Combining Physical and Social Proximity for Device Pairing

Maria Husmann
Department of Computer Science
ETH Zurich
husmann@inf.ethz.ch

Sivaranjini Chithambaram
Department of Computer Science
ETH Zurich
chithams@student.ethz.ch

Moira C. Norrie
Department of Computer Science
ETH Zurich
norrie@inf.ethz.ch

Presented at the Cross-Surface '16 workshop, in conjunction with ACM CHI'16.
May 7, San Jose, USA.

Abstract
Cross-device interaction often starts with a pairing step where devices are associated. Pairing that is based purely on physical proximity risks accidental or even hostile pairing when users are physically but not socially close. In this paper, we explore how data on a user's social circle can inform the pairing process. We propose three pairing modes that balance user control with ease of use and trigger them based on social and physical proximity of the devices and users involved.

Author Keywords
cross-device; social proximity; physical proximity; device pairing

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction
As the number of devices in our lives has increased, researchers have been investigating how these devices can be used in combination [5, 10, 8]. Both single and multi-user scenarios have been explored [4]. To enable the usage of multiple devices in a coordinated manner, there needs to be an association step connecting the devices. A number of different approaches and interaction techniques have

been explored (see [3] for a summary). One option is to rely purely on proximity and allow interaction among all devices in a certain range, for example the same Wifi network or in range of a Bluetooth connection. However, these approaches come with an inherent security risk as proximity does not necessarily imply an interest in interacting, as illustrated by the case of a woman who received indecent images from a nearby passenger on a train [2]. The woman had enabled Airdrop to share a photo with someone else which allowed the perpetrator to also send pictures to her. There are security settings that allow the user to restrict access to people in the contact list, however, the user has to remember to switch to the right settings depending on who they want to share content with.

Another option is to require some manual coordination, for example a synchronous gesture with physical contact such as bumping two devices together [6], or continuous gestures that start on one device and continue on the next, for example stitching [7].

We propose another approach for pairing devices that adds *social proximity* as a dimension. Our main idea is to make device pairing very easy for devices with the same or socially close owners, and gradually add more steps as social distance increases.

Scenarios

Physical proximity does not necessarily imply social proximity, but rather is an orthogonal dimension. Figure 1 illustrates these two dimensions with sample scenarios.

1. In the top left corner, we have people who are physically but not socially close to each other, such as strangers on a train.
2. When we move to the top right, we decrease social

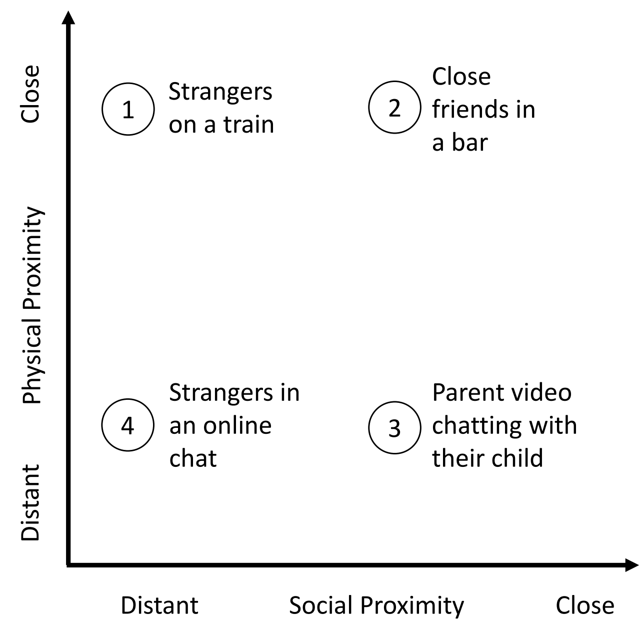

Figure 1: Scenarios of social and physical proximity.

