HandsDown: Hand-contour-based User Identification for Interactive Surfaces

Dominik Schmidt, Ming Ki Chong, Hans Gellersen
Computing Department, Lancaster University
Lancaster UK, LA1 4WA
{schmidtd, chong, hwg}@comp.lancs.ac.uk

ABSTRACT
HandsDown is a novel technique for user identification on interactive surfaces. It enables users to access personal data on a shared surface, to associate objects with their identity, and to fluidly customize appearance, content, or functionality of the user interface. To identify, users put down their hand flat on the surface. HandsDown is based on hand contour analysis; neither user instrumentation nor external devices are required for identification. Characteristic features of the hand are initially extracted from images captured by the surface's camera system and then classified using Support Vector Machines (SVM).

We present a proof-of-concept implementation and show results of our evaluation which indicates the technique's robustness for user identification within small groups. Additionally, we introduce a set of interaction techniques to illustrate how HandsDown can improve the user experience, and we discuss the design space of such interactions.

Author Keywords
Interactive tabletops, surface computing, multi-touch interaction, user identification, authentication

ACM Classification Keywords
H.5.2 Information interfaces and presentation (e.g., HCI): User Interfaces—*Input devices and strategies (e.g., mouse, touchscreen)*

INTRODUCTION
Interactive surfaces are a compelling platform for natural input and collocated collaboration. They have become a focus of research and commercial activity in recent years. A large body of work in this area is concerned with multi-user interactions [26]. For example, researchers have investigated the role of territoriality in tabletop workspaces [27] or multi-user coordination policies [15]. Although most interactive surface systems can track multiple points of contact, only very few attempt to distinguish between different users. Without user identification, all input is alike: It is impossible to tell apart and respond individually to interactions from different users working together.

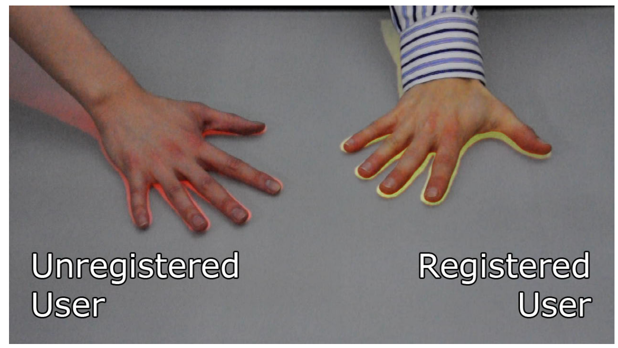


Figure 1. Users put down their hand flat on the surface to identify. In this example, visual feedback about the success is given by means of differently colored contours displayed right underneath the users' hands.

User identification enables a whole new set of interaction possibilities, including multiuser-aware interfaces [18] and access control [17, 25]. For example, users can instantaneously access personal data or customize interactions: Starting a browser application will bring up the user's customized start page, or touching the "My Documents" folder will show personal documents of the user who invoked the action.

In this paper, we introduce HandsDown, our approach to user identification for interactive surfaces (Figure 1) To identify, users put down their hand flat onto the surface, the fingers spread clearly apart (Figure 2(a)). For example, a personal picture collection can be retrieved and displayed in front of the user, once successfully identified. Leveraging hand contour information, the user interface element is automatically oriented towards the user (Figure 2(b)). HandsDown seamlessly extends conventional multi-touch on interactive surfaces: Users can manipulate elements using common multi-touch gestures, such as pinching to resize (Figure 2(c)). Hand gestures and finger input can be used simultaneously. While the left user in Figure 2(d) is browsing through photos using finger interactions, the right user puts down their hand for identification. Appropriate feedback is displayed if an unregistered user attempts to identify (Figure 2(e)).

NordiCHI 2010, October 16 - 20, 2010, Reykjavik, Iceland.

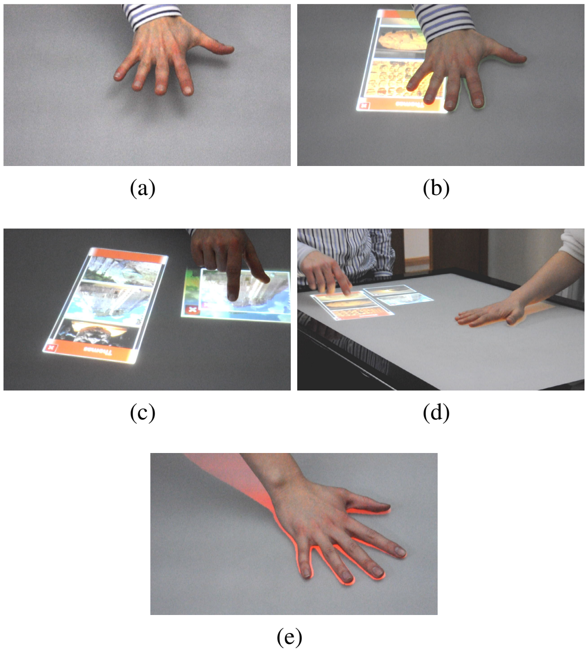


Figure 2. Here, HandsDown down is used to identify users and access personal picture collections. (a) A hand can be put down at an arbitrary location on the surface. (b) Once identified, a personal picture collection is displayed, automatically oriented towards the user. (c) Multi-touch manipulation is possible using finger interactions. (d) HandsDown and finger input can be used simultaneously. (e) Appropriate feedback is displayed if a user cannot be identified.

