IdLenses: Dynamic Personal Areas on Shared Surfaces

Dominik Schmidt, Ming Ki Chong, Hans Gellersen
Lancaster University, UK
{schmidtd, chong, hwg}@comp.lancs.ac.uk

ABSTRACT
IdLenses is a novel interaction concept to realize user-aware interfaces on shared surfaces. Users summon virtual lenses which allow for personalized input and output. The ability to create a lens instantaneously anywhere on the surface, and to move it around freely, enables users to fluidly control which part of their input is identifiable, and which shall remain anonymous. In this paper, we introduce the IdLenses concept and its interaction characteristics. Further, we discuss how it enables the personalization of input and output on shared surfaces.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. – Interaction styles

General terms: Design, Human Factors

Keywords: Interactive tabletops, surface computing, user identification, interaction techniques.

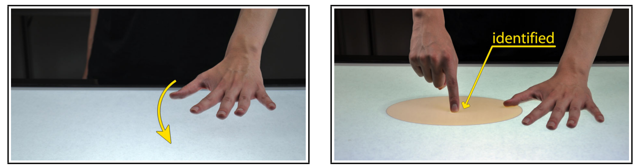
INTRODUCTION

Interactive surfaces are a compelling platform for collocated collaboration. When multiple users work together around a shared interface the question arises as to how personalized interactions can be realized. For example, users may wish to retrieve and manipulate personal data, access restricted functionalities, or customize the interface appearance.

Related work has outlined potential benefits arising from user-aware interfaces, that is interfaces employing the parameter of user identity. In particular, Ryall et al. [10] introduced the conceptual framework of identity-differentiating widgets. By knowing who interacts with a widget, its appearance or function can be dynamically customized to better support collaborative interaction. Further, user identity has been applied in surface computing applications to solve specific challenges, for example in access control scenarios [13] or techniques for document sharing [9].

Before personalizing any shared interface, users need to be identified. Previous work in this context has taken various approaches. DiamondTouch [4] instruments users and employs capacitive coupling through the body to distinguish the person touching. Other techniques take the users' locations around the tabletop [5] or orientation of fingers [3] to infer identities. Further, mobile devices can act as proxy for their users [15].

ITS'10, November 7-10, 2010, Saarbrücken, Germany.



(a) (b)

Figure 1: (a) IdLenses can be accessed anywhere on the surface, using a distinctive gesture. (b) All touches through the lens are identified. Lenses can be moved around freely.

Inspired by the fluidity of HandsDown [12], we present IdLenses as a novel interaction concept for dynamic interface personalization. On the lines of Toolglass and Magic Lenses [2], IdLenses exist on a transparent layer between applications and user input. In contrast to previous Magic Lens approaches [7, 14], each IdLens user has its own personal lens. In other words, the user's identity is always attached to a lens, hence enabling the dynamic customization of input and output. Figure 1 illustrates the user experience. IdLenses can be instantaneously summoned anywhere on the surface, by using a distinctive gesture. IdLenses constitute personal areas and can be freely moved around on the surface, and at the same time users are in full control of their identities attached with their inputs.

The contributions of this paper are as follows. We first introduce the IdLenses concept as a novel way of dynamic interface personalization for surface computing by describing its interaction characteristics. We then discuss what kind of personalizations are enabled by IdLenses (like, how user input and output can be customized).

INTERACTION CHARACTERISTICS

Since IdLenses are personalized, identity must be attached. As a result, user identification is mandatory. Conceptually, any user identification technique for interactive surfaces is eligible. Here, we have selected HandsDown [12] – a biometric identification approach based on hand contour analysis. Once registered, users can identify anywhere on the surface by simply putting down their registered hands. HandsDown requires systems that supports hand contour detection (like camera-based detection); nonetheless, no user instrumentation or further hardware is required. The technical evaluation in [12] showed that HandsDown identification of 17 registered users is substantially robust. Other than HandsDown, an alternative solution for user identification on an interactive surface is PhoneTouch [11]. PhoneTouch uses mobile phones as an enabling mechanism for attaching identities to IdLenses, but it requires external devices.

