specifically requested that we use his real name and that we credit his artwork, which can be found at http://www.evoiceart.com/.



(a) (b)

**Figure 3: (a)** The voice painter's computer setup, and **(b)** one of the art pieces produced by the voice painter using his current tools prior to being introduced to VoiceDraw.

of art, many of which are viewable on his website. Many others have been on display at various exhibits.

Much of Mr. Chavez's work is abstract, composed mainly of straight lines, geometric shapes, and solid color fills. This is due in part from the constraints of his current tools, but he also attributes much of his style to Jackson Pollock and the abstract expressionists, as well as his Navajo and Mescalero Apache heritage. He is also a strong believer in R. Buckminster Fuller's concept of "ephemeralization," of doing more with less [7]. Mr. Chavez told us, "there are many things that can be accomplished faster if you use very basic tools … When your options are too great, it can inhibit the creative process." This is reflected in the experience he had when he once tried using Adobe Photoshop and found it too frustrating due to the large number of features, particularly since many of them were not easily accessible using speech commands.

### *3.1.2 His Current Tools*

The current computer setup used by the voice painter is shown in Figure 3a. The primary mode of his interaction with the computer is through speech commands using Dragon Dictate Classic (version 3) and a microphone on a stand. He is also able to use a trackball by controlling his arm from the shoulder and using the back of his hand to move the trackball and to click on buttons. Despite the lack of tactile sensation in his hands, the voice painter has gained enough precision to click on targets as small as a standard scroll bar arrow. However, he still prefers to use speech as his primary modality, and only relies on the trackball when speech fails. A drawback of using the trackball is that it causes neck and shoulder strain, so speech is preferred.

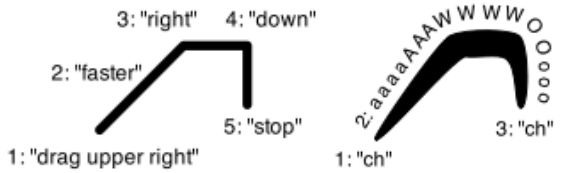
The voice painter has tried a number of mouse alternatives, including an eye tracker, head tracker, and a mouth-operated joystick, but has settled on speech for a number of reasons. He found the eye tracker to be difficult to control with the level of precision he desired, and found the process of turning off the tracking mode to be too cumbersome as he frequently needs to recline his powered wheelchair to stretch out his back. With the head tracker, he found the neck and shoulder strain to be too great for prolonged use. He was also unable to get his desired level of control with the mouth-operated joystick, and did not like the fact that it could get quite messy.

### *3.1.3 Current Drawing Process*

For the voice painter, the first step in interacting with his computer typically begins with the activation of his speech recognizer by issuing the verbal command "wake up." Up to this point, the speech recognizer was in "sleep mode," where all verbal commands and extraneous sounds were ignored except for the "wake up" command. (If the speech recognizer was not already running, it can be manually invoked using the trackball.) The voice painter then uses a sequence of verbal commands to open a Microsoft Paint file that has been pre-configured to the appropriate dimensions. He then begins the drawing process.

Once Microsoft Paint has been launched, the voice painter selects a tool from a tool palette or a color from the color palette by using the MouseGrid feature, with which a point on the screen is selected by recursively calling out one of the nine grid cells of a subdivided screen area [4].

"Strokes" are made using discrete speech commands. The voice painter begins moving the mouse cursor in one of the eight 45° angles using a speech command like "drag upper right." The pointer begins moving in the specified direction at the specified speed, both of which can be altered on the fly by uttering commands such as "much faster" and "move left." The artist also occasionally moves his trackball while the pointer is in motion to intentionally introduce randomness into the stroke. Issuing the "stop" verbal command terminates pointer movement. Figure 4a illustrates how the voice painter makes a stroke using speech commands. To change the color of the brush, he must use the MouseGrid or trackball to select the desired color from the color palette. He also often uses the paint bucket tool to fill an area with a solid color. A sample piece produced by the voice painter using this current process is shown in Figure 3b. The time spent on each piece ranges from a few hours to 40 hours, with some taking up to 100 hours.




