Cooperative Stitching: Spontaneous Wireless Connections for Small Co-Located Groups (TECHNOTE)

Ken Hinckley
Microsoft Research
kenh@microsoft.com

Gonzalo Ramos
University of Toronto
bonzo@dgp.toronto.edu

ABSTRACT

Forming spontaneous connections between wirelessly networked devices is an important problem for ubiquitous computing. We explore *cooperative stitching*, which allows small collocated groups to easily create one-to-one or one-to-many connections between devices by performing simple dragging gestures to or from the top of the screen (to initiate or accept a connection, respectively). A usability study finds that users prefer cooperative stitching to a technique that requires physically reaching to the other devices, or to selecting names of devices from long lists.

Author Keywords

Co-located collaboration, mobility, wireless nets, pen input

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: Input.

INTRODUCTION

The proliferation of wireless mobile devices has led to increasing need for effective methods to form spontaneous connections to nearby devices without *a priori* knowledge of network names or addresses. For example, friends may wish to share music files and photos between their mobile phones, or office workers using tablet computers at a meeting may wish to exchange design sketches or distribute a document to the other participants. The latter meeting scenario requires spontaneous one-to-one connections as well as one-to-many connections.

The Infrared Data Association (IrDA) standard or Radio Frequency Identification (RFID) tags can enable a pair of mobile devices to discover one another when a user aligns them properly and places them in very close proximity [10,12]. However, unlike the simple act of handing a physical document to a colleague, properly aligning the devices remains awkward and disrupts the flow of a conversation.

Recent research explores connection of devices via physical gestures, such as bumping two devices together [3]. Pick-and-Drop [8] enables a user to pick a file from one screen using a special pen with an embedded unique identifier, and then drop the file onto another screen. Stitching [4] allows a user to link two pen-operated devices by making a straight-line pen gesture across the screens of the devices. Although these approaches seem to offer promising solutions for pairs of users, they do not support one-to-many connections, and do not support connections between proximal devices that may lie beyond arm's length. Furthermore, these techniques require users to touch their devices together or reach into the personal space of the other user, which users may find awkward in contexts where maintaining social distance is desired [2,4,11]. Location sensing [1] or discovery of proximal devices via wireless signal strengths [6] can bypass some of these issues, but are only partial solutions because they just enumerate candidate devices. To connect to a specific device, users still have to search through potentially long lists of network names for all the nearby devices.

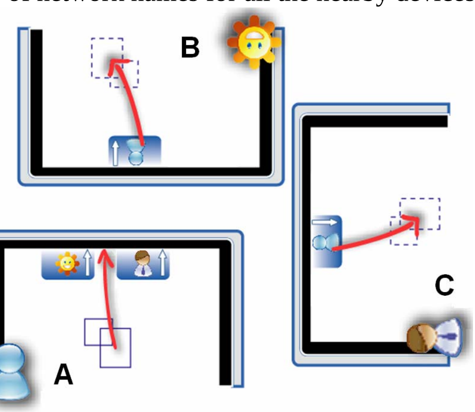

Fig. 1. Three users' devices arranged around a table. One user (A) initiates a transfer by dragging files to screen edge. Other co-located users (B, C) can receive them by pulling down from the top of their screens. As users B and C start dragging, intermediate feedback identifies the sender and previews the files being offered. The sender (A) then sees feedback showing who has accepted the proposed connection.

Cooperative stitching is a new technique that builds on "traditional" stitching [4]. Traditional stitching requires a single user to draw a pen stroke across two adjacent devices, so both devices must be within reach. By contrast, cooperative stitching can support connections between *two or more* devices that may lie *beyond arm's length*. Cooperative stitching splits a stitching gesture into two separate halves, each of which is performed by a separate, cooperating user (Fig. 1). One user initiates the stitch by drawing a stroke to the top of his device's screen, and a second user can complete the stitch by pulling the pen down from the top of the other device. This connects the devices.

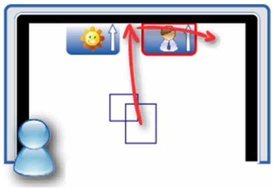

Fig. 2. Either user (here, the sender) can rescind a connection by simply crossing out the undesired user's icon. The actual file is never transferred between such users.

When one user initiates a stitch, multiple users can complete the gesture to form a one-to-many connection. Both the sender and receivers see graphical feedback showing who the participants in the connection are as well as a preview of any offered files. Either the sender or receiver may rescind the connection if desired (Fig. 2). Usability testing with groups of 4 users suggests that users can easily create shared connections, and strongly prefer cooperative stitching to either traditional stitching or choosing names of devices from long lists. However, connecting via a short list with only the names of 3 other users worked well.

