

Stitching: Pen Gestures that Span Multiple Displays

Ken Hinckley, Gonzalo Ramos, Francois Guimbretiere, Patrick Baudisch, Marc Smith
Microsoft Research, One Microsoft Way, Redmond, WA 98052

{kenh, baudisch, masmith}@microsoft.com, bonzo@dgp.toronto.edu, francois@cs.umd.edu

ABSTRACT

Stitching is a new interaction technique that allows users to combine pen-operated mobile devices with wireless networking by using pen gestures that span multiple displays. To stitch, a user starts moving the pen on one screen, crosses over the bezel, and finishes the stroke on the screen of a nearby device. Properties of each half of the user's pen stroke are observed by the two separate devices, synchronized via wireless network communication, and recognized as a unitary act performed by one user, thus binding together the devices.

We identify the general requirements of stitching and describe a prototype photo sharing application that uses stitching to allow users to copy images from one tablet to another that is nearby, expand an image across multiple screens, establish a persistent shared workspace, or use one tablet to present images that a user selects from another tablet. Preliminary usability testing suggests that users find it compelling to have a straightforward means to combine the resources of multiple mobile devices. We also discuss design issues that arise from proxemics, the sociological implications of users collaborating in close quarters.

Author Keywords

pen computing, mobile devices, distributed systems, proxemics, gestural input, synchronous gestures

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Input

INTRODUCTION

Much recent discussion has focused on the promise of the wireless internet, yet there has been relatively little work on techniques that help users of mobile devices to collaborate with others and to share information with other persons [17][18][22]. For example, when attempting to copy a file between mobile devices, it remains difficult for users to *name* a device to connect to, specify *how* to connect, or indicate *what information* to share.

While several previous systems foster collaboration between ubiquitous devices [15][19][21][23], they may require special hardware such as overhead cameras or unique ID tags on each device, or they may require manual

entry of network addresses and the geometry of displays. We propose *stitching* as a new interaction metaphor that uses commonplace pen input capabilities to establish and manage serendipitous connections between pen-operated mobile devices. A *stitching gesture* consists of a continuous pen motion that starts on one mobile device, continues past the bezel of the screen, and ends on the screen of another device (Fig. 1). Such gestures serve as the basis for a flexible architecture that allows users to dynamically bind together mobile devices. Stitching can be implemented on a variety of pen-operated devices, does not conflict with existing uses for the pen, and provides a versatile framework that can accommodate future extensions.

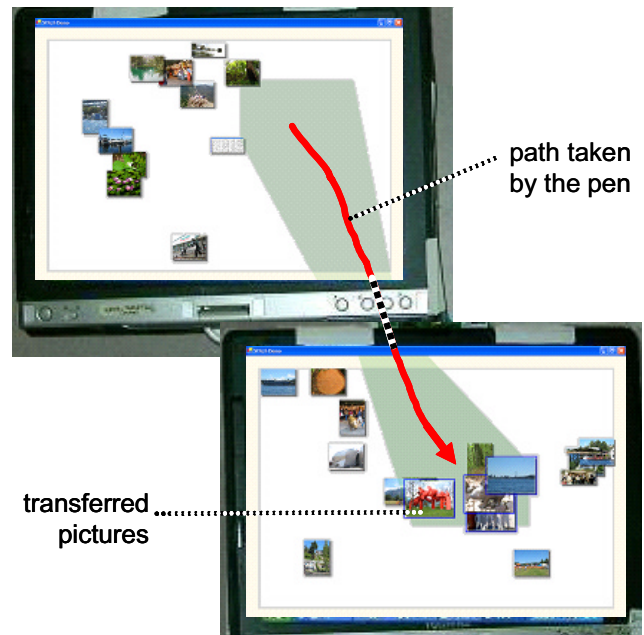


Fig. 1 Here, a user gives some photos to another user by dragging them from the top tablet to the bottom tablet.

From the user's perspective stitching seems like a single cognitive chunk [3], but the gesture actually specifies a number of properties of the connection:

- It selects which devices participate in the connection.
- We can phrase together selection of operands (e.g., a file to copy) and commands (how to combine the devices) with the stitching gesture itself.
- By fitting a line to the pen coordinates as they leave one device and enter another, we can calculate the approximate spatial relationship between the two devices. This allows us to place graphics or provide feedback that appears to span the displays (Fig. 1).

Submitted to CHI 2004

We describe a prototype photo sharing application for the Tablet PC that supports operations such as copying images from one tablet to another that is nearby, establishing a persistent shared workspace for collaboration, expanding an image across multiple screens, or using one tablet to display a slideshow of images that a user selects from another tablet. Usability testing suggests that users readily grasp stitching, and find it compelling to have a straightforward means to perform cross-device operations. We also observed that sociological issues of co-located collaboration raise several design issues. We found it is important to support a range of device configurations, from intimate combinations of devices in direct contact with one another, to sharing information while maintaining social distance. The latter requires support of stitching between devices that are nearby one another, but not touching.