(a) (b)

**Figure 4:** Sample interaction of drawing a stroke using **(a)** speech-based cursor control, and **(b)** VoiceDraw. With speech-based control, changes are discrete and incremental. With VoiceDraw, changes are continuous and fluid. Loudness of the utterance can be mapped to stroke thickness or brush speed.

### *3.1.4 Limitations of the Current Setup*

The voice painter pointed out a number of limitations that he encounters when using his current tools. One problem is when the speech recognizer fails, especially for mid-stroke corrections. When the voice painter is in the process of making strokes that consist of numerous direction changes similar to those shown in Figure 3b, if the speech recognizer fails to recognize a command, the pointer continues moving beyond the desired point. In order to recover from such errors, the entire stroke needs to be undone, which can be very costly if the error was near the end of a long or complex stroke. This is a general problem with so-called "passive modes" and the lack of continuous control using speech commands. Once a command sets a mode (e.g., brush movement), another command must be successfully issued to stop or amend it. The use of the eraser tool is also plagued by this problem, as an area larger than desired may be erased accidentally, requiring the whole erasure to be undone.

Another limitation raised by the voice painter pertains to the expressiveness afforded by the current tool. He stated his frustration, saying, "I can't get real brush strokes, and can't get the texture I'm thinking about." He also pointed out that he would often have a vivid image in his mind that he wished to express; the image often included curves and smooth shapes, but with the current tools, he was only able to achieve a crude representation using ovals or by manipulating individual pixels. This motivated us to prioritize supporting continuous curves as an essential feature in VoiceDraw's repertoire.

When we asked the voice painter whether he feels attached to his tools given their role in shaping his artistic style, he stated:

> *"I think I'm out-growing this [Microsoft Paint] and am ready to explore new options. My artwork and progression is limited by the tools I have. I'm excited about the possibility of expanding that."*

## 3.2 Design Goals

Based on the interviews with the voice painter, we focused on the following design goals for our initial implementation. We acknowledge that they are based on our interaction with only one artist, and that they will need to be supplemented by feedback from additional users in the future. However, we found that the insights and design goals derived from an initial deep exploration can provide us and other researchers with a solid starting point for iterating towards a more general design in the future.

### *3.2.1 Continuous and Fluid Input*

A number of issues raised during the interview can be traced to the fact that the current method and tools lack necessary direct manipulation and fluidity. For example, the voice painter cites the ability to express realistic brush attributes such as variable thicknesses and smooth curves as being key to expressing some of his artistic visions. Also, his use of the trackball for introducing randomness during pointer movement reflects his need to have direct, immediate control over his output. In order to support such realism and expressiveness, our new tool must be able to:

- provide fluid direct manipulation of the brush, and
- provide the ability to simultaneously manipulate brush characteristics during motion.

### *3.2.2 Sustaining the Flow of Creative Process*

The voice painter stressed the importance of not interrupting the flow of creativity: "When I'm in the zone, the cumbersome maneuver to get the [stroke] angle I want hinders the creative process." He also expressed the following:

> *"If I can verbally change the angles without stopping, it probably wouldn't interrupt the creative process so much, and I'll be able to create much, much better pieces."*

In light of these sentiments, we strove to enable fluid control over brush angles and characteristics (e.g., thickness) without the need for accessing menus, toolbars, or issuing discrete commands, all of which can break the creative process.

### *3.2.3 Reducing Strain*

One important consideration in reducing the artist's burden is to keep training time to a minimum. The voice painter mentioned that he has had to retrain his speech recognizer numerous times due to the computer crashing or his adapted profile not working well. On the other hand, the voice painter appreciates the reduced physical strain of speech-based interaction compared to using his trackball or a head tracker. He said:

> *"Doing it by voice really allows me to work much longer."*

### *3.2.4 Flexible Support for Reversing Actions*

His inability to undo and redo just a small portion of his strokes in Microsoft Paint has been a big problem for the voice painter. This is particularly problematic when the most recent stroke was the result of a long or complex process.

> *"Having to redo an entire stroke after a mistake at the very end is extremely painful... I also save at each stage so that I can figure out how I did the piece later on."*

Voice Draw was designed to support all of these goals, as described in the next section.

# 4. VOICEDRAW

We now present an overview of the VoiceDraw application followed by the details of its features. We describe how the design of each feature was influenced by the feedback obtained in our user-centered design process. We divide the features into two categories: those that are specific to stroke creation itself, and those that are more general features of the VoiceDraw application.

