CoSke - An Exploration in Collaborative Sketching

Jessica David  
College of Engineering  
and Computer Science  
University of Evansville  
Evansville, Indiana 47714–1506  
Email: jd141@evansville.edu

Brian Eoff  
Department of Computer Science  
and Engineering  
Texas A&M University  
College Station, Texas 77843–3112  
Email: bde@cse.tamu.edu

Dr. Tracy Hammond  
Department of Computer Science  
and Engineering  
Texas A&M University  
College Station, Texas 77843–3112  
Email: hammond@cse.tamu.edu

Abstract—Digital collaborative sketching has been possible and freely available for years, but little progress has been made in developing these programs due to their lack of sophistication. This paper focuses on the development of a program entitled CoSke, a server application that works with a variety of clients and accepts multiple inputs to allow drawing on a shared canvas. A user study was performed to analyze how using these digital technologies changes when users are collaborating in the same room versus being isolated in separate locations, as well as the benefits of using these technologies over traditional pen and paper methods and vice versa. Users in the study sketched collectively while being observed by proctors in three unique setups: in a group on paper, in a group on computers, and isolated on computers. Points of communication such as hand gestures, eye contact, facial expressions, user emotions, and user contribution were recorded by the proctors and examined after the user studies were completed. The results from these metrics, as well as comments from user surveys, will be used to provide insight as to how developers can overcome the limitations of current collaborative sketching software programs, and most importantly how to envision and build a system that will truly provide for the capabilities and natural flow of face-to-face human sketching communication.

I. INTRODUCTION

Collaboration proves to be a helpful tool when used to inspire creativity and promote idea generation. This can be translated into the digital world in the form of several existing pieces of software that promote collaboration both locally and from a distance, with a subset of these programs emphasizing sketching and drawing with others on a shared canvas. While freely accessible and useful for general purposes, digital collaborative sketching has not been deeply investigated in terms of its effect on how users draw or how users contribute to a final image, and thus most available programs are not as robust as they potentially could be. They are also limited in their availability to different software and hardware platforms.

Our goal is to develop a new computer application and study how people draw in groups, especially when face-to-face communication is limited and users are drawing from remote locations. To do this, an application was developed, titled CoSke (short for Collaborative Sketching), that provides the basic features of a digital collaborative sketching. After the software was developed, a user study was conducted to discover what users do in terms of communication when they have the opportunity to draw together. Each user was asked to answer a few short questions about his or her experiences and compare various methods of sketching, such as with paper and pencil versus digital drawing. The results of the user study were used to determine what functionality could be added to future versions of CoSke to create a more natural environment for collaborative sketching. The ultimate goal would be to create a client and server that would extend across any operating system or piece of hardware (whether it is a netbook, mobile device, Tablet PC, etc.) that would work just as effectively if the users are in the same location or remotely working with one another.

II. PREVIOUS WORK

Collaborative sketching programs are not new to the field of human-computer interaction. Several versions have been developed over the web or for mobile devices. Most of these web versions use Flash, such as General Electric’s Imagination Cubed [1] or Sketchr [2]. While fully functional in their capabilities, they are rather simplistic and crude in their drawing techniques. They seem like they are mostly for fun, and while GE’s focuses on idea generation, it lacks the sophistication that would make researchers consider it more seriously.