RELATED WORK

Pick and Drop [18] enables users to pick (copy) an item from one screen and drop (paste) it onto the screen of another nearby device. Stitching also supports copying items between co-located devices, but unlike Pick and Drop, our technique does not require a unique ID on the pen; rather, stitching uses the timing and dynamics of the gesture to pair up pen strokes observed by the devices.

The Pebbles project [17] explores multi-machine user interfaces that spread computing and user interfaces across multiple devices. Stitching contributes a versatile interaction paradigm that could be used to dynamically form some of the configurations envisioned by Pebbles.

ConnecTables [22] and Smart-Its Friends [14] both form connections between mobile devices. Smart-Its Friends connects two devices if they are held together and shaken. ConnecTables are wheeled tables with mounted LCD displays that can be rolled together so that the top edges of two LCD's meet, forming a collaborative workspace. Each LCD senses the presence of the other one using radio-frequency identification (RFID) tags. These systems require special hardware, and support only one type of connection.

Synchronous gestures [13] are distributed patterns of activity that take on a new meaning when they occur together in time, or in a specific sequence in time. For example, the synchronous gesture of bumping two mobile devices together can connect the devices in various ways [12][13]. Stitching represents a new type of synchronous gesture. Bumping provides a means to connect devices that lack pen input, but stitching is a natural extension of pen computing that can leverage pen techniques for selecting commands, operands, and indirect objects [5][10][16].

Co-located collaboration involves users working in close physical proximity to one another. *Proxemics* is the study of how people use the invisible bubble of space that surrounds an individual [11]. Appropriate social distances vary between cultures, but violations of intimate personal space (within approximately 45cm) may produce discomfort or physical withdrawal to maintain a

comfortable social distance [7][8]; touching is particularly unwelcome in non-contact cultures [11]. Our experience with stitching suggests that the proxemics of co-located collaboration may yield critical insights for designers.

THE REQUIREMENTS OF STITCHING

We define a stitch as a gesture, spanning two or more devices, which establishes a communication infrastructure or otherwise combines the resources of multiple computers. In order to provide a flexible and potentially extensible facility that can support a number of different ways of combining devices, stitching addresses the following central design questions:

- 1. How is a connection established?** A user must name the devices that are involved in a multi-machine operation, and the system needs to provide feedback to the user(s) of those devices that a connection has been established.
- 2. What type of connection is required?** The user needs to be able to choose among several possible ways to combine the devices. Does the user want to copy a file from one device to another? Establish a persistent shared workspace for collaboration? Expand an image across multiple screens? These all represent *multi-device commands* that transcend the barriers between devices.
- 3. What information is shared?** Multi-device commands may require operands, such as which file to copy to another computer. Users need mechanisms to select one or more objects as part of a stitching gesture.
- 4. How do users share physical space?** Proxemics suggests that the arrangement of spaces can influence communication; as Hall writes, "what is desirable is flexibility... so that there is a variety of spaces, and people can be involved or not, as the occasion and mood demand" ([11], p 110). Interaction techniques that form impromptu associations between mobile devices should likewise support the range from users who know each other well and want to work closely together, to users who are strangers and want to exchange files while keeping their distance.
- 5. What is the spatial relationship between the devices?** Several previous systems support features, such as combining the screens of two devices, that require knowledge of where one display is relative to another [15] [19]. Stitching uses the information provided by the pen to infer the spatial relationship between devices. This also allows us to provide graphical feedback for multi-device operations that appears to span devices, as seen in Fig. 1.
- 6. How do stitching gestures coexist with traditional pen interactions?** Stitching gestures must coexist with existing uses for the pen including widget interactions, inking, character entry, and naturally occurring human-human communicative gestures (such as waving the pen near the device while discussing the contents of the screen).

THE MECHANICS OF STITCHING

The above design questions suggest that stitching represents a new class of interaction techniques that could be implemented in a variety of ways. In the remainder of this paper, we discuss the general concept of stitching in

reference to a proof-of-concept photo sharing application called *StitchMaster*. With digital photography becoming widespread, sharing photos with others is a task of interest to many persons. Also, many of the semantics that we wanted to explore with stitching, such as expanding an image across multiple screens or copying objects from one screen to another, represent useful and compelling operations for digital photographs. To begin, each user launches *StitchMaster* on his own tablet, which displays that user's photo collection as a set of thumbnail images.

1. Establishing a Connection

Stitching requires devices that can sense the same pen; the user names the devices to connect by moving the pen across them. Since there is a natural order implied by the gesture, stitching also establishes which machine is the sender of information, and which machine is the receiver. Some connection techniques are inherently bidirectional [9][14][22] and do not naturally provide this information.