## 4.1 Application Overview

A screenshot of the VoiceDraw application is shown in Figure 1. The screen area consists of a status bar at the top of the screen (Figure 5), a canvas area below it, and a translucent help overlay showing the valid speech commands within the current context. Although non-speech vocalization can be used to operate VoiceDraw, speech commands are also provided as shortcuts for users that may want them.

In contrast to the sequence of discrete speech commands that must be issued using Dragon Dictate, a typical VoiceDraw stroke proceeds as follows. The brush head is moved to the desired canvas location by making a continuous vowel utterance based on the 2-D sound-map (Figure 2). The discrete sound "ch" sets the brush down. The user then vocalizes a vowel sequence corresponding to the desired path while varying the loudness to get the desired variation in stroke thickness. The same discrete sound "ch" lifts the brush off the canvas (Figure 4b).

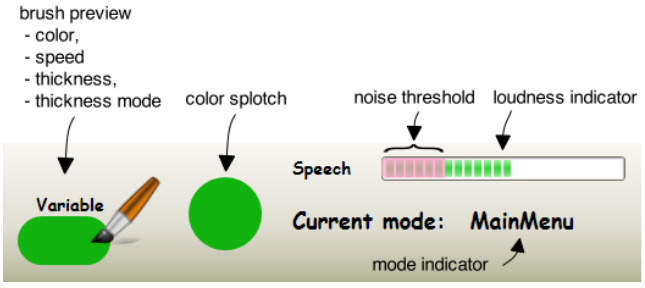
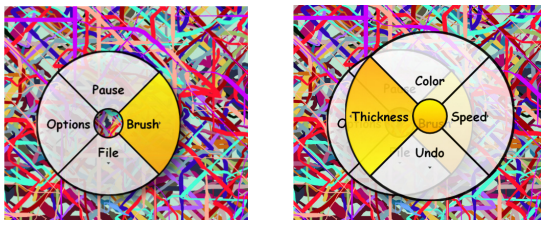



**Figure 5:** Status bar showing the brush preview, current color splotch, loudness indicator, and current mode indicator.

## 4.2 Stroke Creation

We decided not to overload the application with extraneous functions, but to focus on highly controllable stroke creation that is currently not possible using speech under existing systems.




(a) (b)

**Figure 6:** The vocal marking menu supports menu navigation using only non-speech vocalizations. The menu is invoked by issuing the discrete sound "ck". **(a)** The user finishes uttering "aww" (right), and is about to open the submenu by issuing the discrete sound "ch"; **(b)** the user finishes uttering "eee" (left) within the submenu and is about to execute the command by issuing the discrete sound "ch".

### *4.2.1 Brush Speed*

VoiceDraw provides four preset brush speed settings (very slow, slow, fast, very fast). Once a particular brush speed is chosen, the brush moves at that speed for as long as the user continues to vocalize. The current brush speed is reflected in the status bar as the length of the stroke in the brush preview (Figure 5).

In the first iteration of the prototype, the brush speed was mapped directly to loudness, allowing the user to dynamically vary the brush speed while the stroke thickness remained constant. However, based on the findings from our field investigations and feedback from the voice painter, the default mapping was changed to the current setup, where speed is based on the brush itself (e.g., slow brush, fast brush) and loudness controls stroke thickness. This gave the voice painter a greater sense of artistic control.

### *4.2.2 Stroke Thickness*

The current version of VoiceDraw maps loudness of the vocalization directly to the thickness of the stroke (see Figure 4b). The user can change the loudness mid-stroke, which will result in a stroke with varying thickness. There are four preset brush thickness settings (very thin, thin, thick, very thick) that define the maximum thickness achievable when loudness is the highest. There is also an ability to set the brush thickness to be fixed so that the entire stroke will be drawn with the same preset thickness, in which case the loudness can be used to control the brush speed. The currently selected brush thickness is reflected in the status bar and the thickness of the stroke in the brush preview (Figure 5).

Our initial version utilized the pitch of the vocalization to control thickness, but we discovered that users' change in pitch resulted in a change in the recognized vowel, even when users were sustaining the same vowel. Our voice painter also mentioned that he is not good at controlling his pitch, so we instead map loudness to stroke thickness. At the same time, a fixed-thickness option was also added to address the difficulty the voice painter experienced in sustaining a constant loudness for an extended period of time due to his limited lung capacity.