No user instrumentation or external devices are required. A hand can be put down at any location on the surface. Moreover, it can be arbitrarily oriented. This is especially important for setups where users are standing at different sides of a table. Multiple users can put down hands to identify simultaneously (as long as their hands do not overlap). In addition, HandsDown can be applied for biometric user authentication. From a user's perspective, the interaction is similar. However, before putting down a hand, users have to claim their identity. This can be achieved, for example, by selecting a name out of a list. Here, putting down the hand can be used as alternative to entering a password.

Before users can be identified, they need to register by providing sample hand contours and associate them to an identity stored on the system. This procedure is similar to enrolling for finger print authentication on an enabled laptop computer, for example. It is important to note that HandsDown is *not* a general approach to user identification in a sense where every touch on an interactive surface can be directly associated to a user. The technique rather integrates the concept of hand-contour based identification (previously employed in access control systems) with instantaneous and fluid interactions on interactive surfaces, enabling users to immediately identify on demand.

The remainder of this paper is structured as follows. First, we review related work. Then we describe our system design and report results from a formal evaluation of the identification performance for small user groups. Next, we explore the surrounding design space by illustrating a series of concrete interaction techniques which are enabled by HandsDown. Finally, we make a series of design recommendations for those looking to adopt HandsDown for user identification.

RELATED WORK

Before surface computing systems emerged on a larger scale, the application of user identity information to interfaces has been explored in the context of single- or shared-display groupware (SDG). Stewart et. al [28] associate users to cursors and input devices. They present a collaborative drawing application which only allows the user who selected a tool to change its properties. Similarly, the "multi-device multi-user multi-editor" [1] was designed for groups collaboratively editing text on a single screen. This system supports multiple pointing devices, each registered to a user. PDA are used to control cursor and keyboard on a shared display in the Pebbles project [16]. As every device runs its own client application, cursors can be associated with devices and users.

DiamondTouch [6] is a tabletop technology for front-projected systems which supports user identification for up to four users. It uses capacitive coupling through the user who has to be in constant contact with a receiver. Dohse et. al [7] take a first step towards identifying users in a vision-based tabletop system. By overlaying hands and fingers, touches can be grouped and assigned to users based on table sides. Without capturing hand information explicitly, Dang et al. [5] exploit the fingers' orientations to infer higher-level information about the users' hand positions. On a conceptual level, Ryall et. al [18] introduce a framework of identity-differentiating widgets which gives rise to a set of novel interaction techniques.

Identification systems based on hand geometry have been developed as early as in the 1970s [10]. Most of them are used for single user authentication, for example in access control systems, time and attendance monitoring, or point of sales applications [31]. Sanchez-Reillo et. al [19] propose the extraction of finger lengths and widths, among others, for user identification and evaluate four different pattern recognition techniques. However, their approach requires the hand to be aligned on a special platform to take top and side view pictures with a camera. While Boreki et. al's [3] approach does not impose restrictions on the hand alignment, a flatbed scanner is used for acquiring an image of the isolated hand. They present a curvature-based approach for feature extraction and use mean values of finger lengths and widths in conjunction with a distance-based classifier for system access authentication in a static context. Likewise, Yörük et. al [30] describe a method for hand-based person identification for unconstrained hand poses. In their experiment, they used a flatbed scanner to acquire images and showed a robust performance for groups of about 500 users.

Hand contour identification appears to be a promising candidate also for interactive surfaces because of its modest hardware (low resolution camera) and software (low computational cost algorithms) requirements. In addition, it does not require user instrumentation and can easily be integrated with existing systems. In contrast to the application of hand contours for identification or authentication so far, using this mechanism on interactive surfaces poses new challenges. First, the large, unconstrained surface requires methods which are robust to different parts of the hands and other objects placed on the surface. Second, multiple users must be able to use the system simultaneously. Third, the system must operate in real time to enable fluid interactions. Fourth, the identification procedure must be integrated with conventional surface interactions.

IMPLEMENTATION

In summary, HandsDown is based on characteristic hand contour features which are extracted from images captured by the system's camera. In line with prior work [3], we require a flat hand posture, the fingers kept clearly apart, to afford identification. Neither the hand's location on the surface nor its rotation is restricted; multiple users can be identified simultaneously. Out of the hands' silhouettes we extract characteristic features, such as finger lengths and widths, by applying computer vision methods. A supervised classification method determines the users' identities, or authenticates them by verifying a claimed identity.

System Design

HandsDown is designed for interactive surfaces which can detect arbitrarily shaped objects in addition to finger touches. A prominent example is Microsoft's Surface [14], a commercial interactive tabletop which uses the diffused illumination (DI) technique to detect not only finger touches but also visual markers or other objects. In a nutshell, DI is based on emitting infrared light from behind the surface. This light is reflected by objects coming close to the surface which are then detected by a camera.

For our experiments, we use a custom-built tabletop system [20]. Similar to Microsoft's Surface, we also employ rear-projection and a camera-based detection method. However, instead of DI, we use a closely related technique, namely diffused surface illumination (DSI). While in DI light is emitted from behind the surface, shining through it, DSI uses an array of LED around the edges of a special surface material which emits the inserted light uniformly across the surface. As a consequence, DSI allows for an easier and more compact system setup compared to DI. In addition, it makes touches appear with a higher contrast due to the effect of frustrated total internal reflection (FTIR) [24].