Each participating device sends its pen events to a *stitching server*, which may be hosted on a machine in the environment to offload computation from the mobile devices. The stitching server synchronizes time between the devices [13] and looks for matching pen traces; when a match is found, the server sends a stitching event that informs the two devices of each other's network address. Each participating device must know the network address of the server, but this is the only address needed to bootstrap the system. In the future, we may instead find this address via service lookup mechanisms, or by using wireless signal strengths to locate a nearby server [1].

Stitching Recognition

The stitching server recognizes a stitch by looking at the patterns of pen activity from each pair of participating devices. We define an *envelope* as the time interval during which the pen is in range of the screen and is moving at a speed above a predetermined threshold. The stitching server then looks for two consecutive envelopes from a pair of devices that match a specific pattern:

- The first envelope must end near the first screen's border and last longer than dT_{min1} (250 milliseconds).
- The second envelope must start near the second screen's border, and last longer than dT_{min2} (=100ms).
- The second envelope must start after the first envelope, but it may occur no longer than dT_{max} (=3.0s) after the first envelope. This time interval is sufficiently long to support stitching between tablets that are within arm's reach (a maximum of about 75 cm).

We found these criteria suffice to recognize intentionally executed stitching gestures, but just as importantly, false positives were not a problem: incidental pen motions from two users concurrently using pens on their mobile devices rarely satisfy these criteria. The main limitation of our stitching recognition algorithm is that it is difficult to perform a stitch that starts close to an edge of the screen; the system does not receive sufficient samples of the

changing pen location, before it leaves proximity, to be certain of whether the user is lifting the pen from the screen, or whether the user is sliding the pen towards another device to perform a stitching gesture. Also, our current implementation limits stitches to pairs of devices.

User Feedback for Stitching-

A stitch is recognized as soon as the first 100 milliseconds of the second envelope have been observed by the stitching server; it does not wait for the user to finish the motion. Performing this eager recognition allows us to provide users with feedback of the stitching gesture as soon as possible after the user has entered the second screen.

Feedback for a successful stitch consists of a short chirp sound as soon as eager recognition takes place. If the stitching gesture includes any operands, then the system shows a semi-transparent blue shadow on the screen in the shape of the selected photos (Fig. 2a). Upon completion of the stitching gesture, the system may also provide additional feedback. For example, for a copy or move operation, *StitchMaster* shows an animated semitransparent cone that appears to whisk files from one machine to the other (Fig. 1). This provides clear feedback of where the files came from, and where they were copied to (Fig. 2b).

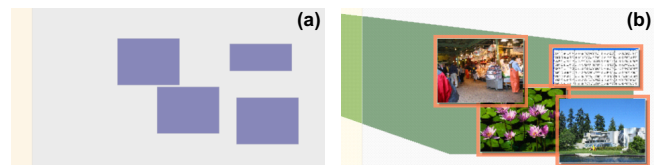


Fig. 2 Feedback for remote copy. (a) Shadows appear on the other device, then (b) when the user drops the photos, a cone connects them to their origin on the other device.

2. Specifying Connection Type: Multi-Device Commands

Multi-device commands supported by *StitchMaster* include copying or moving photographs, establishing a persistent shared work space, expanding an image across multiple displays, or entering a presentation mode known as the *gallery* (described below). *StitchMaster* presents these options in a pie menu. There are two basic design choices for where the command selection can occur:

- *Local menus*: Users choose the command (e.g. Copy) on their local screen, and then stitch to indicate the remote device that is involved.
- *Remote menus*: Users stitch to another device, and then a menu appears on the remote device providing options for how to combine the devices.

StitchMaster implements remote menus, which allows us to limit the visibility of multi-device operations to situations where they are known to be applicable; we did not want to complicate the single-device user experience with options for multi-device operations. Remote menus appear at the end of a stitching gesture when the user holds the pen still for 0.5 seconds. To provide feedback that a menu is a *remote* menu, *StitchMaster* shows a transparent blue cone that connects the remote menu back to the display where the stitching gesture originated (Fig. 3).

For some stitching gestures, StitchMaster assigns a default operation, eliminating the need to use the menus. For example, when stitching with a selected photo (that is, stitching using an operand as described in the next section), by default the selected photograph is *moved* to the other screen. We chose not to make *copy* the default since we found during pilot studies that users would repeatedly copy files back and forth while trying out stitching, quickly creating cluttered screens for themselves.

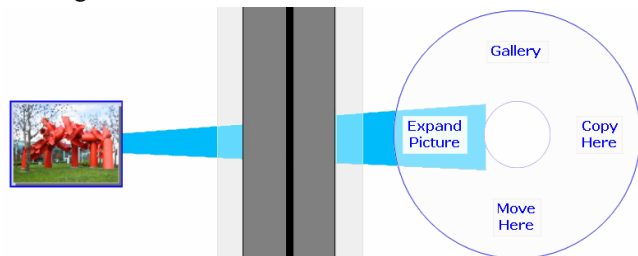


Fig. 3 A remote menu shows a link between screens.

