Synchronous Gestures for Co-located Collaboration on Multiple Mobile Devices

Yuguang Zeng¹, Jingyuan Zhang¹

¹Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487 USA

Abstract—Synchronous gestures have an intention to solve the sharing problems between mobile devices. It is always tedious for users to share their files with nearby users without wireless access points. We propose synchronous gestures to provide an interaction interface to connect users’ mobile devices. Those gestures include Pinch, which can be used to connect devices and share files between two devices, and DragAndPress, which enables one-to-many communication. We identify the general requirements of those gestures and implement a prototype system allowing users to share digital objects. Our approach is secure, has no bezel issues, and require no centralized servers or additional equipments.

Keywords: mobile devices; synchronous gestures; spontaneous device sharing; co-located collaboration

1. Introduction

Mobile devices have been becoming ubiquitous and it is not uncommon for people to gather together to exchange documents or photos through their mobile devices. Currently, even for nearby devices, users have to make efforts to share by email attachments, on-line storage services and other Web applications. With wireless technology, direct connection between surrounding devices is possible and more effective. Users still need to configure their devices to make them connected with each other. How to make mobile devices with limited I/O capabilities to achieve that in an efficient way? This is the spontaneous device sharing problems, defined as how purposeful connections can establish dynamically between two or more devices without knowing each other’s network address in advance. This is not only research issues for interaction design and system implementation, but an emerging problem desiring practical solution for daily life.

A new interaction technique known as synchronous gestures are proposed by researches to address spontaneous device sharing problem. With the technique, users make gestures to connect devices rather enter network address. Currently, proposed gestures include shaking two closed devices [1], bumping a pair of devices together [2], pressing a button on each device at the same time [3], and stitching a pen across surfaces of multiple devices [4]. The synchronous gestures are still under investigation.

We propose two synchronous gestures to let users communicate in a group with ease. Pinch gestures establish ad hoc connections between two nearby devices. When there is an item under the Pinch strokes, the item is shared on the other device involving the pinch gesture. DragAndPress enables a file to be shared among multiple devices. When one user drags a file out of screen side and other can press on their own screens if they want to receive the file. We build a prototype to show proof of concept and demonstrate the usability of our gestures.

2. Related Work

Previous researches have proposed systems for collaborating with mobile devices. However, some work requires special hardwares such as radio-frequency identification (RFID) tags [5], cameras [6], or sensors [7]. Some systems involve using nearby devices, and need manual configuration, including network configuration [8] and display configuration [9].

Synchronous gestures describe patterns of distributed user activities. Those activities have to happen at the same time or in a short consecutive time. There are literatures describing research on synchronous gestures. "Smart-Its Friends" [1] need users to hold together and shake a pair of accelerometer-augmented handled devices to connect them. The technique enables devices to receive movement data from other device and compared received data to its recent own movement data. If a similar pattern is recognized, a connection will be established between them. SyncTap [3] chooses synchronous buttons on devices at first. Two devices are connected when a user presses synchronous buttons down at the same time. The user needs to repeat synchronous operations if there is a collisions of overlapping actions at the same time. Bumping devices together can be used to create a shared display that spans two or more devices [2]. Signal patterns of accelerometers from two devices are compared to determine whether the bumping is intentional. Stitching [4] asks users to draw a continuous line across the screens of different devices. By analyzing the path between two devices, the system determines whether the path is a stitch. A server is needed to recognize stitch gestures and then informs devices network address if the server found a match. In addition, devices need to know the particular network address of the server.

There are some existing wireless standards related to the spontaneous device sharing problem. Bluetooth standardized as IEEE 802.15.1 [10] supports device discovery within radio range, but explicitly requires users to press button to add
nearby Bluetooth devices. Infrared Data Association (IrDA) [11] is used to transfer data between two devices and requires devices keep still during transmission. Wi-Fi Direct [12] enables devices to connect each other without an access point and to transfer data at a typical Wi-Fi speed as high as 250Mbps [13], which makes it possible for users to share large data such as videos, photos and documents.

3. Procedure for Pinch

The Fig. 1 shows the procedure of using pinch gesture to connect two devices. When two devices are laid together, pinch gesture can be performed on them. Each device captures the performed gesture and sends the gesture to nearby devices. After sending their own gestures, devices are waiting to receive gestures sent by nearby devices. On getting gestures from other devices, the gesture recognition algorithm described below is used to determine whether received gestures and its own gesture form a pinch gesture. If they do, a connection will be established between those devices.