Our tabletop has a surface diagonal of 100cm with a display resolution of 1280×768px. Figure 3 shows a schematic drawing of the camera and processing setup. A Point Grey Dragonfly2 camera with a resolution of 1024×768px at 30Hz is used for simultaneous finger and hand tracking from below the surface. In addition to this basic setup, we equipped our system with an optional, overhead mounted camera of the same type. This allows us to also track hands *above* the surface, that is, hands not in touch with the surface but hovering over it. In doing so, all subsequent finger interactions originating from an already identified hand can be associated to a user. Because of the diffusing projection screen, this is not possible from below; hands at a height of only a couple of centimeters are already too blurred to be clearly visible. Both cameras are equipped with infrared bandpass filters to avoid interferences with the projected image.

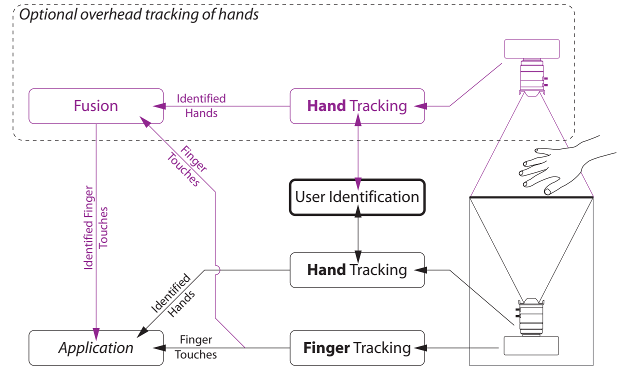


Figure 3. In our tabletop system, the camera below the surface tracks fingers and hands simultaneously. Optionally, an overhead mounted camera can be used to track hands also *above* the surface. This allows fingers to be associated with hands (and user identities) during conventional finger input.

Feature Extraction Steps

Hand Contour Extraction

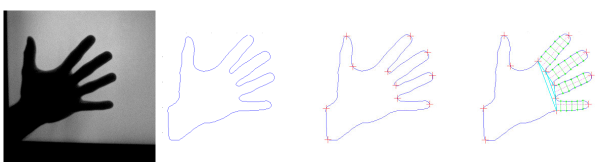
In our DSI setup, hands put down on the surface reflect the emitted infrared light and are therefore clearly visible by the integrated camera. The camera's infrared bandpass filter removes the projected image to prevent interferences with the detection. We apply two sorts of image filter chains to extract finger touches and hand contours in parallel out of the same source. The overhead camera is an alternative source for hand contour extraction. Here, hands appear as clear shadows in front of the illuminated surface (Figure 4(a)). Again, an infrared bandpass filter is used to remove visible light. In contrast to the integrated camera, the overhead camera bears the advantage of tracking hands also above the surface, that is in a hovering state. Independent of the camera used, we subtract the image background, that is the empty surface, and apply a binary thresholding filter. Contours are extracted using the chain code contour extraction method (Figure 4(b)).

Preprocessing

The following steps are only initiated once a user puts down their hand flat onto the surface. A flat hand can be detected by inspecting the contact area underneath it. When pressing down the hand, the lower part of the palm clearly shows up in a similar intensity as regular finger touches. In doing so, we can avoid unnecessary computations. More importantly though, by ensuring that a hand is completely put down onto the surface before identification, the distance between hand and camera is constant. Consistent measurements are required for biometric hand identification as the whole process is based on the hand's geometry.

To localize hand extremities (finger tips and valleys, that is the lowest point between two fingers) in a rotation and translation invariant way, we analyze the contour's curvature profile, as described by Boreki et al. [3]. The profile is extracted using the difference of slope technique. As points with high curvature correspond to changes in contour direction, we apply a threshold filter to select them. The intervals' respective center points are selected as hand extremity candidates (indicated by red crosses in Figure 4(c)). In contrast to previous hand contour systems for single user access control, we have to take into account that the whole surface is captured rather than an isolated area. Therefore, we do not only encounter multiple hands, but also have to deal with a variety of different shapes since parts of the arms might be visible, depending on how far users have to lean over the surface to reach a point (see Figure 5(b)).

Consequently, these non-hand parts have to be ignored. We remove them by searching for a pattern of alternations in contour direction which is characteristic to the five spread fingers. In the same way, unsuitable hand postures and objects other than hands which triggered the identification procedure can be excluded from further processing. The outer points of pinkie and index finger are reconstructed in a post-processing step, as they cannot be reliably detected due to their low curvature. They are placed at the same distance from the respective finger tip as the already identified valley points on the other side.



(a) (b) (c) (d)

Figure 4. Extraction steps: (a) raw camera image, (b) extracted contours, (c) high curvature points, (d) extracted hand features

Feature Extraction

The lines connecting finger tips and center points between two adjacent finger valleys are extracted as the fingers' main axis and divided into six equally sized partitions (Figure 4(d)). For each finger, we select the following features: length of main axis, widths at five equidistant points, and mean width. In addition, we include the palm width as well as three distances between different finger valley points. Note that the thumb is not included as its detection proved to be unreliable.

EVALUATION

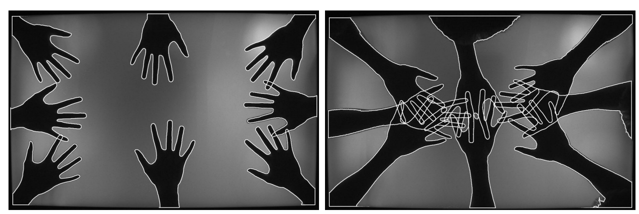
We use Receiver Operating Characteristics (ROC) curves [8] for performance evaluation as they provide a performance measure independent of class skew (i.e., unequal occurrence of individual classes) and classification threshold. They plot true positive (or genuine acceptance) and false positive (or false acceptance) rates as a function of the classifier's threshold. This threshold can be chosen depending on application requirements to achieve suitable trade-offs between security (low false positive rates) and recognition performance (high true positive rates).