RELATED WORK

Many techniques for spontaneous connection require the user to physically touch the devices [4,8,9], place the devices in close proximity [10,12], or both [3,5]. SyncTap [9] requires *one user* to simultaneously press a connection button on both devices, and does not consider connections formed by multiple users on multiple devices.

The mediaBlocks technique [13] uses wooden blocks with an embedded RFID tag as physical containers for data, but the blocks must be physically transported between devices. This precludes convenient establishment of one-to-many connections as it is impossible to pass the same physical block to more than one user at a time.

Pick-and-Drop [8], like mediaBlocks, depends on a unique identifier, which precludes connecting to more than one device at a time. Both Pick-and-Drop [8] and traditional stitching [4] require reaching onto the screen of another user, which forces a user to invade the personal space of the other person. This may produce feelings of social discomfort or anxiety in non-contact cultures [2,11].

Morris proposes multi-person *cooperative gestures* for tabletop interaction [7]. Cooperative stitching is an example of a cooperative gesture, but cooperative stitching is distributed across multiple devices as well as multiple users, and is designed specifically to address the spontaneous network connections problem.

COOPERATIVE STITCHING TECHNIQUES

Since cooperative stitching is designed for small collocated groups, users can employ social protocols such as body language, eye contact, or explicit verbal cues to coordinate their actions. This influences the *relative timing* the system can require for each user's contribution to a cooperative stitching gesture, as well as the degree of graphical *feedback* that may be necessary.

Relative Timing of Cooperative Stitching Gestures

Serial action: Traditional stitching recognizes a pattern where a pen stroke starts on one screen, leaves the screen, and then finishes on a second screen. Two cooperating users could emulate this pattern by having each user draw half of the gesture in a strictly serial order. Traditional stitching required the second half of the gesture to be draw within 1.5 seconds of the first half, but this short delay may not long enough for multiple cooperating users. However, using a longer time-out delays the completion of a connection, so choosing a good time-out value is difficult.

Synchronous action: We could instead follow the example of SyncTap and require simultaneous synchronous stitching gestures from each user. Unlike SyncTap, stitching gestures have a direction (the sender pushes up, and the receiver pulls down) so there is no ambiguity as to who is sending and who is receiving. However, it may be difficult or unnatural for multiple users to precisely coordinate their actions, particularly when the natural social grammar of gift-giving is for the giver to first offer the gift, and then for the taker to accept it.

Overlapped action: Cooperative stitching can allow for partially or fully overlapped action on the part of the cooperating users. This avoids the requirement for precise synchronization, and also can avoid the need for an arbitrary time-out by allowing the sending to start a stitching gesture and then "hold" open the offer to connect (by keeping the pen on the screen) for as long as is necessary or desired.

To explore these design options, we implemented two cooperative stitching techniques. The first, known as *Stitch+Lift* (Fig. 3 middle), is a serial action design. The

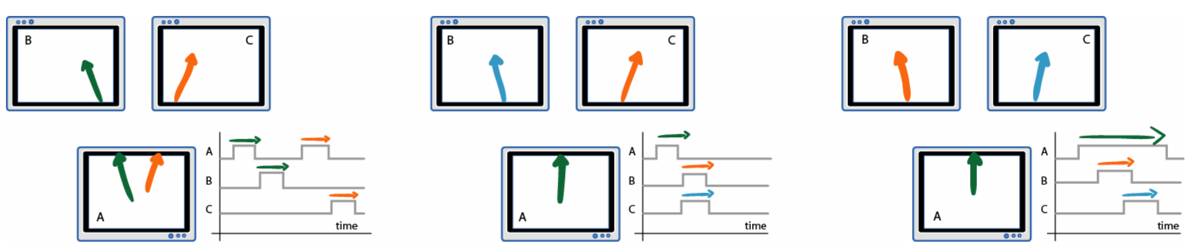

Fig. 3: ***Left: Traditional stitch***. To connect to two other devices, a sender (A) draws a stitch onto each device (B,C). ***Middle: Stitch+Lift.*** The sender initiates the connection, and each recipient must accept the offer within a short time window. ***Right: Stitch+Hold.*** The sender initiates the connection and holds the pen at the top of the screen until all desired recipients have accepted the connection. (In this figure, marks with the same color are done by one person, e.g. green-green. A multi-colored stitch is done by two people, e.g. green-orange.)

sender draws a straight line (stitching gesture) to the top of the screen and lifts the pen. Any other users then have 4 seconds to complete the stitch by drawing a straight line down from the tops of their screens.