Figure 1 illustrates the basic interaction to activate an IdLens. John walks up to an interactive surface and places his preregistered hand on it (Figure 1(a)). He uses his non-dominant hand for opening the lens, as this allows the dominant hands to be available for finer dexterous interactions. The system captures the hand's contour and features to identify John. Once identified, the surface displays a personal virtual lens (Figure 1(b)). By John keeping his registered hand on the surface, the lens remains active and John can reposition his lens by sliding his non-dominant hand along the surface, potentially leveraging benefits of asymmetric bimanual input [6].

John can interact with his lens for functions that require his identity. All touches made through the lens are associated with his identity, while touches outside remain anonymous. Content displayed through the lens can be personalized. At the end of the session, John simply removes his registered hand from the surface, and the system closes or deactivates John's personal lens. In summary, John places his hand down to open his IdLens and the lens remains activated while his hand is on the surface; when he lifts his hand, the lens deactivates.

The concept we introduced so far focuses on the activation of a personal IdLens. After the lens is called, supplementary user interactions are needed for manipulation, like changing the lens' size. In the following, we identify and discuss these supplementary interactions.

Personalized Input and Output. Conceptually, an IdLens is a virtual layer overlaying above the application layer. In surface computing, this virtual layer can be used for both input and output of information flow between users and applications. By attaching user identity to the virtual layer, any information that flows through the layer can be personalized. In other words, any input made through the personal lens is attached with a user identity, while any input made outside the lens is anonymous. Similarly, display output can be personalized through the IdLens as well.

Initial Positioning. The position of the IdLens should avoid causing any obstructions. For instance, if a left hand is used to activate an IdLens, the lens should appear on the right side of the hand. If not, the user needs to cross arms to interact with the lens, hence an obstruction. An alternative solution is allowing the user to define the lens' position, for example by drawing its outline with the dominant hand. Of course, this flexibility comes at the cost of an extra user interaction each time an IdLens is called.

Display Persistency. After a user has activated a personal lens, there are two options to control the lens' display persistency. As described before, it can either be hidden or deactivated as soon as the user's hand is lifted, or remain activated even the user's hand is lifted. We selected the former approach, as it is appropriate for applications that are security sensitive. The approach prevents an IdLens to be left active while its owner is away, but it limits the use of the registered hand since it must linger on the surface to keep the lens active.

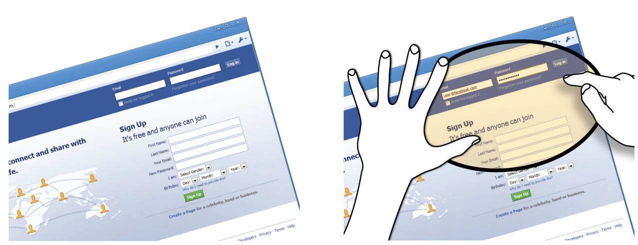
Asymmetric vs. Symmetric Bimanual Interactions. Users have both hands available if the application allows the users to lift their registered hands after activating an IdLens, and hence, symmetric bimanual interactions [1] are permitted. However, if one hand must remain on the surface (i.e. to keep the IdLens open), only asymmetrical bimanual interactions [6] are possible. In this case, the non-dominant hand coarsely positions the lens, setting the frame of reference for dexterous interactions with the dominant hand, much like positioning a sheet of paper to write on it.

Overlapping Personal Lenses. In a collaborative scenario where multiple IdLenses are opened, some of the lenses may overlap. The overlapping denotes a metaphor of users sharing a private group space. Hence, any action performed on the intersected area has all the IdLenses' identities attached. Nevertheless, this may cause conflicts; designers can avoid such situations by forbidding overlapping of IdLenses.