In the ideal case, one could simultaneously use loudness and pitch to vary two continuous parameters of the brush, but more work needs to be done on the underlying Vocal Joystick engine before this becomes feasible across all users. Even *if* such work is done, however, our findings here indicate that users may not be comfortable controlling multiple brush parameters in this way. Instead, it may be more effective to streamline the brush-selection process and use different brushes with different characteristics (e.g., fast brush, slow brush) along with one intuitive mapping, such as from loudness to stroke thickness.

## 4.3 Application Features

### *4.3.1 Verbal Commands*

Because the voice painter was more familiar with issuing commands through speech, we kept the ability to access all features using direct speech commands. The user can issue a number of commands to directly manipulate brush properties, such as "(very) slow/fast," "(very) thin/thick," "fixed/variable thickness," "draw," as well as changing to other states such as the color picker and erase mode.

In order to show what commands are valid in each mode, we implemented a floating help overlay with a list of commands that are always valid in the current mode, as well as non-speech vocalizations that are valid (Figure 1b). The voice painter found this to be particularly helpful in the beginning as we experimented with various sets of commands and wordings. In order to ensure that no part of the canvas is ever hidden behind the help overlay, the overlay automatically repositions itself to the side opposite the brush.

### *4.3.2 Vocal Marking Menu*

In response to the feedback about not breaking one's creative flow, we implemented a "vocal marking menu" (Figure 6) that allows the user to invoke menu commands without leaving VoiceDraw's non-speech vocalization modality. This design was modeled after the marking menu concept [20].

When the vocal marking menu is invoked by the discrete sound "ck", it presents the top-level menu items as pie wedges. An arrow next to their label indicates menu items that have submenus. The user can highlight a pie wedge containing the desired menu item by uttering the vowel sound corresponding to the direction of the menu item from the center of the menu. The menu item can be selected by making the discrete sound "ck", which executes a command or opens a submenu. The menu can be canceled by uttering the neutral vowel sound ("uh" as in "but") at the center of the vowel map (Figure 2).

### *4.3.3 Continuous, Incremental Undo*

VoiceDraw supports a novel undo feature in addition to the standard whole-stroke undo found in Microsoft Paint. The standard per-stroke undo works just like normal undo (executed by issuing an "undo stroke" speech command) and removes the most recently added stroke from the canvas. By contrast, the continuous undo feature (Figure 7) erases incrementally from the end of the stroke while the user continuously utters the "up" vowel sound ("aaa" as in "cat"). It is also possible to reverse the process by uttering the "down" vowel sound ("ooo" as in "boot").

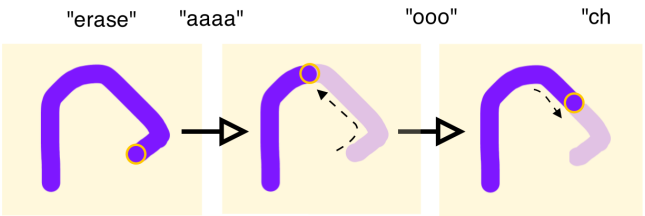
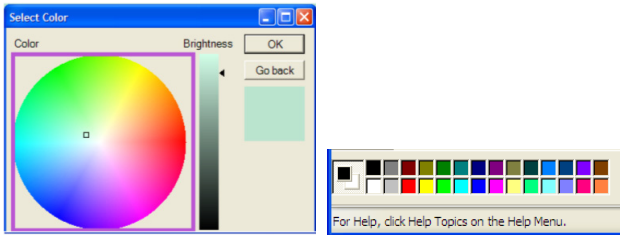



**Figure 7:** Continuous undo and redo are initiated by issuing the "erase" command, followed by continuous utterance of the "up" vowel ("aaa" as in "cat") to incrementally erase from the end of the stroke, and continuous utterance of the "down" vowel ("ooo" as in "boot") to incrementally restore the portion of the stroke. The change is committed with the discrete sound "ch".




(a) (b)

**Figure 8: (a)** The VoiceDraw color wheel (currently active) and the brightness slider, used with continuous 2-D vowel sounds. **(b)** Microsoft Paint's color palette with small color squares that the user must click with the pointer—a difficult task for anyone.