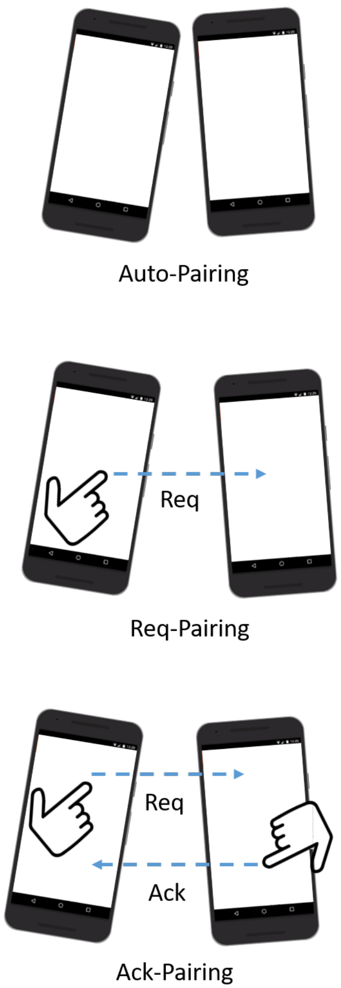

Figure 2: The three pairing modes with varying user control.

distance. An example here would be close friends spending time together in a bar.

3. In the bottom right, we have people who are socially close but not physically, such as a parent video chatting with their child.

4. Situated in the bottom left corner are people who are neither physically nor socially close, for example strangers chatting on the internet.

Our main interest is to provide a pairing mechanism for scenarios 2 and 3, while avoiding accidental pairing in scenario 1. Scenario 4 is not considered here since existing methods such as sharing a pairing code over an alternative channel such as a chat are sufficient.

Social Circles

Our social contacts can be grouped into circles [9], for example family, close friends, bike buddies or colleagues. Social networks such as Facebook or Google+ reflect this structure in their user interface and provide mechanisms such as friend lists or circles to organise one's contacts. Facebook even tries to automate this process to some extent with their Smart Lists[1] that group friends based on profile information, such as working for the same company. Furthermore, there are two standard lists, *close friends* and *acquaintances*. On the one hand, these lists can be used to tailor the information in a user's news feed to show more updates from close friends and fewer from more distant acquaintances. On the other hand, this information allows a user to share content with only a specific subset of contacts, for example all friends except acquaintances or only family.

[1] https://www.facebook.com/notes/facebook/improved-friend-lists/10150278932602131/

We propose that this information could be used to determine social distance, which, in turn, could help select an optimal pairing mode. The closer the owner of a given device is to us socially, the easier it should be to pair our devices. For more socially distant people, such as acquaintances, more control should be given to each of the users involved in the pairing.

Pairing Modes

We propose three different pairing modes (Fig. 2) for people who are connected via a social network.

- *Auto-Pairing* automatically pairs two devices and requires no user interaction.
- *Req-Pairing* requires a pairing request from the initiating device. No interaction is required from the receiving device.
- *Ack-Pairing* requires a pairing request from the initiating device and the receiver has to accept the request in order to establish the connection.

The modes illustrate the trade-off between ease of use and control. While the first mode (auto) requires no user interaction, it also offers no direct control over the pairing. In contrast, the last mode (ack) gives control to both involved users, however, it also requires both to explicitly interact with device. We propose that the appropriate mode should be chosen based on both the social and physical proximity of the involved users. Devices belonging to the same user are considered to have high social proximity and should be paired automatically if they are sufficiently close physically. As physical distance increases, Req-Pairing should be used. The assumption here is that a user would frequently pair devices that are physically close, for example in the

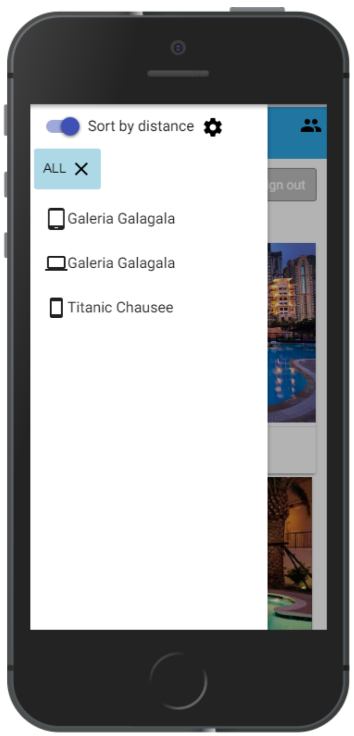

Figure 3: Contacts and their devices (represented by an icon) that are available for pairing, showing closest devices first.