AUC (i.e., the area under the ROC curve) reduces the ROC performance to a single scalar value for comparison while preserving its advantages. Extending ROC to our multi-class problem, we generate an ROC curve for each user, with the respective user as the positive class and all other registered users as the negative class. The AUC is calculated for each curve separately and then averaged.

Based on a pilot study [22] comparing the classification performance of Support Vector Machines (SVM) and Naive Bayes Classifier (NBC), we chose SVM as classifier in our system due to its consistently better performance. Additionally, we compared the effect of different numbers of training samples (5, 17, and 29), that is hand contour snapshots used for enrolling a new user. Based on the results, we chose to use 29 training samples per user in the following evaluation. As the employed camera is capable of capturing 30 images per second, enrolling a subject is quickly accomplished.

Data Collection

In total, we collected 544 hand images of 17 different subjects using the camera mounted over the surface. As we employed identical cameras and the processing steps remain the same, we expect the results to be comparable when using the integrated camera. The desired hand locations and orientations on the surface were briefly explained and demonstrated to the subjects beforehand; neither markers nor pegs were used. Thirty-two images were captured per subject as follows ($32 \times 17 = 544$): We asked them to position their right hand successively at eight different locations on the surface, close to its edge, with varying orientations; each position was recorded twice (Figure 5(a)). We repeated the same procedure with the hands positioned farther away from the edge, closer to the surface's center (Figure 5(b)).



(a) Close-to-edge condition

(b) Center condition (arms and sometimes the head is part of the extracted contour)

Figure 5. Location and orientation of captured hand images for evaluation (super-imposed camera shots; white contours added for clarity). 544 hand images of 17 different subjects were collected.

Procedure

Using our database of 544 collected hand contours, we simulate six scenarios which differ in the numbers of known and unknown users. Here, a known user is someone which has registered with the system; an unknown user is someone who has not provided any hand contour information before. Ideally, a known user is identified correctly while an unknown user is rejected.

For each scenario, we generate 100 sets of randomly drawn known and unknown users. In turn, we perform a 100 trial cross validation with a stratified random selection of training samples for each of these sets, resulting in $100 \times 100 = 10,000$ trials per scenario of known/unknown users. In each trial, we train the classifier using only training samples of the known users. Testing samples of known and unknown users are presented to the classifier afterwards.

We use LIBSVM [4] with probability estimates as SVM implementation in this evaluation. Multi-class support for SVM is realized by training one classifier for each user and employing a one-against-all strategy. That is, a separate classifier is trained for each known user. This user provides samples for the positive class, while samples of the other *known* users are used for the negative class. The initially separate classifiers are then combined into a joint classifier.

During testing, the best-scoring classifier determines the result, that is the identified user. A user is rejected as unknown if the reported score is below a certain threshold. For evaluating the identification performance, the joint classifier is provided with test samples of known and unknown users. For each test sample, we record the classifier's reported score (i.e., the probability estimate of the best scoring single classifier) together with the classification result (i.e., correct or incorrect). In the end, we merge all results to create a single ROC curve and calculate the AUC value for each scenario.

Results

		Known Users		
		5	10	15
Unknown Users	0	0.999	0.999	0.998
	5	0.990	0.995	×
	10	0.987	×	×

Table 1. AUC comparison (a value of 1.0 is equivalent to a perfect identification performance). Not all combinations are possible due to our limited database of 17 users.

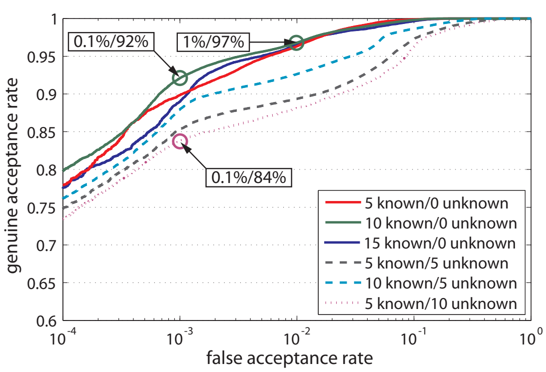


Figure 6. ROC comparison (tradeoffs between genuine and false acceptance rates as a function of classifier threshold). Three example rates are pointed out.

Figure 6 combines the six ROC curves into a single diagram for comparison. Table 1 lists the resulting AUC values for the six different scenarios of known/unknown users (higher values mean a better performance, with 1 being equivalent to perfect identification). The best performance is achieved for scenarios without unknown users, that is scenarios where only registered users can access the system. The performance varies only slightly for the three tested numbers of known users (5, 10, and 15). With an increasing number of unknown users the identification performance slowly decreases.

These results suggest that HandsDown enables robust identification for small user groups. Depending on application domains, the classifier's threshold can be adjusted to meet different security requirements, or rather tradeoffs between genuine and false acceptance rates. For example, for the scenario of 10 known and 0 unknown users, we can achieve genuine acceptance rates of 92% and false acceptance rates of 0.1%, or, with a different threshold, of 97% and 1%. If the aim is to achieve a false acceptance rate of 0.1% for the scenario of 5 known and 10 unknown users, the expected genuine acceptance rate is 84%, for example.