Example Multi-Device Command: The Gallery

The Gallery (Fig. 4) allows one user to give a presentation of selected photos to another user. To start the Gallery, the presenter selects an image to start with, stitches to the other screen, and chooses *Gallery* from the remote menu. The other tablet then displays a full-screen view of the selected image, while the presenter's tablet displays thumbnails of all of his photos. The presenter can click on any thumbnail to change the image that is displayed on the other tablet.

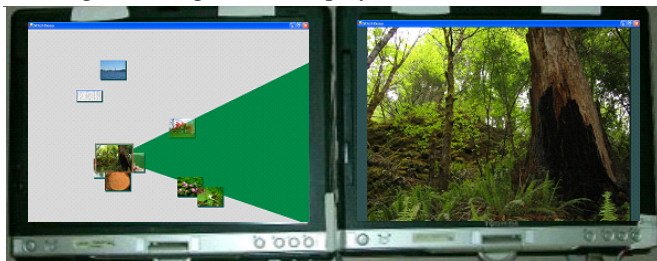


Fig. 4 Gallery: The right tablet displays a full-screen view of an image that the presenter selects on the left tablet.

The Gallery changes the roles of the devices. Instead of two identical devices, we now have one tablet for interaction, while the other primarily serves as a display. If users separate the devices, but keep the Gallery running, the presenter's tablet becomes a private view, while the other tablet represents a public view of selected information.

3. Specifying What to Share: Stitching with Operands

StitchMaster supports tapping on a single photo to select it, or drawing a lasso to select multiple photos. StitchMaster outlines the selected photos in orange and scales them to be slightly larger than the others (Fig. 5). Users can select a photo and then perform a stitching gesture to another device all in one gestural phrase [3][5][10]. The user makes a selection, and then lifts the pen slightly so that the pen is no longer in contact with the screen, but is still within tracking range of the Tablet PC screen. The user then stitches to the other display, and the selection is treated as the operand of the stitching gesture.

Phrasing works well, but we observed that users sometimes become focused on the selection step, and momentarily forget about stitching. Therefore, we do not *require* that stitching follow selection in a single uninterrupted gestural phrase. A stitching gesture that starts over a selection also includes that object as an operand, but after 3 seconds, the selection *cools* and will no longer be treated as the operand for a stitching gesture. The highlights around selected photos turn blue once the selection has cooled. This approach prevents old, stale selections from mistakenly being interpreted as the operand to a stitching gesture.

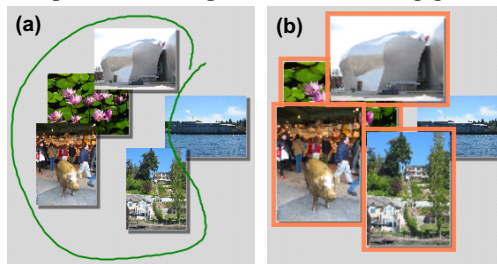


Fig. 5 (a) Multiple selection using the lasso gesture. (b) Selected photos scale up and highlight in orange.

4. Sharing Physical Space

With varying social and cultural conventions, individual preferences, and changing needs depending on the task, users need flexible ways to share physical space when combining devices. Hall distinguishes two distances within arm's reach, *intimate* and *personal*, with *social* and *public* distances beyond that [11]. StitchMaster includes support for intimate, personal, and social distances.

Intimate spaces support tight collaboration between friends or colleagues who may need to work together on a large document. For example, StitchMaster supports placing two tablets together and then expanding an image to fill both screens. The displays act as tiles of the same virtual space. This style is also well suited for a single user wishing to extend his workspace with additional devices.

Personal spaces. Users can stitch together tablets that are separated by up to about 75 cm. This allows a space created by stitching to act as a whole, yet each user maintains his or her own personal space. For example, StitchMaster allows users to create a *persistent shared workspace* by making a "simple stitch" from one screen to another without any operands. A vignette that appears to surround the two screens turns red to give users ongoing feedback that the two machines are connected. Either user has veto power over the connection and can close the workspace by choosing *Disconnect* from a menu.

Social spaces. Once users join a shared workspace, they can further separate their devices, yet still work together. For example, a user can employ the *transporter* to give photos to the other user, even if that user is no longer within arm's reach. The user drags a photo to the edge of the screen, and dwells with the pen. After a brief pause, during which we display an animation of a collapsing blue square, the photo is transported to the other device. This

pause is necessary to separate transporting a photo from normal drag operations; the collapsing blue square gives the user feedback that the picture is about to be transported.

Orientation of spaces. StitchMaster only supports stitching between tablets that are at the same physical orientation, so users must sit shoulder-to-shoulder. However, research suggests that communication patterns change when persons sit face-to-face, shoulder-to-shoulder, or at 90 degree angles to one another [11][20]. We expect it is technically feasible and may be valuable to extend stitching to support pen gestures that span tablets in any of these orientations.