4. The requirement of Pinch

Pinch is an explicit user command that enable multiple devices to be connected. Pinch is expected to provide a flexible and potentially extensible facility to support a number of different ways of combining devices, rather than supporting only a single operation or a very limited set of options as in previous systems like [1], [14]. Considering the expectation, Pinch address those design problem:

- **Connection**: How do devices connect to each other? A pinch gesture is performed on those devices which are to establish connection, and the system gives users visual feedback indicated a connection is established.

- **Coexistence**: How the system distinguish gesture for connection from gestures? A connection window shows up to perform Pinch gesture and disappears when a connection is established. Users invoke that window to appear if they need to connect other devices.

- **Proxemics**: How is physical space shared between users? It is a requirement of interaction techniques for impromptu association between devices to maintain social distance while users work closely. Pinch only requires users put their devices together to make connection. After establishing connection, devices can be moved to any place during users' collaboration activities.

5. The Mechanics of Pinch

With those design questions above, Pinch is developed as a new synchronous gestures. We now discuss general concept of Pinch on those design questions.

5.1 Establishment of a Connection

Before getting connected to other devices, devices needs to discover nearby devices, using services provided by underlying networks. We employ discovery mechanism from Wi-Fi Direct in the paper.

To establish a connection, users put their devices together and perform pinch gestures on those devices. The established connection is bidirectional [1], [3]. We focus on interaction techniques to form a purposeful connection between devices.

In our prototype system, each device automatically finds its nearby devices and sends them its gesture strokes. Comparing gesture strokes and time between its own and other devices, devices add as neighbors those having a match. Users do not need to enter any network address during the procedure.

5.2 Recognition of Pinch Gestures

Devices can recognize Pinch gesture by looking at the gesture strokes from nearby devices. Timeframe is defined as the time interval during which fingers start pressing on the screen, and end up leaving the screen as in Fig. 2. The participating device expects gesture strokes from other devices during timeframe.

The specific criteria are summarized in spatial and temporal aspects:

- Those gesture strokes must end near the adjacent screen edges and last longer than a timeout (150 milliseconds).

![Fig. 1: Procedure for Pinch](image)
The direction of those two strokes are opposite.
The slopes of those strokes are similar (must match within $\pm 5^\circ$).
Timeframes of gesture strokes are concurrent. After sending out their gesture strokes, devices must receive the gesture strokes from other devices within network delay (20 milliseconds).

These criteria are sufficient to recognize distributed pinch gestures done in purpose. False positive was not a problem because incidental finger motions from other devices rarely satisfy these criteria.

5.3 Coexistence of Traditional Interactions
Pinch aims to establish connection between devices. However, all the gestures near the edge make the system assume the users need to connect another device. We divide gestures into two categories. One is for connection and usually happens near the screen edge. No parameters is required for those gestures. The other is for operations for files and requires parameters to continue. The prototype system below will give better design for distinguishing those activities.

5.4 Pinch with Parameters
When an item is under a pinch gesture, the action of pinch gesture is re-defined as sharing the item to other device involving the gesture. That provides a one-to-one communication for users and that is useful in a group communication.

5.5 Sharing Physical Space
Users need flexible ways to share physical space in collaboration with others according to social conventions, task type, and individual preferences. Pinch gesture is designed for two kinds of distance: intimate and personal. Pinch gestures require devices put together for connection, which only happens in friends, colleagues working together. After connection, users can choose any space orientation that they feel like and protect their privacy information, and continue the communication or collaboration.

6. DragAndPress
DragAndPress is to provide one-to-many communication. The requirement of DragAndPress is to notify other group members when a user wants to share a file to others. That notification can be in either oral or message. When a file is dragged out of the screen on the user, only those devices pressed by users can receive the files.

7. Implementation
7.1 Development Environment
We choose Android platform and Android devices, Google Nexus 7 from Google. Android 4.0 (API level 14) or later devices support Wi-Fi peer-to-peer (P2P). Android Wi-Fi P2P complies with the Wi-Fi Alliance’s Wi-Fi Direct™ certification program. Those Android APIs enable developers to write applications that allow devices to discover and connect to other devices supporting Wi-Fi P2P, and then to communicate over a speedy connection.