Some versions of collaborative sketching programs that introduce a level of sophistication are previously developed programs for handheld devices. One method of collaboration focuses on using multiple PDAs to a PC [3]. At the time of this publication, PDAs were a popular device that were smaller and more convenient to carry around than laptops, and the authors wanted to create a way for users to have mouse and keyboard inputs to a PC without having them leave their seats. They created a single display groupware application called PebblesDraw which allowed users to share one display and draw on it at the same time. Each user is assigned a different shaped cursor so that strokes can be associated with a certain user; this option was chosen over color for multiple reasons including inabilities to switch colors and possible detriments to colorblind users. Also, operations such as cut and copy are only performed on the desired user’s selected subjects. It also implements gesture recognition and several menus to aid in the collaborative drawing process.
Another example of collaboration on handheld devices is MCSketcher, developed at the University of Chile, which is "a system that enables face-to-face collaborative design based on sketches using handheld devices equipped for spontaneous wireless peer-to-peer networking" and mostly targeted for those working in the field (such as gardeners or building designers) to do collaborative sketch work [4]. Everyone starts with a synchronized blank page to work on, and a user with a camera can take a picture to use as the background (which is distributed among all users). One feature the program has is the ability to create design spots, which can be used to link to other pages where an area of the image can be described. The design can also be viewed as a whole in a document tree environment. Users can also work on their own or in a session with other users. The software offers gesture recognition for features such as selecting and resizing items, as well as basic navigation gestures.

Both of these programs provide features to be considered when creating a collaborative sketching environment, but both are slightly inconvenient to use as the sketching areas are small due to the size restraints of handheld devices. Because of this, it limits the amount that one can do with a sketch. These programs are also limited to their respective handheld devices, and it would be ideal to have a program that could be used on various hardware platforms. Also, these methods investigate using their respective collaborative implementations when the users are in the same room. A different dimension of collaborative sketching comes in when the users are in remote locations.

There are other pieces of software that have been developed to encourage collaboration over a distance and have proven to be useful communication techniques. One example is a real-time collaborative software engineering (CSE) program that has been shown to benefit programmer communication and resolve conflicts between developers [5]. Since there tends to be many teams working on one project, it would make sense to have an architecture that would help facilitate real-time programmer communication. The architecture is called CAISE, and it supports real-time development of projects and seamlessly integrates tool support to raise levels of communication within current software engineering practices. The designers are synchronized in real time using the concept of continuous integration, and users can either work where their progress is shown publically or in private. Some of the basic principles of collaboration in this environment can be carried over to that of collaborative sketching, such as allowing either a public or private sketching feature.

The benefits of collaboration can be seen outside of technology in the field of idea generation. Remko van der Lugt of the Delft University of Technology performed a study to investigate whether functions of sketching in design activities are helpful in idea generation meetings [6]. He acknowledged that both informal design group and brainstorming meetings are intended for problem solving, thus sketching functions found for design groups can also be applicable for brainstorming meetings in product design. These functions are supporting a re-interpretive cycle in the individual thinking process, supporting re-interpretation of each other’s ideas in group activity and enhancing the access to earlier ideas. At the time of publication, aspects of group sketching were not thoroughly investigated but are being furthered through research in the field of Computer Supported Collaborative Work (CSCW) and HCI, as they are looking into allowing designers to work together from remote locations, and this is one aspect that is analyzed in this project.

Along with these ideas, it is becoming more prevalent in HCI to discover how technologies can be used in creative platforms. A recent workshop at CHI 2009 encouraged people from a variety of creative backgrounds to come together to discuss how they are becoming increasingly comfortable with emerging technologies around them, from music to photography to writing [7]. The goal of their workshop was discuss tools and methods to help develop novel creative tools. By creating an ideal collaborative sketching client, it could then also be used not only professional setups but also in creative environments with artists working together on more complex, detailed image creation.

III. CoSke

A. Previously Developed Code

Software development began by studying the code of Dr. Aaron Adler from MIT who had been working on an unpublished collaborative sketching program with some of the same features that were desired for CoSke. By gaining a better understanding of an already existing collaborative sketching program that utilized a client/server architecture, CoSke development would be easier to approach. Adler’s code had a server written Java and client in C#. Each took a properties file as an argument that contained information such as the server name (e.g., an IP address or computer name) and several port numbers for communication. The server would initialize these ports and listen for a client to connect. When a client connected with an identical properties file, general handshaking would occur so the server would recognize that a new client had been added, and the client would listen for any messages coming back from the server. Events were sent back and forth between the client and server in the form of strings and parsed accordingly when received, calling appropriate functions for different operations. This communication technique is described in more detail in the next section in terms of CoSke’s implementation.