Reshape and Resizing. The appearance of IdLenses can be determined by the system or the users. Although our examples used an elliptical shape to resemble a real lens, the shape can vary according to the application needs. Alternatively, the lens can be drawn by users. Also, to resize an IdLens, we can adopt existing objects resizing gestures. For example, using a dedicated button or a slide-bar; dragging the frame of the lens towards or away from its center; or using multi-touch gestures like pinch-to-zoom.

APPLICATION SCENARIOS

After having introduced the basic interaction characteristics, we focus on new functionalities enabled by IdLenses; application scenarios are provided to illustrate the presented concepts. In particular, IdLenses add functionalities on three levels: The first two are related to input, the latter to output, as described in the following.



(a) (b)

Figure 2: Example of personalized behavior (with preview). (a) A protected web page requiring login. (b) Moving the lens over the login area automatically fills in credentials and enables login by simple touch.

Touch Identification

On the input side, an identity is attached to every contact made through a lens. In doing so, underlying applications and interface elements can attribute interactions to a user. This allows for several user-aware concepts.

- *Access Control:* Certain functionalities may only be available to a privileged subgroup of users; a button related to such a function can only be activated through the lens of an authorized user. For example in a crisis management scenario, any user could inspect the map while only authorized personnel would be allowed to issue commands to units in this interface.

- *Personalized Behavior:* Essentially the same function is performed but with a different behavior, depending on the user. The button "My Bookmarks"—appearing the same to all users—would retrieve a different list depending on who touches it. Browsing the web, a click on "Login" on a protected web page such as Facebook or GMail would log in the user if performed through a lens (Figure 2). Appropriate login credentials are automatically retrieved, hence rendering unnecessary the input of a password on a public surface. Furthermore, a start button could display the user's favorite applications, and even provide applications with the user's identity (e.g. enabling an E-mail client to start up instantly presenting the personal inbox).
- *Auditing and Logging:* As all input through a lens can be attributed to a user, such input can be recorded to track changes in a collaboratively edited document for example. In particular, it is clear who made an annotation. In this way, users can attach personal data to publicly available objects on a surface. IdLenses allow users to keep full control over which of their interactions can be identified and hence recorded.

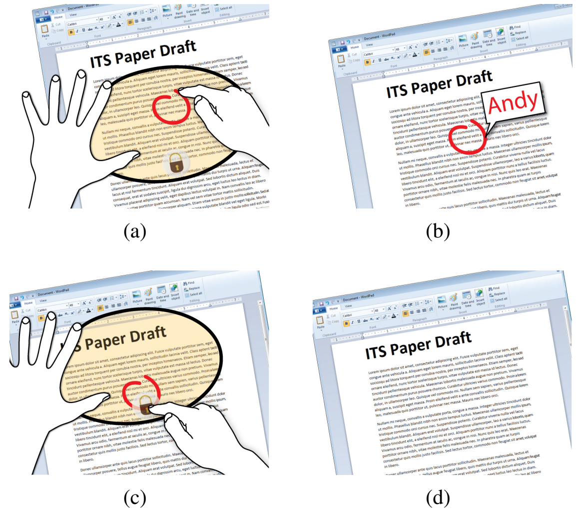


Figure 4: Personal annotations. (a) Adding an annotation to a document. (b) As entered through the IdLens, the annotation's author is known. (c) To make it private, the annotation is selected through a lock button. (d) Only the author can now see the annotation through his lens.