7.2 Implementation Issues
7.2.1 One-to-one communication
Although Android Wi-Fi P2P framework provides APIs to discover and communicate with other devices, there is the following limitation on the framework. The communication mode is many-to-one when a group is formed. A device
creating the group acts as a group owner. Group members can only communicate with the group owner, and cannot communicate with other group members. Based on the Android Wi-Fi P2P, we implement a message exchange protocol, which enables devices to communicate with others directly. The following is the procedure:

(i) Acting as group owner, a device create a group, and invite other devices to join the group.
(ii) After nearby devices join the group, they will send their own IP address as message content to the group owner.
(iii) The group owner collects the incoming message and store them as a peer list. The list is wrapped in a message package and is sent to every device on the list.

7.2.2 Join pinch group

Our pinch gesture perform on two devices. However, it is not convenient and efficient to get every device connected!"n, if \(n\) is the number of device) . How spread quickly the transitivity character of pinch gestures? We employ the following method. When a pinch gesture is recognized, those devices add each other in their pinched list and send the list to group members. The friend of my friend is my friend. That greatly improves the efficiency\(n\), if \(n\) is the number of devices).

7.2.3 Gesture recognition

Recognition algorithm of gesture is based on the characters of the pinch gestures in Fig. 2. The algorithm is described as the following:

\[
\begin{align*}
\text{Input:} & \quad \text{Two gesture strokes, } g1, g2; \text{ Two timestamps, } t1, t2; \\
\text{Output:} & \quad \text{Decide whether two gesture strokes form a pinch gesture, } isPinch; \\
1: & \quad \text{// Compare difference of two timestamps } |t1 - t2| \text{ with time stamp difference threshold } T_{\text{threshold}}; \\
2: & \quad \text{if } |t1 - t2| > T_{\text{threshold}} \text{ then} \\
3: & \quad isPinch = false; \\
4: & \quad \text{return } isPinch \\
5: & \quad \text{end if} \\
6: & \\
7: & \quad \text{Extracting the set of sampling point arrays } sg1 \text{ and } sg2 \text{ from } g1 \text{ and } g2, \text{ respectively;} \\
8: & \quad \text{Calculate the cosine similarity, } similarity, \text{ of } sg1, sg2; \\
9: & \quad \text{if } similarity > Sim_{\text{threshold}} \text{ then} //Sim_{\text{threshold}} = 0.8, \text{ by default} \\
10: & \quad isPinch = false \\
11: & \quad \text{return } isPinch \\
12: & \quad \text{end if} \\
13: & \\
14: & \quad \text{Compare the direction of } g1 \text{ and } g2 \text{ on } x \text{ and } y \text{ axis;} \\
15: & \quad \text{if they are the same direction on } x \text{ and } y \text{ axis then} \\
16: & \quad isPinch = false \\
17: & \quad \text{return } isPinch \\
18: & \quad \text{end if} \\
19: & \\
20: & \quad \text{return } true
\end{align*}
\]

8. Discussion

8.1 Security and Privacy

Security was considered when mobile devices were used in collaboration. In order to connect nearby devices, pinch gestures have their own security measures. Devices are put together only under the permission from users. With permissions, one user can touch other’s devices. The physical nature of pinch gestures indicate the inherent social protocols, which invokes verbal communication on the beginning of collaboration.

8.2 Cooperative Pinch

While pinch gesture is performed by a single user, we discover pinch gesture can be performed in a cooperative way: users start draw strokes on their screens at the same time. The strokes need to meet the characters of pinch gestures. With cooperative pinch, we can develop a natural metaphor to support one-to-many connections. With such connection, a user sends digital objects such as documents to a group of users.

9. Conclusion and Future Works

Synchronous gestures are proposed to improve the collaboration in a dynamic peer-to-peer setting. Pinch provides a convenient interaction technique for users to bind mobile devices in ad hoc. DragAndPress provides one-to-many communication. To demonstrate concepts of our gestures, a prototype system is implemented. We also note the future exploration based on our gestures. We recognize the pinch
Fig. 3: Initial Window

Fig. 4: Sharing file
gesture by the behavior character, which was inaccurate on some time. Learning algorithm will be investigated to discover user patterns when performing pinch gestures. Exploring learning users’ patterns improve not only accuracy of gesture reorganization, but the security for connection. We hope more novel interaction techniques emerge to foster communication and collaboration and greatly improve user experience of multiple mobile devices.

References