B. CoSke Development

Before CoSke development began, the decision was made to rewrite the client in Java to make the software more cross compatible. The preexisting C# code only worked with Tablet computers and used the Microsoft.Ink package, which is Microsoft’s namespace that implements digital ink on Tablet PCs [8]. For the purposes of this project, we focused more strongly on the client software, and the server code was built upon Adler’s preexisting code.
Both the client and server begin by displaying a GUI window to the user, asking them to select a properties file which contains the server name and port numbers for communication, which can be seen in Figure 1. From here, users can view the current properties they have loaded or set properties manually.

On the server side, a user can change the size of the image to be drawn as well as the initial background color. When all the settings have been finalized, the user can begin to run the server, and the GUI is used to output debugging statements such as information about connecting clients. The client, on the other hand, has the ability to change the user name of whoever is drawing, as well as viewing and setting the connection settings. When the client is ready to connect, a similar handshaking process occurs as in the original code and sketching is ready to occur.

To make using the stylus of a Tablet computer easier with Java, the JPen library was used which allowed Java to access pen input and use event/listener architecture for various stylus events (e.g., pen down, pen button presses) [9]. The layout of the client is shown on the next page in Figure 2. The drawing area utilizes a panel with all of the necessary JPen event handlers for stylus or mouse events. There is a text area which, for the purposes of this study, only gives feedback for which color the current user is drawing with. The area next to it controls general drawing functions. The user is allowed to change their brush color and size, and they can clear or save the entire image.

To draw a stroke, a user can either click with their mouse or press the stylus to the screen. This triggers a pen level event and the current location of the stylus, and the pressure are recorded. As the user drags the pen/mouse, points are frequently registered, and short line segments are drawn from the previous point to the current location. Since the stylus is so sensitive, a threshold was added to prevent erroneous strokes; the pressure value returned is between zero and one and a value of about 0.18 seemed to create the smoothest drawing atmosphere. As each portion of a stroke is completed, a message is sent to the server containing information such as the brush size and color and then immediately drawn to the screen. When the whole stroke is completed and the pen is raised (or mouse button released), information about the whole stroke is sent to the server and a repaint is called within the whole stroke’s bounding box on each client. Each client keeps track of the partial and completed strokes in a collection for future repaint calls to the panel.

The communication protocol was based on the older version, and future versions will be standardized such that anyone wishing to create their own client can use the same communication technique. In this version of the code, there were four different types of messages that needed to be sent to all clients: processing a new line or stroke, clearing the screen, and changing color by a user. Each message begins with an identifier as to which operation needs to be performed, followed by parameters that are necessary for the clients to perform the tasks correctly. Table I above shows the format for each of these message strings. Once the client has received the message and processed the string, it would call the appropriate function and either draw the new line, repaint the whole stroke, clear the screen, or display the user’s new color.

### IV. Experiment

Once CoSke development was completed, a user study was performed to analyze several aspects of collaborative sketching. The user study had twelve participants divided into four groups of three. The participants were undergraduate students in Computer Science or Engineering Research Experience for Undergraduate (REU) programs at Texas A&M University and had minimal to moderate Tablet experience. The participants were asked to complete three collaborative drawing tasks, two digitally and one analog. The digital drawing was done on Wacom Cintiq monitors and the analog drawing used a single sheet of letter paper and colored pencils (with and without erasers). Each study took about an hour and included a five minute overview for the digital drawing system, ten minutes

<table>
<thead>
<tr>
<th>Command</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>&lt;line strokeID=&quot;id&quot; author=&quot;author&quot;&lt;br&gt;color=&quot;r,g,b&quot; brushSize=&quot;size&quot; x1=&quot;val&quot; y1=&quot;val&quot; x2=&quot;val&quot; y2=&quot;val&quot; /&gt;</td>
</tr>
<tr>
<td>Stroke</td>
<td>&lt;stroke strokeID=&quot;id&quot; author=&quot;author&quot;&lt;br&gt;color=&quot;r,g,b&quot; brushSize=&quot;size&quot; &lt;polyline&gt;nPoints;x1,y1;x2,y2;...;xN,yN;&lt;/polyline&gt; /&gt;</td>
</tr>
<tr>
<td>Clear</td>
<td>&lt;clear userName=&quot;author&quot; /&gt;</td>
</tr>
<tr>
<td>Color Change</td>
<td>&lt;colorChange username=&quot;author&quot;&lt;br&gt;color=&quot;r,g,b&quot; /&gt;</td>
</tr>
</tbody>
</table>
Fig. 2. The Client GUI

for each task, and fifteen to twenty minutes to fill out a user survey.