INTERACTION TECHNIQUES AND DESIGN SPACE

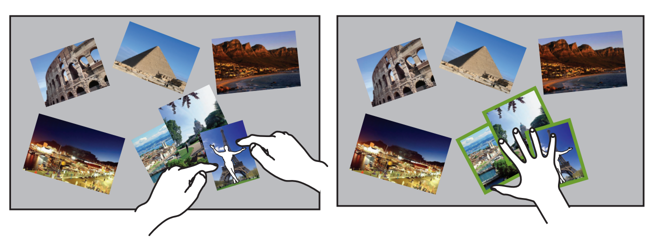
HandsDown enables many opportunities for novel interactions in surface computing environments. In this section, we illustrate and discuss example interaction techniques where HandsDown can be applied to enhance application benefits. All presented examples can be realized with a single camera setup. That is, they do not require an additional overhead camera to track hands above the surface.

Self/Group Identification

Users Tagging

The notion of tagging people's identities to digital files provides systems the advantage of associating files with specific users. In applications like photo albums, tagging is particularly useful; it affiliates photos with users' identities, and, later, it is convenient for photos searching by people's identity. For photo albums applications, HandsDown can be adopted as an interaction technique for users to tag themselves.

Figure 7 illustrates an example of a user tagging his/her identity to a group of photos. A user uploads a photo album onto a surface device. While browsing through the photos, users find pictures that they took part in. To tag themselves, they place a registered hand on the pictures. The system automatically recognizes the hand and attaches the user's identity to the pictures.



(a) Photo browsing (b) Photo tagging

Figure 7. Photo tagging. (a) illustrates a user browsing a photo album on an interactive surface and (b) shows the user placing their registered hand on top of the selected photos to tag their identity.

Unlike traditional photo tagging (by inserting textual names), HandsDown is a much faster, natural and intuitive approach.

Moreover, HandsDown can provide additional security where only registered users can tag themselves. A third person cannot tag another user, hence giving users the control of self-tagging.

Attaching Identities to Tangible Objects

Besides gesture inputs by hands, many literatures have covered the idea of mixing tangible interfaces into interactive surfaces for combine physical and virtual modalities (e.g. [11, 13]); thus, giving virtual identities to physical representations. Similarly, in many applications, it is useful to attach user identities with tangible objects. For example, multiplayer-games often require physical objects to give representations of the players. HandsDown provides a natural approach where users can effortlessly attach their identities to tangible objects on an interactive surface. Simply placing an object and a registered hand on the surface next to each another, the surface can establish an identity association between the two. Furthermore, this technique enhances applications like BlueTable [29] or PhoneTouch [21]. They allow users to pair mobile phones with an interactive surface by simply placing the phones on the surface, or touching the surface with the phone. We propose to combine these techniques with HandsDown (Figure 8); during device pairing, users can place their hand along with the mobile device on the surface to attach their identity. This can further be extended for access control, where only registered users have the authority to pair mobile phones with the surface.

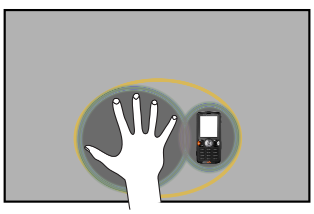
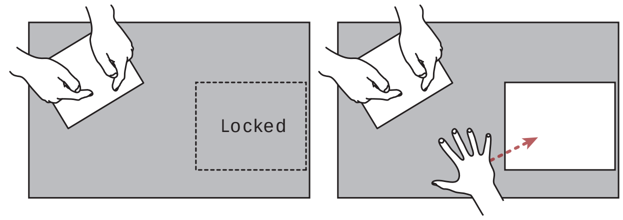


Figure 8. An illustration of a user performing HandsDown to attach their identity to a mobile phone on an interactive surface.

Access Control

Multiple Personal Spaces on a Shared Surface

Large interactive surfaces can simultaneously accommodate interactions of multiple users; in other words, a surface can be shared amongst multiple users, whilst each user still retains a personal workspace on the surface. Nevertheless, the sharing of a surface space requires security to protect the users' data. Although users are sharing the same input surface, they may not be willing to share their access rights to their private documents, like viewing, editing or copying, with other users. To protect the users' documents, we propose the use of HandsDown for administering access to personal data. When someone wants to access their own files, the person must first identify (or authenticate) to the system by performing a HandsDown gesture. Once verified, the system grants the user access to personal files. One could argue that using passwords is as effective; however, we must consider that password authentication on a large input device is vulnerable to shoulder-surfing attacks and password authentication also requires user memorability to retain passwords.



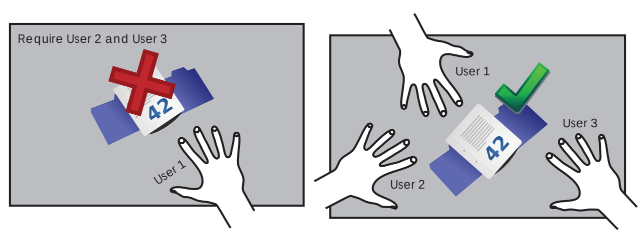
(a) Locked workspace (b) A user unlocks their workspace

Figure 9. Illustrations of users sharing a surface and a user reactivating a personal workspace using HandsDown.

Further suppose that a user working on a surface wants to rest for a short moment. The user minimizes or locks the personal workspace, while other users continue working with their workspaces (Figure 9(a)); this concept is similar to the "log off user" function on personal computers. When the user returns, the user can reactivate the workspace by placing their hand on the surface (Figure 9(b)). Thus, giving users the flexibility of walking away from and returning to the surface at any time.