5. Calculating the Spatial Relationship between Devices

To infer the spatial relationship between devices, stitching fits a line equation to the coordinates traversed by the pen on each screen. Of course, users do not move the pen in perfectly straight lines, but users do tend to move in arcs that can be locally approximated by a straight line.

When the stitching server detects a stitch from Device1 to Device2, it records a small window of samples as the pen leaves one screen and enters another, yielding p_0 (the exit point of the first pen trace), p_{10} (the entry point for the second pen trace), p_{11} (the point at which the stitch was recognized), and α_0 (the angle of motion at p_0); see Fig. 6.

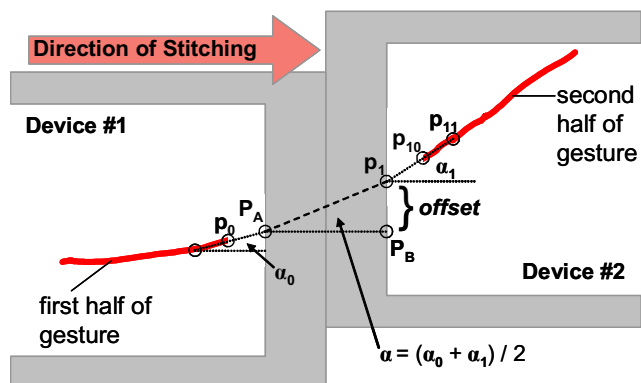


Fig. 6 Fitting a line to the user's pen gesture.

Due to the sampling rate of the pen, the first and last pen locations reported by the tablet may fall up to 3-4 cm from the edge of the screen. We found that calculating the width of the screen bezel or the extent of any empty space between the devices by using the time interval between the last and first observed samples may lead to inaccurate distance estimates. For this reason, we initialize the device's bezel thickness as a fixed constant, and then ignore any empty space that may be present between the devices.

We estimate the intersection of the stitching gesture with the edge of each screen, yielding the points PA and p_1 . PA is the intersection of the screen edge of Device1 with the line that passes through p_0 at an angle α_0 ; p_1 is the intersection of the second screen edge with line that passes through p_{10} and p_{11} at angle α_1 . If the line between PA and p_1 has angle α , the *offset* between the two screens is then $\tan(\alpha)$ times the bezel width. We estimate α as the average of α_0 and α_1 , which seems to work well, even if

the user follows an arcing path while stitching. We then calculate PB as the displacement of p_1 along the edge of Device2's screen by *offset* pixels.

Using this approach, our system can transform points from one device's coordinate system to the other, thus allowing the presentation of graphics that appear to span the devices.

6. Coexistence of Stitching with Traditional Interactions

Stitching must allow users to establish connections between devices without interfering with existing uses for the pen. Input states supported by pens [4] include *tracking* (moving the pen in close proximity to the screen, causing the tracking symbol to move), *dragging* (moving the pen in contact with the screen, causing an action such as dragging an object or leaving an ink trail), and *out-of-range* (the pen is not in the physical tracking range of the screen).

Stitching can be implemented using the dragging state, or using the tracking state. StitchMaster implements options to use either style of stitching, or both can be supported simultaneously (this is the default).

Stitching in the Dragging State

Since traditional GUI interactions occur in the dragging state, performing stitching by dragging could conflict with them. For example, when stitching via dragging, the first device cannot be sure whether to interpret a pen stroke as a drag until the second device recognizes the completion of the stitching gesture. To circumvent this problem and allow stitching via dragging to coexist with other dragging operations, we use *speculative execution* [2]: StitchMaster initially assumes all pen strokes are intended as drags. If the stitching server then reports a stitch, StitchMaster undoes the drag and instead treats the gesture as part of a stitch.

During preliminary user testing, we found that users can easily make a stroke while keeping the pen in contact with the screen, but when stitching to another display, the screen bezel gets in the way. This makes it hard for users to make a quick, fluid pen motion across the bezel while bearing down on the pen. Instead, users must drag the pen to the edge of the first screen, lift the pen to jump the screen bezel, and then complete the stitching gesture by pushing the pen back into contact with the second device's screen.

Stitching in the Tracking State

Stitching from the pen's Tracking state represents a more advanced skill than dragging, as it requires moving the pen while keeping the tip within 2 cm of the screen surface to prevent it from entering the *out-of-range* state. However, stitching by moving the pen just above the surface of the screen (with the base of the hand resting on the screen) allows the user to make a continuous, quick, and fluid movement that is not interrupted by the physical "speed bump" of the screen bezel. Another advantage of stitching in the tracking state is that it avoids the need for a speculative execution scheme: stitching gestures occupy a separate layer that rests on top of GUI interactions.