For the drawing exercises, the users were asked to draw in three different scenarios:

- Drawing in the same room on pencil and paper (analog)
- Drawing in the same room digitally using CoSke
- Drawing in different rooms digitally using CoSke

Each exercise had a different drawing task associated with it, each with elements that users could draw together or separately while in a collaborative environment:

- Draw a scene of a house and its surroundings
- Draw a dragon
- Draw a still life image consisting of a bowl of fruit

The users were encouraged to draw as many details as they saw fit. The order of the setups and pairings with the scenes were chosen at random. The digital images were saved after each exercise, and a program was used to capture the image as it was being drawn for further inspection after the exercise was finished. The size of the shared canvas was chosen to be approximately the same size as a sheet of paper (1200x1000 pixels). Each exercise was observed by a proctor who would mark when users used hand gestures, made eye contact with other users, evoked different emotions or facial expressions, and approximately how much they contributed to the image.

V. RESULTS

A. Quantitative

Results from the proctors’ notes were analyzed quantitatively to determine what users did differently in each environment. The results from these notes were used to conceive different features that could be added to enhance the digital remote drawing experience.

When users were isolated, they used a very limited amount of hand gestures and their hands were kept to themselves. Even when they were in the same room, the users never inadvertently pointed to a feature on the screen as if assuming others could see. However, when using pen and paper, hand gestures could be used to point out features, comment on a certain area, ask for writing utensils, and so forth. Interestingly, the users never tried to find a digital equivalent to making a gesture to a feature (to potentially point out a flaw, a feature to add, etc.) like they would when drawing in the analog setting.

A similar result was determined about the amount of eye contact users made with each other (with the exception of the isolated scenario, in which no eye contact could be possible). In the same room, users rarely turned around to try and address each other. However, when working together on paper, there were multiple instances where one would acknowledge another by making eye contact. Eye contact and gesturing were also used in tandem to make comments about features, as well.

Emotionally, the users tended to be very relaxed and happy in each scenario. While there were instances of frustration in
the isolated situations, they stayed focused and would laugh while observing other’s drawings. This related to their facial expressions, ranged from being stern when concentrated to smiling when enjoying what was happening in the developing image. The users also remained relatively relaxed in terms of divvying out tasks, and usually a leader would naturally come in to move things along or all would work together effectively.

Users contributed in different ways depending on their scenario. In the paper method, users tended to ask permission more to draw elements than in the digital environment; digitally, when in the same room, they were more likely to announce what they were working on and then proceed to draw it. In isolation, users tended to go along with what other users were doing or developing off of what was being created around them. They would also tend to talk at the screen when someone would draw what they were about to or pause to analyze what was going on. In general, though, users seemed to contribute an equal portion of each drawing and it did not seem to affect them whether they were isolated or not.

B. Qualitative

In the surveys, the users were asked to rank the drawing methods in order of preference. Drawing digitally in the same room proved to be the most preferred method, followed by pencil and paper and then isolated. The reasons the users gave emphasized the differences between the digital and analog methods, as well as problems they had with communication when separated.

Almost all of the users felt that actual paper was more intuitive, i.e., since they were more used to drawing in that method it felt more natural than the digital method. However, they felt they had more power over their sketch in a digital environment. Pencil and paper required them to take turns whereas digital let them do things freely - while they shared a common canvas, they were able to have their own set of tools and the proper canvas orientation to draw effectively. Some of the frustrations that users had when isolated were similar to that of having a small area to work on. For example, someone would either begin drawing in the same location or draw what another user was about to draw, and these situations also happened during analog drawing. Still, a majority of the users felt that, in the digital environment, they had no need to share and it made it a much more effective experience, even if communication was limited.