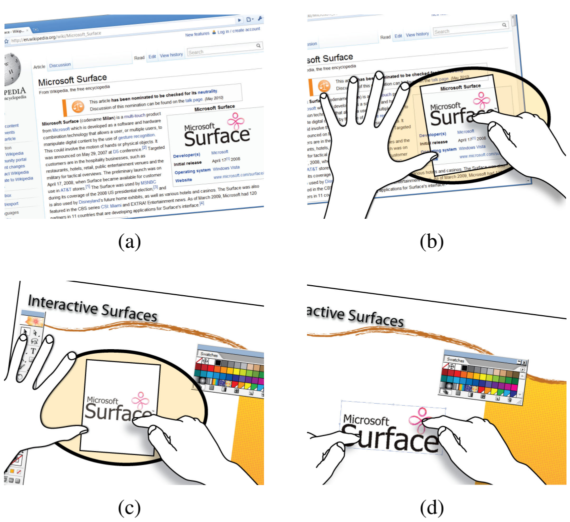


Figure 3: Individual clipboards as example of a personalized Toolglass. (a) The user browses a web page to find an image of interest. (b) Selecting the image through the lens puts it into the user's clipboard. (c) The lens is invoked over the target document where the image is to be pasted. (d) It is now part of the document and can be edited.

Personalized Toolglass

So far, touch input through a lens is simply given the additional parameter of user identity. Going further by applying the Toolglass [2] concept, we can add click-through elements (such as buttons) onto the lens, thus enabling extended functions (e.g. assigning a color to a shape). Now, users can directly activate a variety of functions, which otherwise might be difficult to access on a tabletop (e.g. menus might be out of reach). While the general concept of having a set of tools readily at hand anywhere on the surface is compelling as such, we focus on functions which are used in conjunction with user identity.

- *Custom Set of Functions:* Users can customize the set of functions to be shown in the lens. This way, multiple users can work in parallel with different sets of tools and easily access their preferred configuration.
- *User-dependent Functions:* Functions which inherently rely on user identity to be useful in a collaborative scenario are candidates for being added to an IdLens. For example, a clipboard function allows users to maintain a personal clipboard and hence work independently on a shared surface (Figure 3).
- *Alternative to Modal Input Sequences:* Selecting a tool or property to perform a related action (e.g. selecting a color to draw a line) is not straight forward to realize in shared applications [8] (e.g. how to represent on a shared palette that users selected different colors?) IdLenses provide a convenient alternative as these selections can be made through functions available on a lens.

Personalized Magic Lens

Besides input, the output can also be customized dependent on the user. Acting as a personalization filter, the appearance of elements underneath a lens is modified.

- *Personalization Preview:* As discussed before, the IdLenses concept allows the behavior of functions to be personalized. In this context, it makes sense to provide visual feedback to indicate that a personalized behavior is available. Continuing the example mentioned before, moving a lens over the login area of a web page fills out user name and password fields to indicate that personalized credentials are used when clicking on the login button (Figure 2).

- *Appearance:* Users may specify personal preferences, such as language, which are then applied when viewing the surface through a lens. For example, multiple users can explore a shared application in parallel in their respective language in a museum or exhibition setting. In another example users may adjust color schemes or contrast for easier readability.
- *Accessing Personal Data:* IdLenses further allow access of personal data attached to public objects. Moving the lens over a private or hidden piece of information makes it visible. For example, personal annotations are revealed when the lens is placed above it (Figure 4).

DISCUSSION

User identification is a prerequisite for IdLenses. Without user identity, all user inputs are regarded as anonymous by default; consequently, personalized interactions cannot be achieved. We chose HandsDown for identification, as it naturally allows the system to differentiate and identify users. Putting down a hand onto a surface constitutes a distinctive gesture to invoke an IdLens on the spot, while allowing the system to identify the user at the same time. Further, using the non-dominant hand for identification supports asymmetric bimanual interaction. The non-dominant hand can be made responsible for coarsely positioning the lens while the dominant hand performs fine-grained interactions. Other user identification techniques can be employed to enable the IdLenses concept. However, it might be necessary to introduce further steps.

As a defining characteristic, IdLenses puts users in full control of revealing their identity: It is the users who choose if their input remains anonymous (i.e. outside any lens) or can be attributed to their identity (i.e. input through a lens). Nevertheless, when an IdLens is active, it is up to the user to protect the lens. Any touch input made through a lens is attached with a user's identity. Thus, social protocols must be obeyed.