The main drawback of implementing stitching in the tracking state is that currently available personal digital assistants (PDA's) do not support tracking, so future extensions of stitching to PDA's would have to use the dragging state. Another potential problem is that users may use the pen to gesture while talking about the contents of the screen with a colleague, potentially resulting in a false-positive recognition of a stitching gesture. We designed our stitching recognition with this issue in mind, so false positives are rare, but no recognition scheme is foolproof.

USABILITY TESTING

We conducted a usability study of StitchMaster to identify usability issues and user concerns with stitching. Our primary goal was to verify if users could effectively use stitching gestures to perform multi-device operations.

Participants: 13 participants were recruited from the general public through Microsoft's usability pool. As the study required pairs of participants, the experimenter assumed the role of the "collaborating" participant for the 13th participant. None of the paired participants knew each other prior to the study. Collaborating pairs were of the following genders: 1 pair female-female, 3 pairs male-female, and 2 pairs male-male. Four of the participants had previously used a Tablet PC; an additional 6 participants had previously used some other type of pen-based device.

Materials: We ran the study on Toshiba Portege 3500 Tablet PC's with built-in 802.11 wireless networking. These devices measure 28.5 x 23cm with 25 x 18.7cm screens. Users employed the tablets in the slate (flat) mode.

Procedure: The participants sat shoulder-to-shoulder on the same side of a table; the experimenter sat at the opposite side of the table. Each participant was provided with a Tablet PC running StitchMaster. In a 2-3 minute practice session, the participants learned basic pen operations such as selecting images and dragging images on the screen. The experimenter then explained how to use features of StitchMaster, and allowed participants to try them out one-by-one, but did not show users what to do. Each session lasted approximately one hour.

Results

The experimenter first asked participants to "connect the devices by making a pen stroke across the devices" but did not show participants how to do this. With this instruction, all 13 participants, on their first or second try, created the *persistent shared workspace* by stitching with no operands.

All but two users made their first attempt at stitching by moving the pen in contact with the screen. Participants expressed no clear preference for performing stitching in the tracking state versus the dragging state; both seemed to work well. All participants at some point during the study used stitching in both manners, and users often would mix styles within the same gesture: for example, a user would perform the first half of a stitch by dragging, but then jump the bezel and complete the stitch from the tracking state.

The experimenter next explained how to move files between devices by stitching with operands. Originally, StitchMaster required the user to place the pen in contact with the other screen at the end of the stitching gesture to drop a photo. On our first day of testing, we found this was problematic for users who mixed the dragging and tracking styles of stitching. Users repeatedly moved the pen *away* from the screen when they wanted to drop a photo, rather than touching the screen.

We fixed this problem by having the software drop the photo if the user lifted the pen at the end of the stitch. When asked on the *first day* if "Sometimes I made mistakes when moving items" subjects tended to agree, with an average Likert scale response of 5.5 out of 7, but on the *second day* the average response improved to 4.0 (neither agree nor disagree); one user commented that "it was nice to drag items to the other screen without having to touch it."

While moving a photo to the other screen, participants sometimes would pause too long, which caused the remote menu to appear. Increasing the time-out for the remote menu reduced the number of subsequent cases of accidental activation, but did not completely eliminate this problem.

As we had anticipated, participants sometimes failed to stitch if they started too close to the edge of the screen. Participants wanted feedback of how far from the edge of the screen they had to start stitching in order to be successful. Adding 1-2cm margins would make this limitation visible, and prevent users from leaving photos at the extreme edges of the screen where this problem arises.

The only instances of false positive recognition of stitches that we observed occurred if users failed to successfully stitch, and then returned the pen to their screen to try again. Without realizing it, users often returned the pen to their original screen while remaining in the tracking state, and this was sometimes recognized as a stitch.

Overall, users were enthusiastic about the concept of stitching as embodied by StitchMaster. When asked if "I would use this software if it were available" the average response was 6.7 out of 7. However, one area of concern for many participants was security and privacy. For example, participants wanted to know if "Once connected, can a person take my other stuff?" or if there was a "lock-out for security and privacy." Currently, there is not.

The Proxemics of Co-located Collaboration

Our usability testing led us to two primary "lessons learned" in relation to proxemics:

Do Not Require Contact. We began testing sessions by instructing users to "put your tablets together." Although many users followed these directions, some users seemed hesitant to place their tablet in direct contact with that of the other user. In 3 of the 7 sessions, participants placed their tablets together, but asked "Do they have to be right next to one another?" When the experimenter replied that they did not, subjects moved them approximately 15 to 40

cm apart. Clearly, stitching must support gestures between tablets that are not immediately adjacent to one another. Fortunately, we had anticipated this in our design, so stitching worked well for these participants.