#### *4.3.4 Color Picker*

In the color dialog (Figure 8), we let the user specify the color and brightness separately using non-speech vocalizations. The circular color wheel can be navigated using the same 2-D vowel mapping that controls the brush (Figure 2), and the vertical brightness is adjusted by using just the "up" and "down" vowels. We could have used individual sliders for each of the RGB or HSV values, but that would have required the user to explicitly switch among each slider. The color wheel/brightness representation was preferred by the voice painter over other representations. The color wheel also allowed the voice painter to quickly locate a desired color instead of being restricted to a limited palette, and without having to move the cursor into a tiny palette square.

### 4.4 Implementation

The VoiceDraw application is written in C#, leveraging the Windows Presentation Foundation (WPF) and Extensible Application Markup Language (XAML) from the .NET Framework 3.0. The underlying Vocal Joystick engine is written in C++ as a dynamic link library (DLL); therefore a C# wrapper was created to facilitate integration with our front-end application.

The speech command recognition functionality is integrated into the VoiceDraw application through the use of Microsoft Speech API version 5.3. For our sessions with the voice painter, a desktop computer running Windows XP was used. An untrained user profile was used throughout the process for the speech recognizer, as it provided sufficient accuracy given the limited size of our command grammar. The total training time required for the Vocal Joystick engine is only 18 seconds, at 2 seconds per vowel (more detail on the training process can be found in [3]).

Due to the fact that we implemented the brush strokes as a collection of scalable vector graphics within the WPF framework, there were a set of benefits and shortcomings that we had to consider. The benefit of choosing this representation is that the resulting artwork is resolution-independent, and that operations such as the continuous undo and redo become possible. The downside to this approach is that it is not easy to replicate some of the features of bitmap-based painting programs such as Microsoft Paint, in particular the "flood fill" operation. We provide a compromise by enabling the production of freeform strokes with variable properties such as thickness, and providing extra thick brushes to efficiently fill in large areas.

## 5. OUTCOMES

### 5.1 Tasks

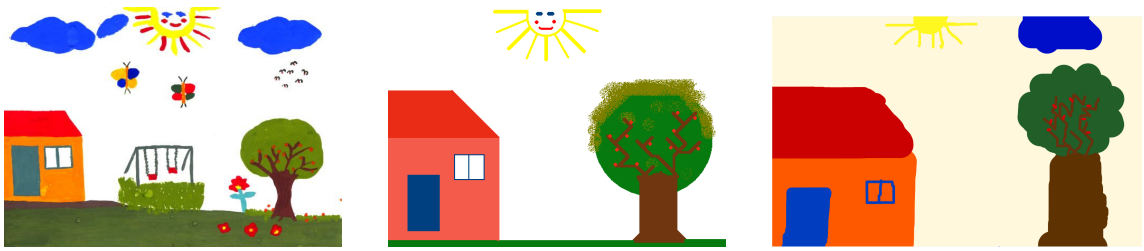
In order to assess the usability of VoiceDraw, we asked the voice painter to engage in two tasks. The first was a directed painting task in which we presented him with a target painting (Figure 9a) and asked him to create similar ones using his current tools and VoiceDraw. The second task was an undirected painting task where the voice painter created whatever he liked using VoiceDraw, while we observed.

The target drawing for the directed painting task[3] was chosen to explore how each tool can support the portrayal of a scene that consists of both rectilinear and non-rectilinear elements. We told the voice painter not to worry about replicating every exact detail, but to portray as much as he felt was necessary for a good representation of the original scene.

The results are shown in Figure 9 and Figure 10. For the directed painting task, VoiceDraw took about one hour, which was 15 minutes longer than Microsoft Paint. For the undirected painting, VoiceDraw took about 3 hours.

### 5.2 Reflections and Lessons Learned

The results from the painting tasks, especially the undirected painting task (Figure 10), are confirming of VoiceDraw's design. The detailed views of the paintings in Figure 10c and Figure 10d indicate the richer vocabulary of strokes supported by VoiceDraw over the voice painter's previous method. VoiceDraw's strokes exhibit unconstrained smooth paths and variable stroke thicknesses. The voice painter expressed great satisfaction in having been able to produce a piece that much more closely mimics his original inspiration, Jackson Pollock.