same room, and this should be made as easy as possible. At the same time, this approach avoids that devices that are distant are paired accidentally, such as a user's phone in the office being paired with their computer at home. There could be occasions where this is desired, but we assume that these are rare, and would be supported with an explicit pairing request from the user. For devices that belong to socially close people, such as close friends or family, we propose Req-Pairing and, for socially more distant people, such as colleagues or acquaintances, Ack-Pairing could be used. Physical distance could be used to only display contacts within a specific distance or to sort contacts by distance, showing closer ones first.

For people who are physically close, but who do not (yet) have a social connection, we propose that existing proximity-based approaches such as scanning a QR code could be used for pairing.

Prototype Implementation

We have implemented the pairing modes outlined above in a prototype as an extension to our web-based cross-device framework XD-MVC[2] and used them in two sample applications. One challenge in the implementation was accessing a user's social circles and we have not yet found a satisfactory solution. While this information is available in Facebook and Google+, their APIs do not allow third-party applications to access it for privacy reasons. Google+ offers access to a user's circle via the Domains API[3], however, this API is only available to paid, corporate Google Apps accounts. Our workaround is based on Google's People API[4] and relationships are determined with the *relationship* attribute in a user's contacts. However, we assume that this attribute is rarely used and access to Facebook's friend lists would yield better results. Physical distance was determined based on the Geolocation API[5] that is provided by modern browsers.

Let us illustrate how the pairing modes can be used from a user perspective based on our sample application *Shared-Booking* which is a hotel booking application that distributes its user interface among paired devices (Fig. 4). As a first step, the application needs to be opened on all devices that are to be paired and the user needs to authenticate. The authentication can be stored on the device, so this only needs to be done the first time the application is used. If the same user is authenticated on multiple devices, these pair automatically and trigger redistribution of the user interface. In addition, there is a component in the UI that lists a user's contacts who also have the application open at the same time (Fig. 3). By default, the list is sorted to show physically close devices first. When the user selects a device, either *Req-* (for close friends and family) or *Ack-Pairing* is used. In the latter case, a popup is shown on the selected device asking its owner to confirm the pairing.

Since social relationships are not necessarily reciprocal [1], we choose the pairing mode based on the more distant relationship.

Discussion

Our prototype implementation has demonstrated that there are still a lot of open issues and questions. First, access to social circles is not easy to obtain with current API, despite the existence of the information in current social networks.

The fact that the user needs to open their application and

[2] https://github.com/mhusm/XD-MVC
[3] https://developers.google.com/+/domains/getting-started
[4] https://developers.google.com/people/

[5] https://developer.mozilla.org/en-US/docs/Web/API/Geolocation/Using_geolocation

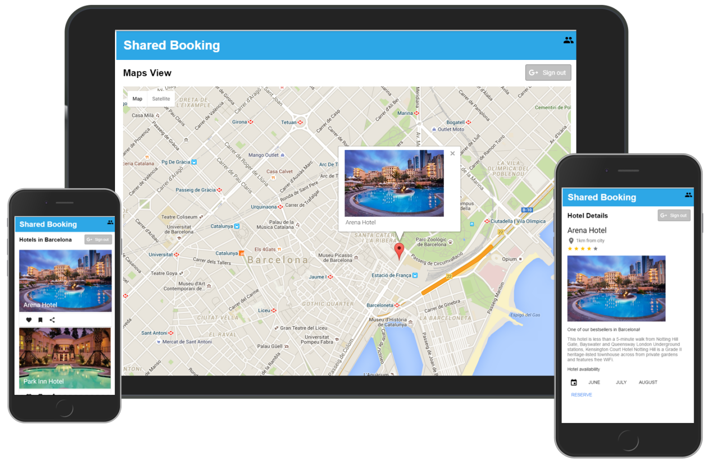

Figure 4: The SharedBooking sample application when used on three paired devices. The left device shows a list of hotels, the tablet in the middle a map marking the hotel chosen from the list, and the right phone shows details of the chosen hotel.