When comparing the digital methods, all of the users focused on the communication element as being the biggest hurdle. By not knowing where a user could draw next, users tended to draw over another’s work. One group tried to overcome this hurdle by attempting to write messages on the screen to the others to try and communicate an action. Despite most users wanting some way to communicate, one user did comment that it did not matter because they tended to work on separate elements anyway, therefore making communication not as critical.

C. Analyzing Results

Users definitely showed a preference to methods in which they could communicate with one another. However, even though the preferred technique was to be able to communicate in the same room, this did not necessarily mean that the images were any more detailed than in the isolated situation. In some cases, communication even hindered image development because users could not decide who should take initiative on constructing individual elements of the drawing. One user
commented that they worked more chaotically in the same room because they would argue over who would draw a certain characteristic of the picture. From this observation, an interesting question arises: how does collaborative sketching in the same room impede progression of the sketch and idea generation, and do users only think that being able to communicate is a crucial part of the process?

This problem also exists in the analog setting; it was observed that fighting over the same space often caused more problems than not. From this, another interesting aspect to consider appears: how willing are others to impede another’s work when the element of communication is removed? Users would normally erase their own work when isolated or be hesitant to continue another’s work because of a certain dimension of social etiquette; no one wanted to be rude and interrupt what another user was doing. By doing this, users were able to watch a sketch evolve around what they were doing, as if the program was designed to draw with them instead of against them. This led to very interesting, detailed images that were on par with the drawings done when the users sketched in the same room. The element of communication seems to assure users that their sketching is approved by the other users, and this is an interesting aspect to the psychological aspect of how a user draws when collaborating with others.

VI. Future Work

From the results of the user study, there are a few features that could be added to CoSke to make it a more intuitive experience for the user. There needs to be a way to digitally point or gesture with the pen, as users tended to do this more than often in the pencil and paper method, either with hand gestures or making eye contact with another user. By having the cursor propagate across all of the screens or with some sort of indicator, as well as having the color represented for the user by the cursor, an extra layer of communication is provided which gives a similar feeling to drawing in a similar location. Another feature could be adding an annotation mode that would allow the users to draw on a transparent layer over the image to write or type out notes, thereby allowing them to communicate information to other users without worrying about writing over previous work.

Other features that would create a more valuable experience for the users would include a stroke identification system. For example, if you wanted to know who drew a certain line, you could hover over it and the name of the stroke’s author would appear by the cursor. Another feature to add would be a voice chat system. Since users felt they needed a form of communication, this would be most ideal for a sketching environment as a text chat would require the users to stop sketching in order to correspond with one another. The next step after implementing this feature would be to run a user study where, in one situation, the users are allowed to speak to one another while in remote locations and another where they are not permitted to see how it affects their collaborative techniques. An interesting addition to this study would have another group of users watch the video of sketches being created and observing resulting images and ask them if they are able to tell if they were created when communication was limited or not.

VII. Conclusion

The results of this project are twofold: we wanted to know how people collaboratively sketched in groups, and we wanted to use the results from the user studies to make a program that takes these drawing techniques and enhances them. The current version of CoSke is intentionally limited to allow room for development due to the results of the user studies. By observing how people draw in groups, we are able to take this information into account when creating the ultimate collaborative sketching program. New features will be more tailored to the results of the first round of user studies, and by conducting more studies and developing new versions, we can create an environment that feels like pencil and paper and behaves as if the users are drawing as one collective mind.

Acknowledgments

Jessica David would like to thank Dr. Tracy Hammond and Brian Eoff for their assistance this summer, as well as the other graduate students in the Sketch Recognition Lab at Texas A&M University. She would also like to thank Dr. Aaron Adler from MIT for his code contribution and the users who participated in the user study.

References