(a) (b) (c)

**Figure 9:** Results from the directed painting task, with **(a)** the target drawing taken from a school art class, **(b)** the replica using Microsoft Paint and Dragon Dictate, and **(c)** the replica using VoiceDraw. The replicas took approximately 45 minutes and 1 hour to create, respectively. Note that the voice painter had relatively little experience with VoiceDraw at this point.

[3] Taken from http://soe.ucdavis.edu/ms0506/180E/CheungC/

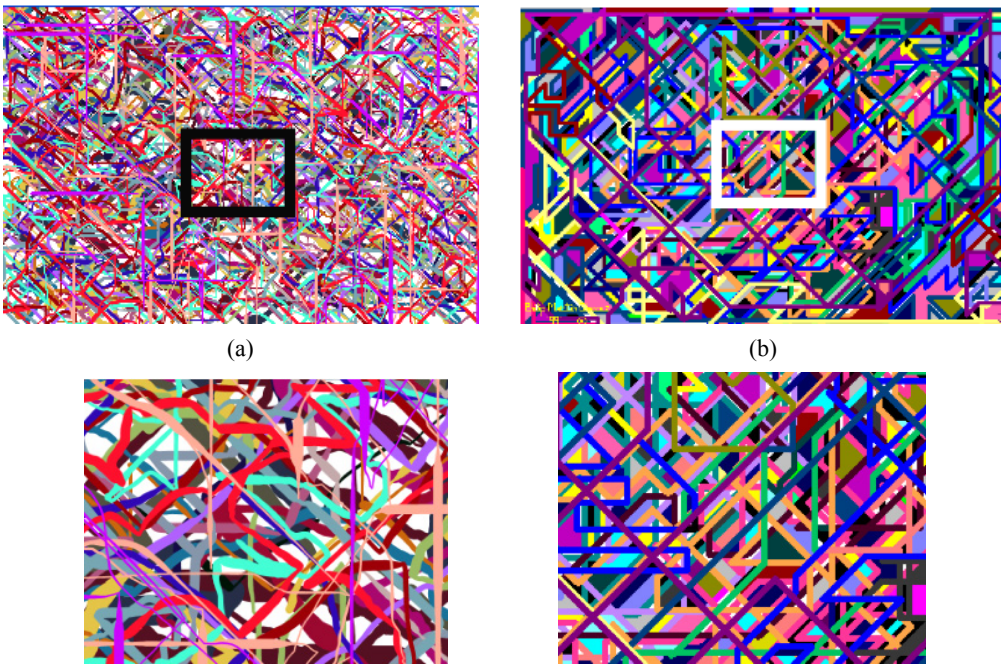



**Figure 10:** Results from the undirected painting task. **(a)** The result using VoiceDraw. **(b)** One of the earlier pieces that the voice painter had created using his previous tools from which he drew inspiration from memory to create **(a)**. The detailed view of a region on each painting (shown below each) reveals the difference in the stroke attributes with each tool. The VoiceDraw painting took 3 hours; the Microsoft Paint version was estimated to have taken 9 hours.

Another exciting result was that the Vocal Joystick version of the expressionist-style painting under the undirected task took almost a third as long as a similar painting using the previous tools, resulting in overall reduced fatigue. The voice painter mentioned that he wanted to take a sip of water more frequently when using VoiceDraw, but he did not complain of any significant vocal fatigue. The voice painter cited the ability to control all aspects of the program using voice as one of the biggest benefits:

> *"At one point, I realized I went for two hours straight, haven't hit the recliner, haven't used the arm ... it was a big plus not to have to use the trackball ... going in and out of drawing, changing the color, all that was verbal."*

It was also encouraging that the voice painter learned to reliably produce all of the vowel and discrete sounds after 30 minutes of guided training. He memorized the mappings of the vowels to the directions after two days (approximately 4 hours of VoiceDraw use). Although a longitudinal study is still in order, these observations give us a good indication of the learnability of our non-speech vocalization scheme. In particular, the subjective quality of the voice painter's new work is more fluid and curvilinear than his prior work, and arguably more in the spirit of his inspiration, Jackson Pollock. Most importantly, the voice painter himself thought so.