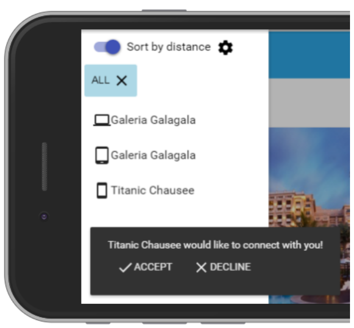

Figure 5: In *Ack-Pairing* mode a popup asks the user to confirm the pairing request.

be authenticated before the actual pairing implies that this approach is best suited for applications that are used frequently and accessible easily from a devic, for example installed on the home screen. Other approaches, such as pairing via QR codes or NFC allow the application URL to be transmitted along with a pairing token and may be better suited for less frequently used applications.

Our approach is based on the notion that people maintain lists of friends that reflect social distance. We have not confirmed to what extent this is actually the case. Alternatively, a social network provider could attempt to estimate the distance based on communication patterns. Furthermore, we have not yet validated our pairing modes with users. A user study could be conducted to investigate, if they meet actual user needs and to fine tune the social and physical boundaries that trigger each mode.

Finally, information on social circles could be used in cross-device applications for purposes other than pairing. Assume two users pair their mobile devices for photo sharing. After the pairing has been established, information on social proximity could be used to determine what pictures should be accessible on each device.

Conclusion

We have investigated how physical and social proximity can be combined to facilitate the pairing of devices and avoid accidental pairing that can occur when pairing is done solely based physical proximity. We have proposed three pairing modes and implemented them in a prototype. Our preliminary implementation showed that current social network APIs are lacking and that further research into using social network data in cross-device applications is needed.

Acknowledgements
This project was supported by grant No. 150189 of the Swiss National Science Foundation (SNF). Discussion during the first Cross-Surface workshop in 2015 stimulated this work.

REFERENCES

1. Abdullah Almaatouq, Laura Radaelli, Alex Pentland, and Erez Shmueli. 2016. Are You Your Friends' Friend? Poor Perception of Friendship Ties Limits the Ability to Promote Behavioral Change. *PloS one* 11, 3 (2016).
2. Sarah Bell. 2015. Police investigate 'first cyber-flashing' case. BBC News. (13 August 2015). Retrieved September 14, 2016 from `http://www.bbc.com/news/technology-33889225`.
3. Ming Ki Chong, Rene Mayrhofer, and Hans Gellersen. 2014. A Survey of User Interaction for Spontaneous Device Association. *ACM Comput. Surv.* 47, 1 (May 2014). DOI:`http://dx.doi.org/10.1145/2597768`
4. Linda Di Geronimo, Maria Husmann, and Moira C. Norrie. 2016. Surveying Personal Device Ecosystems with Cross-device Applications in Mind. In *Proceedings of the 5th ACM International Symposium on Pervasive Displays (PerDis '16)*. ACM, New York, NY, USA. DOI: `http://dx.doi.org/10.1145/2914920.2915028`
5. Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: Enabling and Understanding Cross-Device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2773–2782. `http://doi.acm.org/10.1145/2556288.2557170`
6. Ken Hinckley. 2003. Synchronous Gestures for Multiple Persons and Computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA. DOI:`http://dx.doi.org/10.1145/964696.964713`
7. Ken Hinckley, Gonzalo Ramos, Francois Guimbretiere, Patrick Baudisch, and Marc Smith. 2004. Stitching: Pen Gestures That Span Multiple Displays. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA. DOI: `http://dx.doi.org/10.1145/989863.989866`
8. Steven Houben, Paolo Tell, and Jakob E. Bardram. 2014. ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA. DOI: `http://dx.doi.org/10.1145/2669485.2669493`
9. Charles Kadushin. 1966. The Friends and Supporters of Psychotherapy: on Social Circles in Urban Life. *American Sociological Review* (1966), 786–802.
10. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 45–54. DOI: `http://dx.doi.org/10.1145/2669485.2669500`