However, this does not mean that intimate spaces, with the devices close to or in contact with one another, are not useful. It depends on the users and what they are trying to accomplish. When asked at the end of the study if “Combing the screens of multiple Tablet PCs was a compelling capability,” the average response was 6.8 (agree) out of 7. Users commented that they liked “the ability to split the view, so there are no two faces trying to peek at only one screen” and that the “wide screen would be nice for collaboration, like for two people working on the same spreadsheet.” Thus, although participants worked with a stranger during the study, they seemed to envision other contexts where close, joint work would be valuable.

Establish and relax. Users want to *establish* a connection via stitching, but then *relax* the increasing social tension by quickly exiting the personal space of the other user. In our study, when one user reached over with the stylus, the other user would often lean back to make the intrusion into personal space less acute. Many subjects made short stitching gestures that only extended 3-5 cm onto the other user’s screen, and some users held the pen near the top, like a pointing stick, rather than holding it at its tip, like a writing instrument. Users may have adopted these behaviors in an effort to minimize intrusions into the other user’s personal space. Similarly, the transporter, which allows users to share files without repeatedly reaching into the other user’s personal space, was popular with test users.

Although participants successfully used remote menus to choose how to combine devices, this perspective does offer an argument against the use of remote menus, which require the user to perform command selections while reaching onto another user’s display. To avoid this, one approach we are experimenting with allows the user to stitch, return to his local screen, and then select the multi-device command to execute. Further usability testing will be required to see if users prefer this approach.

DISCUSSION

Security and Privacy

Security was one area of concern for some test users. Since only nearby persons can connect to a device, stitching does offer some inherent security measures. Social rules are at play, and because of the physical nature of the gesture, users who violate these rules by reaching onto a user’s screen without permission are likely to be noticed. Test users often verbalized their intent to stitch to another user’s screen; for example, one user commented “here’s a care package for you” before moving files to the other user’s screen. Nonetheless, users in an untrustworthy environment may wish to “lock out” stitching gestures, accept stitches only from devices which have previously been granted permission, or require a password.

Alternative Hardware for Stitching

One of the strengths of stitching is that it leverages widely available pen-operated mobile devices, but nonetheless future hardware enhancements may offer ways to improve our current implementation of stitching.

Unique ID. Stitching works well without a unique ID on the pen, but if pen ID’s become widely available, the ID could be used to boost the certainty that two separately observed pen traces represent a single pen gesture performed by one user. Whether or not a pen ID is available, recognizing the requirements for a versatile interaction paradigm for combining multiple mobile devices, and providing these via the aspects of stitching outlined in this paper, are the key contributions of our work.

Tracking beyond the screen boundary. We found that it is difficult for users to start a stitch from the extreme edges of the screen. If the tablet could continue to sense the pen location 1-2 cm beyond the edge of the screen, it might be possible to eliminate this problem.

Standardized Pens. The pen of one mobile device may not necessarily work on that of another manufacturer. If pens become standardized, they could work on any device. Alternatively, if all pen computers included touchscreens, users could use their fingers to make stitching gestures.

Multiple Pens. In our system, users cannot perform a stitching gesture to a tablet while the other user is already using a pen on that tablet, because current Tablet PC’s can sense only one pen at a time.

Multi-Device Stitching

We have recently extended our stitching system architecture to support formation of sets of up to 16 devices, but StitchMaster currently only supports formation of pairs of tablets. The stitching server adds a device to a connected set if the user stitches between a connected device and a new, disconnected device. We also plan to experiment with *long stitches* that traverse a series of devices, connecting them all in one continuous gesture.

Stitching with Other Types of Devices

PDA’s. Currently available PDA’s cannot sense the tracking state. We have not yet ported stitching to PDA’s, but since stitching can use the dragging state, we expect it is feasible to support stitching on PDA’s. Including PDA’s in our system may allow interesting new applications. For example, we have considered two designs that use PDA’s to alter the proxemics of stitching (Fig. 7). **Porches** use PDA’s as public areas for receiving visitors; each tablet has its own porch. To give a file to someone else, a user moves it onto the other user’s “porch” via stitching, or to offer a file for taking the user leaves it on his own porch. The other user can then take the file from a public porch into the more closely held main screen area. This reduces the need for each user to violate the personal space of the other user. **The candy dish** places a single PDA in the no-man’s-land between two other devices. Each user may then place files into the PDA via stitching, or take files that have been left

there by the other user. Again, the users would not have to reach into each other's personal space to share files.

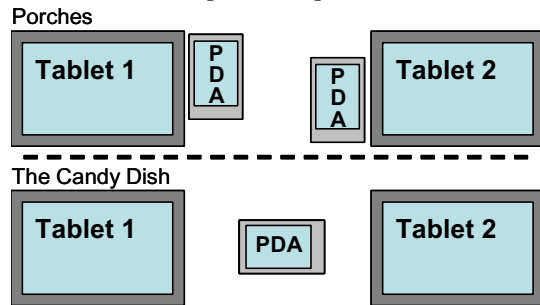


Fig. 7 Changing the proxemics of file sharing by using tablet computers and PDA's together.