## 5.3 Remaining Challenges

Initially, the voice painter expressed a desire to have full program access in VoiceDraw without using any speech commands. However, after he started using the vocal marking menu, he decided he preferred speech commands (a reason we offer both). When asked for the reason behind his changing preferences, he stated that he felt more comfortable with the speech commands since he was more familiar with them from his previous setup, and also for their directness compared to the hierarchical organization of the vocal marking menu. It remains to be seen whether this preference would change given a longer trial period.

One unanticipated challenge was the limit of the voice painter's lung capacity and its impact on the types of strokes that he could generate. Because each of his utterances could last only one or two seconds, most of his strokes tended to be short segments, and did not fully take advantage of the ability to continuously vary the vowel quality for long smooth curves. Over time, however, as he became comfortable with the vowel mappings, he was able to incorporate longer vowel sweeps and create curves.

## 5.4 Observations From a Public Exhibit

VoiceDraw was showcased at a public exhibit at the University of Washington, where people from the general public, mainly children between the ages of 7 and 18 (along with their parents), were given the opportunity to try out the application. Each person was given few minutes of introduction to the vowel sounds and how to control the application, and was then allowed to use the system for a few minutes to produce their artwork. A total of 99 people tried VoiceDraw, most of them using the four-vowel mode (horizontal and vertical vowels only) due to limited training. The results (Figure 11) demonstrate that VoiceDraw can be used with very minimal training for creative self-expression.

# 6. CONCLUSION & FUTURE WORK

We have presented VoiceDraw, an application that investigates the use of voice as a viable modality for creating freeform drawings on a computer without manual interaction. Through five field investigations involving a self-described electronic voice painter, we identified a number of key design goals and solutions for upholding them. The results from this work suggest that VoiceDraw can significantly enhance the process of voice-based

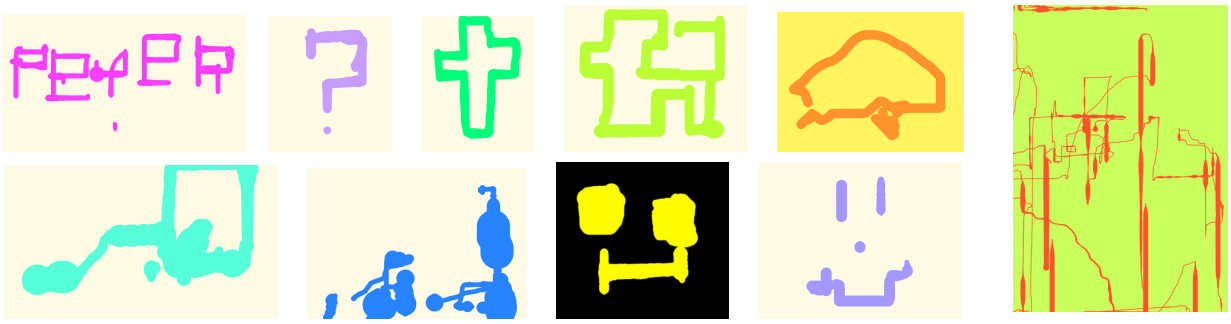


**Figure 11:** Drawings made by children trying out VoiceDraw as part of a public exhibit. All drawings were produced within several minutes after only a few minutes of training.

drawing, and that it opens up a new realm of creative expression that was previously not available to people with motor impairments. VoiceDraw creates qualitatively different art from our voice painter's existing tools due to its provision of continuous, fluid control through manipulation of vocal parameters. The experience gained through building the VoiceDraw application can inform future design of applications optimized for voice.

We plan to continue working with the voice painter to explore more ways in which VoiceDraw can enhance the experience of drawing using voice. Some of these will be to provide features for filled polygons, and canvas control such as panning and zooming.

We also plan to take the lessons learned from the development of VoiceDraw to begin generalizing on a number of interaction techniques and design principles that can form a basis for voice-based applications in general.

## 7. ACKNOWLEDGEMENTS

The authors thank Jeff Bilmes, Jonathan Malkin, Xiao Li, David Levy, Shaun Kane, Scott Saponas, Jon Froehlich, Kayur Patel, Yang Li, Tim Kao and Eric Maruta for their support, and Philip Chavez for working with VoiceDraw. This work is supported in part by the National Science Foundation under grant IIS-0326382. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## 8. REFERENCES

[1] AMFPA - Association of Mouth and Foot Painting Artists of the World. http://www.amfpa.com/.