Accessing Group Files and Documents

In collaborative settings, often users create files and documents that belong to groups; instead of single-user ownership, editing and copying of documents may require consents from all of the group members. We propose the notion of using HandsDown to achieve group access of information on interactive surfaces. We illustrate our idea with the following example (Figure 10): a group of users is collaboratively working on a project on a tabletop. Every user has right to access the project documents for viewing; however, when editing or copying is required, all members of the group must be present. To activate editing, all of the project members place their hands on the surface table simultaneously to identify themselves. Once the table confirms all group members are co-present, it enables the functions for document-editing.



(a) Access Denied (b) Access Granted

Figure 10. Group access. (a) shows a single user, User 1, who tried to access a group file and received an access denied message, whilst (b) shows a group of authorized users accessing the group file.

Using HandsDown for group access has the following implications: (1) all group members must be co-present (same time) and co-located (same place) to access group files; (2) neither password nor access code are required, instead users' physiological properties (or biometrics) are used for accessing files.

User Authentication and Personal Password Vault

When a user wants to access personal services via an interactive surface, like E-mails or websites, the user is required to enter a password. However, if people are around, it is extremely difficult for the user to enter a password privately without others seeing the entry. Kim et al. [12] have devised various password entry methods that are resistant to observation attacks. Nevertheless, although their techniques are obfuscated, they still require users to enter passwords directly on the surface; thus, from a visible angle or with the adequate recording equipment, it is possible to determine users' passwords by analyzing the entries.

Instead, we propose the adoption of HandsDown as a user authentication technique. Two design approaches of HandsDown authentication are possible. One approach is using users' hand contours directly as biometrics; hence, adopting HandsDown as a biometric authentication, as described in section *Identification vs. Authentication*. Alternatively, users can first choose and securely store a list of passwords within the surface device's database (like a vault of users' passwords) during registration. When login is required, the user performs a HandsDown gesture, and then the surface uses the hand contour as an index to retrieve the user's password. Once retrieved, the password is automatically inserted into the login system. Thus, no password is entered or revealed during login. From a user's perspective, the login interactions of both approaches are identical; they differ in registration, as the second approach requires users to choose the passwords they want to store.

Personalization and Customization

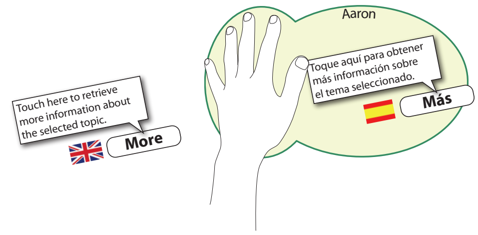
Personalized Interface Access

Large situated devices can be shared amongst many users. With systems that offer services based on users' preferences, we can anticipate personalization and customized interfaces on interactive surfaces. We propose the use of HandsDown as an interaction technique for users to recall their personalized settings on interactive surfaces.

After performing a HandsDown gesture, the surface recognizes the user and displays a user-dependent menu at the same location. Such a menu may include shortcuts to frequently used applications, the user's favorite websites, a list of recently edited files, or personal play lists, for example. In doing so, users cannot only conveniently access applications and data form anywhere on the surface, they also have a personalized user interface at hand.

Identity-aware Lenses

To enable dynamic personalization in an explorative manner, we propose identity-aware lenses, a concept related to a magic lenses [2]. After performing a HandsDown gesture, the user is identified and a lens appears next to their hand. The lens moves together with the hand. As long as the hand stays on the surface, the lens is shown. On lifting up the hand, the lens disappears. We envision the lens to be shaped and positioned in a smart way, being aware of the table's edges, for example.



(a) Regular view (b) Personalized appearance

Figure 11. Upon identification, a magic lens appears next to the hand. Here, it personalizes the interface appearance by translating text to the user's preferred language.

Any user interface element underneath the lens can adopt its appearance, content, and function to the user currently inspecting it. A text element is translated to match the user's language preference (Figure 11), for example, allowing international users to use a shared application simultaneously. In another example, a shared list of bookmarks shows the user's personal collection upon moving the lens over it.

Additionally, finger input performed within the lens area can be attributed to the identified user. Activating a button through a lens can invoke user-dependent functions, such as per-user undo. This style of interaction, where the non-dominant hand performs the HandsDown gesture and the dominant hand executes some finger interactions inside the lens, can be thought of as an example of asymmetric bimanual interaction. Here, the non-dominant hand sets the spatial frame of reference (embodied by the lens) for the actions of the dominant hand. This division resembles most skilled manual activities and hence leverages learned behavior [9].

DESIGN CONSIDERATIONS

HandsDown introduces many new opportunities for surface computing applications. Yet, before exploring this concept, the contexts of the application must be considered. Although the HandsDown technique is applicable in any devices that can read users' hands, it may also not be suitable in certain scenarios; for example, since hands registration is required before users can use the technique to identify themselves, the approach is not suitable for systems where new users are frequently introduced or users' identities require anonymity.

The following categories are crucial and must first be considered before planning adoption of HandsDown as an interaction technique for any surface type application.

Identification vs. Authentication

The shapes and sizes of people's hands vary greatly, depending on age, gender, or genetic diversity; hence, different people have different hand contours. As a result, the contour of a hand is conceptually a user's property, that is a *physiological biometric*. This biometric can be used for *identification* as well as *authentication* of users; users can be recognized by the contour shape of their hands (i.e., identification), alternatively the users can confirm claims of their identities by showing their hands (i.e., authentication).