The second, known as *Stitch+Hold* (Fig. 3 right), attempts to mimic the social protocol of handing an object to a person, and thus follows the overlapped action design approach. The sender draws a stitching gesture to the top of the screen and holds the pen at the edge of the screen to offer a connection. Any other users can complete the stitch as long as the sender continues to hold the pen down.

We decided that designs requiring synchronous action might be difficult for multiple users to coordinate, so we did not pursue them, but multi-user synchronous action may be an area worth exploring further in future work.

Graphical Feedback for Cooperative Stitching Gestures

For the graphical *feedback* that may be necessary to support effective cooperative stitching, there are several different types of feedback to consider, as well as different times at which it may be appropriate to offer that feedback.

First, what does the feedback show? Our early pilot testing made it clear that the system must consider the perspective of two different participants:

- The sender (the user that initiates the offer) should be aware of *who* is accepting his offer of a connection.
- Each recipient who accepts an offer should see *who* initiated the connection and also see *what* files or information is going to be received.

We provide awareness of *whom* by showing a tab containing an icon identifying *who* proposes the connection. The receiver sees what was received as they drag from the tab.

The second aspect of feedback we considered was: When does the feedback occur? Our pilot studies suggested that there are inherent tradeoffs in various strategies for when to present feedback, so we explored several approaches:

Early feedback: When the sender initiates a cooperative stitch, the system does not yet have any way to know which cooperating users will accept the stitch, but the system can broadcast the offer of the connection to other nearby devices. This appears as a tab that drops down from the top of other users' devices showing who is offering the connection. This reduces the need for users to rely on social protocols to form a connection, but the down side of this approach is that users may broadcast offers to unintended recipients and may also be distracted by undesired offers from nearby devices.

Intermediate feedback: Instead of offering feedback to users who may or may not participate in a cooperative stitch, the system can instead rely on users to initiate cooperative stitching through social protocols, and defer feedback until recipients start drawing a stitch. As soon as the system observes the first 100ms of the second half of a cooperative stitch, the system knows the user is accepting the connection, so only then does it reveal a drop-down tab showing who initiated the offer. As the accepting user continues to drag the pen, the system also shows a thumbnail preview (attached to the pen) of any incoming files At this point the accepting user can either complete the stitch, or decide to refuse it by dragging the files back to the top of the screen or crossing out the drop-down tab. Likewise, the sender sees a drop-down tab as soon as a recipient starts accepting a cooperative gesture, and can cross out any undesired recipients (Fig. 2).

Late feedback or no feedback: Finally, the system can wait until all users have finished their cooperative stitching gestures to reveal feedback of who the participants in a connection are, and what files have been transferred; or if social protocol is all that is necessary, the system could even forgo feedback altogether. However, we found during early pilot testing that users had a strong desire to see what they were accepting, and from whom, as quickly as possible. Even though social protocol seems sufficient for users to coordinate their actions, users still want immediate reassurance that they are connecting only with the intended parties.

Thus, we only implemented the *early* and *intermediate* graphical feedback strategies. We also experimented with audio feedback, but test users felt that audio cues would be inappropriate in the social context of a meeting.

Although the Stitch+Hold and Stitch+Lift techniques can both use either early or intermediate feedback, for our usability testing reported in the next section, we implemented the Stitch+Hold technique with early feedback and the Stitch+Lift technique with intermediate feedback.

Our system also supports traditional stitching ([4]). As show in Fig. 3, this offers an alternative way for users to connect to multiple devices, albeit in serial fashion rather than a true one-to-many connection formed in a single gesture. However, because traditional stitching and cooperative stitching can coexist, each technique can complement the other. In particular, the single pen stroke required for traditional stitching allows the relative orientations of each device to be determined, which is particularly useful for showing feedback or images that span two displays placed close together. Because cooperative stitching involves a single pen stroke on the sending device that may be accepted on multiple devices, the system has no idea where the screens are relative to one another. However, there is little need to show feedback that spans displays when those displays are widely separated. In our early pilot studies, for the type of group activities that cooperative stitching is designed to support, it was clear that users sitting around a table expect that "the group shared space is at the top" of the screen, i.e. towards the center of a table. This is why cooperative stitching uses the top edge of the display as the place to offer or accept connections between devices.

USABILITY TESTING

We conducted usability testing on the Stitch+Lift and Stitch+Hold cooperative stitching techniques. To serve as alternative designs for comparison, we also implemented traditional stitching, as well as connection by selecting the names of desired users from a list.

Fig. 4. Arrangement of devices for usability study.