[2] Bergman, E. and Johnson, E. (1995) *Toward Accessible Human-Computer Interaction*. Ablex Publishing, 87-113.

[3] Bilmes, J.A., Li, X., Malkin, J., Kilanski, K., Wright, R., Kirchhoff, K., Subramanya, A., Harada, S., Landay, J.A., Dowden, P. and Chizeck, H. (2005) The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments. *Proc. HLT/EMNLP '05*. Morristown, NJ: Association for Computational Linguistics, 995-1002.

[4] Dai, L., Goldman, R., Sears, A. and Lozier, J. (2004) Speech-based cursor control: a study of grid-based solutions. *Proc. ASSETS '04*. New York: ACM Press, 94-101.

[5] Dawe, M. (2006) Desperately seeking simplicity: how young adults with cognitive disabilities and their families adopt assistive technologies. *Proc. CHI '06*. New York: ACM Press, 1143-1152.

[6] de Mauro, C., Gori, M., Maggini, M. and Martinelli, E. (2001) Easy access to graphical interfaces by voice mouse. Technical report, Università di Siena. Available from the author at: maggini@dii.unisi.it.

[7] Fuller, R.B. (1973) *Nine Chains to the Moon*. Cape.

[8] Guimbretière, F. and Winograd, T. (2000) Flowmenu: combining command, text, and data entry. *Proc. UIST '00*. New York: ACM Press, 213-216.

[9] Harada, S., Landay, J.A., Malkin, J., Li, X. and Bilmes, J.A. (2006) The Vocal Joystick: Evaluation of voice-based cursor control techniques. *Proc. ASSETS '06.* New York: ACM Press, 197-204.

[10] Hornof, A.J. and Cavender, A. (2005) EyeDraw: enabling children with severe motor impairments to draw with their eyes. *Proc. CHI '05*. New York: ACM Press, 161-170.

[11] Igarashi, T. and Hughes, J.F. (2001) Voice as sound: Using non-verbal voice input for interactive control. *Proc. UIST '01*. New York: ACM Press, 155-156.

[12] Keates, S., Hwang, F., Langdon, P., Clarkson, P.J. and Robinson, P. (2002) Cursor measures for motion-impaired computer users. *Proc. ASSETS '02*. New York: ACM Press, 135-142.

[13] Kjeldsen, R. (2006) Improvements in vision-based pointer control. *Proc. ASSETS '06*. New York: ACM Press, 189-196.

[14] Levin, G. and Lieberman, Z. (2004) In-situ speech visualization in real-time interactive installation and performance. *Proc. NPAR '04*. New York: ACM Press, 7-14.

[15] Malkewitz, R. (1998) Head pointing and speech control as a hands-free interface to desktop computing. *Proc. ASSETS '98.* New York: ACM Press, 182-188.

[16] Manaris, B., McCauley, R. and MacGyvers, V. (2001) An intelligent interface for keyboard and mouse control – providing full access to PC functionality via speech. *Proc. FLAIRS '01.* AAAI Press, 182-188.

[17] Mihara, Y., Shibayama, E. and Takahashi, S. (2005) The migratory cursor: Accurate speech-based cursor movement by moving multiple ghost cursors using non-verbal vocalizations. *Proc. ASSETS '05*. New York: ACM Press, 76-83.

[18] Phillips, B. and Zhao, H. (1993) Predictors of assistive technology abandonment. *Assistive Technology*, 5(1):36-45.

[19] Sporka, A.J., Kurniawan, S.H. and Slavík, P. (2006) Acoustic control of mouse pointer. *Universal Access in the Information Society*, 4(3):237-245.

[20] Tapia, M.A. and Kurtenbach, G. (1995) Some design refinements and principles on the appearance and behavior of marking menus. *Proc. UIST '95*. New York: ACM Press, 189-195.

[21] Voice painter. http://www.voicepainter.com/index.php.

[22] Warren, B. (1997) Ch. 13 in *Quality of Life for People with Disabilities (2nd ed.)*. Nelson Thornes, 270-291.