In the HandsDown approach, the simple gesture of a user placing their hand on an interactive surface allows the user to achieve both identification and authentication. Anyhow, we do not recommend designers to adopt the approach for both schemes as a combine. The uniqueness of hand contours is not guaranteed, hence a false identification of a user's hand subsequently implies a false authentication. As a result, we recommend HandsDown is only used for either identification or authentication, but not both at once. Further consideration of robustness between identification and authentication is recommended. Results in [30] show the performance of using hand contours for authentication is more robust than identification. Each hand identification requires a search against the list of stored hands templates to find a specific identity; whilst, authentication only requires comparison between a hand input against a specified template.

Users Cardinality

Although the properties of hand contours vary greatly amongst different people, it is yet insufficient to claim that any hand contour is exclusively unique from the rest of the entire world. Instead, HandsDown is most suitable for scenarios where the size of the population is small.

Physical Privacy

Most interactive surfaces (like tabletops or digital whiteboards) are situated devices; consequently, the fixed locations of the devices mandate where users access the systems. In a public context, anyone who has access to the location can also read the user's input. Hence, when user authentication is required, the user is vulnerable to shoulder-surfing attacks; an adversary can easily record passwords entered by the user. HandsDown is a suitable alternative for resisting shoulder-surfing; adversaries cannot record any password since no secret information is entered (because the system examines the contour of the user's hand instead of reading input from the user directly). In a private context, security can be more lenient. We can assume users are less vulnerable. Ideally, only trusted people have physical access (e.g. in a home environment); thus, using HandsDown for user identification in a private space is sufficient. Certainly, public systems can also adopt HandsDown for user identification; however, alternative shoulder-surfing resistant authentication schemes, like the techniques in [12], are required.

Home environments are the prime settings for the adoption of HandsDown. In general, there are a small number of people in a household; thus, recognition and verification of household members are relatively straightforward. Furthermore, HandsDown is designed for low risk environments; home settings are physically secure and private, they therefore already provide a certain level of required physical security.

Social Context

Interactive surfaces are designed for multiple users to interact simultaneously. Unlike tradition personal computing, surface computing has no restriction of a singular user per device. For this reason, designers must anticipate the social context of multiple users interacting cooperatively.

An interactive surface can indubitably be a standalone personal device operated by one person, in which case, a single user dominates the entire surface space; that is the user can perform a HandsDown gesture anywhere on the surface. However, when multiple users are involved, that is in a collaborative setting, social aspects must be considered. In group scenarios where users operating around an interactive surface, although the surface space is shared, each user occupies a territorial personal space and they must obey social protocols as well as policies to avoid conflicts or intrusions into another user's space [15]. Thus, designers should create systems that a user should never need to reach or move into another user's space to perform a HandsDown gesture on a surface.

Touch Areas

There are numerous ways for users to perform inputs on an interaction surface, for example by means of finger touches, tangible objects, or external pointing devices. In our case, palm touches are used. To capture a palm-touch input on a surface, the input requires an available area proportional to the size of a hand on the surface.

Finger touches generally require small surface areas, whereas palm touches occupy much larger regions on a surface device. Designers should avoid displaying information beneath the area where users should place their hands, leave sufficient space for input, prohibit overlapping, as well as display visual cues to indicate locations where users can place their hands. In addition, in collaborated scenarios where multiple hand contours are inserted, they require further consideration, like whether the interaction requires users to place their hands down simultaneously or individually.

DISCUSSION

The previous interaction techniques and design considerations show that the concept of using HandsDown gives rise to a wide spectrum of compelling uses for surface computing. The advantages of using HandsDown for surface applications are threefold:

No user instrumentation or additional hardware is required. Other surface systems capable of identifying users, such as DiamondTouch [6], require special hardware, or demand the user to wear additional equipment. From a user's perspective, the use of extra hardware to identify may be considered counterintuitive, unnatural, or laborious. HandsDown overcomes these issues by eliminating the use of any additional devices on the users; instead, natural physiological features of the user's hands are used.

Instantaneous and fluid interactions. The simple gesture of users placing their hands on an interactive surface allows the identification process to execute fluidly. When a hand shape is detected, the system automatically analyzes the features of the hand. Our implementation shows the entire identification process, from capturing the image of the user's hand to identifying the user, occurs within a second. Multiple users can be identified simultaneously.

No memorization is required. The use of hand contours provides users the benefit of not needing to memorize any identification information. Since the users' hands represent their identities, the identification features are carried with the users at all time. This is one of the benefits of using biometrics, because the features are consistent even over a long time. However, although this is a usability advantage, it is a security disadvantage. Once a biometric is forged, it remains stolen for life; there is no getting back to a secure state [23]. Compared to fingerprints, hand contours are less distinguishable; they are less appropriate to identify an arbitrary person. As a consequence, we anticipate users to be less concerned about providing hand contours to the system for enrollment.

It is important to note that we do not propose HandsDown as a general user identification technique in a sense where every touch can immediately be associated to a user. The use of an additional overhead camera for hand tracking above the surface comes close to such a scenario, in theory: After a hand is identified, it is continuously tracked; any touch originating from it can be associated to an user identity. However, our observations showed that users frequently leave the surface area, for example to relax their hands, to point at something in the room, or during a discussion with other users; this leads to a loss of tracking. As users are generally not aware of the surface boundaries, they do not realize leaving the trackable area, and hence to not anticipate that they have to put down a hand to identify again.

CONCLUSION

Currently, most interactive surface systems are not capable of distinguishing the identity of users. In this paper, we introduced HandsDown, a novel user identification technique for interactive surfaces, which adopts the concept of using hand contours to recognize or authenticate users. In our system, users are required to perform a quick one-time enrollment of their hands. To enroll, the users place their hands on various locations on a surface device and silhouette images of the hands are captured as reference templates. Once enrolled, a user can naturally place their hand on any arbitrary location on the surface, and the system recognizes the user's identity. Unlike systems that require users to place their hands on an external device (like a dedicated flatbed scanner) for identification, our system allows direct hand interactions on the surface.