Two groups of 4 users each were seated with Toshiba Portege 3500 Tablet PC's arranged around a 1.2 x 0.76 m rectangular table. The experimenter briefly demonstrated each technique, after which users performed a number of trials where they had to transfer an icon to one, two, or all three of the other users.

For the list conditions, we provided lists of either 3, 19, or 131 names. We assigned each participant a fictitious name which they kept throughout the study, and the lists used these fictitious names plus other distractor names, rather than cryptic computer names. The list with only 3 names takes the optimistic view that proximity sensing technology will one day be able to identify only and exactly the desired devices for users that are participating in a collocated group. The lists of 19 and 131 names represent scenarios where there are increasing numbers of nearby devices that might be candidates for connections. The 19-item list was small enough to view without scrolling.

Overall, users strongly preferred the cooperative stitching techniques to the traditional stitching technique in this task context. For the traditional stitching technique, the longest edge of the table was too far to reach the device at the opposite end, but we did observe that users would step closer or slide their device toward the other user to reduce the distance. User's preference between the Stitch+Lift and Stitch+Hold cooperative stitching techniques was evenly split. However, we observed that users would consistently gravitate to the Stitch+Hold manner of performing the stitching gesture once they had been exposed to it. Most users who preferred Stitch+Lift to Stitch+Hold found the early feedback offered by Stitch+Hold to be distracting, and senders also felt uneasy that this might advertise the availability of a connection to undesired participants. Thus, even though our pilot users had felt that early feedback might be helpful, our usability testing of the final implementation did not support this. Thus, we now believe Stitch+Hold with intermediate feedback offers the best design option for cooperative stitching.

The list with 3 names was the overall preferred technique for 5/7 particpants (one user's rank data was not collected). Thus, when a system can know exactly what persons are involved in a group, selecting participants from a list offers a very effective solution. However, only 2/7 particpants preferred the 19-item list to cooperative stitching, and most users found the very long 131 item list very cumbersome. Thus, lists are effective for a small number of candidate devices, but the solution scales very poorly as the number of proximal devices increases. Cooperative stitching thus offers a clear design advantage as the number of wireless mobile devices continues to proliferate.

CONCLUSION

Cooperative stitching offers advances over existing spontaneous connection techniques by allowing small groups of collocated users to easily form one-to-many connections to devices both within and beyond arm's length. Our preliminary usability testing shows that cooperative stitching supports such scenarios better than the "traditional" stitching technique of [4]. Cooperative stitching is also preferred by the majority of users for all but the most optimistic scenario of selecting names of devices from a list. The suggestion from pilot users that early feedback should show when other users were available to stitch was not well accepted in the final study. However, the intermediate feedback design approach proved sufficient to support small collocated groups. In future studies we would like to further explore the techniques, particularly to quantitatively analyze Stitch+Lift and Stitch+Hold style of gestures with intermediate feedback. We are also interested to see if cooperative gestures can be extended to other contexts, such as to support techniques similar to SyncTap [9], but for multiple users on multiple mobile phones, for example.

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. Hall, E. T., *The Hidden Dimension*. 1966, New York: Doubleday.
3. Hinckley, K. *Synchronous Gestures for Multiple Users and Computers*. UIST'03 149-158.
4. Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., Smith, M. *Stitching: Pen Gestures that Span Multiple Displays*. ACM Advanced Visual Interfaces (AVI 2004), 23-31.
5. 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, Springer-Verlag, 116-122.
6. Krumm, J., Hinckley, K. *The NearMe Wireless Proximity Server*. Ubicomp 2004.
7. Morris, M. R., Huang, A., Paepcke, A., Winograd, T. *Cooperative Gestures: Multi-User Gestural Interactions for Co-located Groupware*. CHI 2006, in press.
8. Rekimoto, J. *Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments*. UIST'97, 31-39.
9. Rekimoto, J., Ayatsuka, Y., Kohno, M. *SyncTap: An Interaction Technique for Mobile Networking*. Mobile HCI 2003, 104-115.
10. Rekimoto, J., Ayatsuka, Y., Kohno, M., Oba, H. *Proximal Interactions: A Direct Manipulation Technique for Wireless Networking*. INTERACT 2003.
11. Sommer, R., *Personal space*. 1969: Prentice-Hall.
12. Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N. A., Steinmetz, R. *Connectables: dynamic coupling of displays for the flexible creation of shared workspaces*. UIST 2001, 11-20.
13. Ullmer, B., Ishii, H., Glas, D. *mediaBlocks: Physical Containers, Transports, and Controls for Online Media*. SIGGRAPH'98, 379-86.