Large Displays. It may be possible to support stitching from small devices onto a large-format pen-operated display. Because of the size disparity, the small device may occlude part of the large display, and stitching gestures may leave the edge of a small device but enter the large display almost anywhere. Since our current recognition algorithm looks for stitches that cross the edges of the screens, we would have to adapt our recognition policies. To avoid false-positives, it might become necessary to use a pen with a unique ID capability or to consider further features of the pen motion, including:

- The direction of travel or the curvature of the arc that the pen makes as it exits one screen and enters another.
- The velocity of the pen.
- The pen tilt angles (azimuth and elevation).

Alternatively, we could avoid recognition by requiring the user to explicitly signal stitching gestures. For example, the user could select a menu command such as *Stitch to Another Device...* before starting a stitch, or the user could hold down the barrel button on the pen while stitching.

CONCLUSION

We believe that the true untapped potential of the emerging wireless network lies in dynamic peer-to-peer coordination between proximal devices. Stitching provides an example of this perspective by offering users a versatile means to dynamically bind together pen-operated devices. We have provided some examples of multi-device commands, such as copying photos between devices, expanding an image across displays, creating a shared workspace, or using the gallery to project selected photos on another user's display. It is our hope that by identifying interaction requirements and usability issues for this new class of distributed pen interfaces, our work may spark further exploration of new applications, capabilities, and interaction techniques that foster communication, sharing, and collaboration between users of mobile devices, and empower users with new ways to combine the capabilities of multiple mobile devices.

REFERENCES

1. Bahl, P., Padmanabhan, V., *RADAR: An In-Building RF-Based User Location and Tracking System*, IEEE Computer and Communications Societies (INFOCOM 2000), 775-784.

2. Bartlett, J.F., *Rock 'n' Scroll Is Here to Stay*. IEEE Computer Graphics and Applications, 2000(May/June 2000): p. 40-45.
3. Buxton, W., *Chunking and Phrasing and the Design of Human-Computer Dialogues*, IFIP Information Processing '86: Amsterdam: North Holland Publishers, 475-480.
4. Buxton, W., *A three-state model of graphical input*, Proc. INTERACT'90, Amsterdam: Elsevier Science, 449-456.
5. Buxton, W., Fiume, E., Hill, R., Lee, A., Woo, C., *Continuous hand-gesture driven input*, Proceedings of Graphics Interface '83, 1983, 191-195.
6. Deasy, C.M., Lasswell, T.E., *Designing Places for People: A Handbook on Human Behavior for Architects, Designers, and Facility Managers*. 1985, New York: Watson-Guptill.
7. Efran, M., Cheyne, J., *Affective concomitants of the invasion of shared space: Behavioural, physiological, and verbal indicators*. J. Personality & Social Psych., 1974. **29**: 219-226.
8. Felipe, N., Sommer, R., *Invasions of personal space*. J. Personality and Social Psychology, 1966. **11**: p. 93-97.
9. Gorbet, M., Orth, M., Ishii, H., *Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography*, CHI'98, 49-56.
10. Guimbretiere, F., Winograd, T., *FlowMenu: Combining Command, Text, and Data Entry*, UIST 2000, 213-216.
11. Hall, E.T., *The Hidden Dimension*. 1966, New York: Doubleday.
12. Hinckley, K., *Distributed and Local Sensing Techniques for Face-to-Face Collaboration*, ICMI-PU'03, to appear.
13. Hinckley, K., *Synchronous Gestures for Multiple Users and Computers*, UIST'03, to appear.
14. Holmquist, L., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H., *Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts*, Ubicomp 2001: Springer-Verlag, 116-122.
15. Johanson, B., Hutchins, G., Winograd, T., Stone, M., *PointRight: experience with flexible input redirection in interactive workspaces*, UIST 2002, 227-234.
16. Kurtenbach, G., Buxton, W., *Issues in Combining Marking and Direct Manipulation Techniques*, UIST'91, 137-144.
17. Myers, B.A., *Using Hand-Held Devices and PCs Together*. Communications of the ACM, 2001. **44**(11): p. 34 - 41.
18. Rekimoto, J., *Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments*, UIST'97, 31-39.
19. Rekimoto, J., Saitoh, M., *Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments*, CHI'99.
20. Rodden, T., Rogers, Y., Halloran, J., Taylor, I., *Designing novel interactional workspaces to support face-to-face consultations*, CHI 2003.
21. Streitz, N.A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., P. Seitz, R., Steinmetz, *i-LAND: An interactive Landscape for Creativity and Innovation*, CHI'99, 120-127.
22. P. Tandler, T. Prante, C. Müller-Tomfelde, N. Streitz, R. Steinmetz, *Connectables: dynamic coupling of displays for the flexible creation of shared workspaces*, UIST'01, 11-20.
23. Weiser, M., *The Computer for the 21st Century*. Scientific American, 1991(September): p. 94-104.