As a proof of concept, we have implemented HandsDown on an interactive tabletop. The evaluation of our system shows, with a relatively small number of registered users (seventeen subjects), that HandsDown is substantially robust. The significance of HandsDown lies in its capability to integrate with current surface applications to enrich their overall systems with robust and instantaneous user identification. To demonstrate the potentials of HandsDown, we have suggested and discussed a series of interaction technique where HandsDown provides benefits. Furthermore, we presented categories of design considerations for designers and researchers; the categories must be pondered before determining the adoption of HandsDown as an interaction technique for any interactive surface applications.

REFERENCES

1. E. Bier and S. Freeman. MMM: a user interface architecture for shared editors on a single screen. In *Proc. UIST '91*, pages 79–86, 1991.
2. E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. Derose. Toolglass and magic lenses: The see-through interface. *Computer Graphics*, 27(Annual Conference Series):73–80, 1993.
3. G. Boreki and A. Zimmer. Hand geometry: a new approach for feature extraction. In *Automatic Identification Advanced Technologies '05. Fourth IEEE Workshop on*, pages 149–154, 2005.
4. C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.
5. C. T. Dang, M. Straub, and E. André. Hand distinction for multi-touch tabletop interaction. In *Proc. ITS '09*, pages 101–108, 2009.
6. P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In *Proc. UIST '01*, pages 219–226, 2001.
7. 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 '08*, pages 297–302, 2008.
8. T. Fawcett. ROC graphs: Notes and practical considerations for researchers, 2004.
9. Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior*, 19(4):486–517, December 1987.
10. A. K. Jain, A. Ross, and S. Pankanti. A prototype hand geometry-based verification system. In *Proc. AVBPA '99*, pages 166–171, 1999.
11. S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reactable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proc. TEI '07*, pages 139–146, 2007.
12. D. Kim, P. Dunphy, P. Briggs, J. Hook, J. Nicholson, J. Nicholson, and P. Olivier. Multi-touch authentication on tabletops. In *Proc. CHI '10*, pages 1093–1102, 2010.
13. D. Kirk, A. Sellen, S. Taylor, N. Villar, and S. Izadi. Putting the physical into the digital: issues in designing hybrid interactive surfaces. In *Proc. BCS HCI '09*, pages 35–44, 2009.
14. Microsoft Corp. Surface. 06 May 2010. `http://www.microsoft.com/surface`.
15. M. R. Morris, K. Ryall, C. Shen, C. Forlines, and F. Vernier. Beyond "social protocols": Multi-user coordination policies for co-located groupware. In *Proc. CSCW '04*, pages 262–265, 2004.

16. B. Myers, H. Stiel, and R. Gargiulo. Collaboration using multiple PDAs connected to a PC. In *Proc. CSCW '98*, pages 285–294, 1998.

17. 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 '04*, pages 1441–1444, 2004.

18. 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 '05*, pages 1124–1128, 2005.

19. R. Sanchez-Reillo, C. Sanchez-Avila, and A. Gonzalez-Marcos. Biometric identification through hand geometry measurements. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1168–1171, 2000.

20. D. Schmidt. Design and realization of an interactive multi-touch table. Technical report, Lancaster University, 2009.

21. D. Schmidt, F. Chehimi, E. Rukzio, and H. Gellersen. PhoneTouch: A technique for direct phone interaction on surfaces. In *Proc. UIST*, 2010.

22. D. Schmidt and H. Gellersen. Show your hands: A vision-based approach to user identification for interactive surfaces. Poster ITS, 2009.

23. B. Schneier. Inside risks: the uses and abuses of biometrics. *Commun. ACM*, 42(8):136+, 1999.

24. J. Schöning, J. Hook, T. Bartindale, D. Schmidt, P. Olivier, F. Echtler, N. Motamedi, and P. Brandl. Building interactive multi-touch surfaces. In C. Müller-Tomfelde, editor, *Tabletops - Horizontal Interactive Displays*. Springer, 2010.

25. J. Schöning, M. Rohs, and A. Krüger. Using mobile phones to spontaneously authenticate and interact with multi-touch surfaces. In *Proc. PPD '08*, 2008.

26. S. Scott and S. Carpendale. Guest editors' introduction: Interacting with digital tabletops. *Computer Graphics and Applications, IEEE*, 26(5):24–27, 2006.

27. S. D. Scott, S. Carpendale, and K. M. Inkpen. Territoriality in collaborative tabletop workspaces. In *Proc. CSCW '04*, pages 294–303, 2004.

28. J. Stewart, B. Bederson, and A. Druin. Single display groupware: a model for co-present collaboration. In *Proc. CHI '99*, pages 286–293, 1999.

29. A. D. Wilson and R. Sarin. BlueTable: connecting wireless mobile devices on interactive surfaces using vision-based handshaking. In *Proc. GI '07*, pages 119–125, 2007.

30. E. Yörük, E. Konukoglu, B. Sankur, and J. Darbon. Shape-based hand recognition. *Image Processing, IEEE Transactions on*, 15(7):1803–1815, 2006.

31. R. L. Zunkel. Hand geometry based verification. In A. K. Jain, R. Bolle, and S. Pankanti, editors, *Biometrics*, chapter 4, pages 87–101. Springer US, Boston, MA, 2002.