So far in this paper, we have examined the potentials and applications of IdLenses in the context of a single user per action. In reality, multiple users identities may be required to execute an action. The interaction of overlapping lenses introduces a new gesture for users to share identities on an interactive surface. In a collaborative scenario, multiple users can overlap their lenses to form an intersected area, which contains all the users' identities. Any action performed on this area automatically attaches the identities. For future work, we intent to investigate the potential of overlapping IdLenses for multiple users scenarios.

CONCLUSION

In this paper, we introduced IdLenses as a novel concept for spontaneous interface personalization in collaborative surface computing scenarios. The objective of IdLenses is to provide dynamic personal areas with the users' identities attached. Using the attached identity, an IdLens can personalize display output according to the owner's preference as well as customize functionalities for user inputs. Using hand contour analysis as enabling user identification mechanism, we outlined interaction characteristics to be considered when implementing IdLenses. Further, we discussed possible application areas on the basis of personalized input and output functionalities.

REFERENCES

1. R. Balakrishnan and K. Hinckley. Symmetric bimanual interaction. In *Proc. CHI*, 2000.
2. E. A. Bier, M. C Stone, M. C Pier, W. Buxton, and T. D. DeRose. Toolglass and Magic Lenses: the see-through interface. In *Proc. Computer Graphics and Interactive Techniques*, pages 73–80, 1993.
3. C. T. Dang, M Straub, and E. André. Hand distinction for multi-touch tabletop interaction. In *Proc. ITS*, pages 101–108, 2009.
4. P. H. Dietz and D. Leigh. DiamondTouch: A multi-user touch technology. In *Proc. UIST*, pages 219–226, 2001.
5. K. C. Dohse, T. Dohse, J. D. Still, and D. J. Parkhurst. Enhancing multi-user interaction with multi-touch tabletop displays using hand tracking. In *Proc. ACHI*, pages 297–302, 2008.
6. Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior*, 19:486–517, 1987.
7. S. Izadi, S. Hodges, S. Taylor, D. Rosenfeld, N. Villar, A. Butler, and J. Westhues. Going beyond the display: A surface technology with an electronically switchable diffuser. In *Proc. UIST*, pages 269–278, 2008.
8. B. A. Myers, H. Stiel, and R. Gargiulo. Collaboration using multiple PDAs connected to a PC. In *Proc. CSCW*, pages 285–294, 1998.
9. M. Ringel, K. Ryall, C. Shen, C. Forlines, and F. Vernier. Release, relocate, reorient, resize: Fluid techniques for document sharing on multi-user interactive tables. In *Proc. CHI EA*, pages 1441–1444, 2004.
10. K. Ryall, A. Esenther, K. Everitt, C. Forlines, M. R. Morris, C. Shen, S. Shipman, and F. Vernier. iDwidgets: parameterizing widgets by user identity. In *Proc. INTERACT*, pages 1124–1128, 2005.
11. D. Schmidt, F. Chehimi, E. Rukzio, and H Gellersen. PhoneTouch: A technique for direct phone interaction on surfaces. In *Proc. UIST*, 2010.
12. Dominik Schmidt, Ming Ki Chong, and Hans Gellersen. HandsDown: Hand-contour-based user identification for interactive surfaces. In *Proc. NordiCHI*, 2010.
13. J. Schöning, M. Rohs, and A. Krüger. Using mobile phones to spontaneously authenticate and interact with multi-touch surfaces. In *Workshop on designing multi-touch interaction techniques for coupled public and private displays*, pages 41–45, 2008.
14. M. Spindler, S. Stellmach, and R. Dachselt. PaperLens: Advanced magic lens interaction above the tabletop. In *Proc. ITS*, pages 77–84, 2009.
15. A. D. Wilson and R. Sarin. BlueTable: Connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proc. GI*, pages 119–125, 2007.