Bootstrapper: Recognizing Tabletop Users by their Shoes

Stephan R. Richter, Christian Holz and Patrick Baudisch
Hasso Plattner Institute, Potsdam, Germany
stephan.richter @ student.hpi.uni-potsdam.de
{christian.holz, patrick.baudisch} @ hpi.uni-potsdam.de

ABSTRACT
In order to enable personalized functionality, such as to log tabletop activity by user, tabletop systems need to *recognize* users. *DiamondTouch* does so reliably, but requires users to stay in assigned seats and cannot recognize users across sessions. We propose a different approach based on distinguishing users' *shoes*. While users are interacting with the table, our system *Bootstrapper* observes their shoes using one or more depth cameras mounted to the edge of the table. It then identifies users by matching camera images with a database of known shoe images. When multiple users interact, Bootstrapper associates touches with shoes based on hand orientation. The approach can be implemented using consumer depth cameras because (1) shoes offer large distinct features such as color, (2) shoes naturally align themselves with the ground, giving the system a well-defined perspective and thus reduced ambiguity. We report two simple studies in which Bootstrapper recognized participants from a database of 18 users with 95.8% accuracy.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces: Input Devices and Strategies, Interaction Styles.

Keywords: tabletop systems; user identification; indoor tracking; touch; personalization; Microsoft Surface.

INTRODUCTION
In order to enable personalized functionality and menus, to track achievements in multi-player games, or to enforce social protocol [15], a tabletop system needs to be able to know which touch belongs to whom.

Electronic rings [16] allow identifying touches reliably, but require maintenance, such as recharging batteries. Alternatively, *DiamondTouch* identifies users based on the chairs they sit in [5]. However, this only identifies chairs, thus users still need to identify themselves at the beginning of each session. When dealing with users who are likely to move around, such as children in a free-flow learning environment, these limitations are problematic.

Researchers have therefore explored how to distinguish users directly. Schmidt et al distinguish users' hands using the table's built-in camera [17], but to see a sufficient number of features, the approach requires specific hand postures which can interfere with the interaction. Alternatively, face recognition recognizes users reliably [1], but requires a frontal high-quality image of users' faces, which interferes with users moving around the tabletop.

In this paper, we propose a new perspective on the problem: instead of distinguishing users' hands or faces, we distinguish their *shoes*.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

BOOTSTRAPPER
Figure 1 shows our prototype system *Bootstrapper*. Its main element is a depth camera (*Microsoft Kinect*), which is mounted at each side of a diffuse illumination [10] multi-touch table (*Microsoft Surface*). The depth cameras point downwards at the space where users stand, which is illuminated homogenously by lights located inside the base.

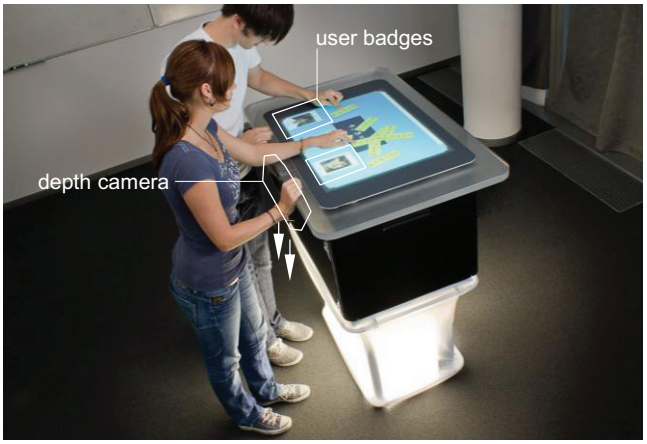

Figure 1: Bootstrapper recognizes users interacting with the table by observing their shoes using depth cameras.

Bootstrapper works as follows. (1) It observes the space where users stand. (2) It extracts shoes from the video stream by thresholding the depth image. (3) It matches the shoes against a database of shoe-user pairs and retrieves the user's name and associated meta information. (4) If Bootstrapper sees more than one user, it pairs up touches with shoes based on the orientation of users' hands, as observed by the table's built-in camera.

Walkthrough
Figure 2 shows how Bootstrapper tracks students' progress with an educational software package. (a) Stephan approaches the table. (b) Bootstrapper finds his shoes in the database and (c) displays Stephan's badge with his current math score. (d) Whenever Stephan solves a math puzzle, the math score on his badge increases. (e) Stephan is joined by Caro, a first-time user. (f) Since Bootstrapper cannot

find Caro's shoes in its database, it automatically creates a placeholder badge for her (with a photo of her shoes).

Figure 2: Walkthrough: Bootstrapper recognizes users to track their math achievements. Finally, Bootstrapper creates an overview for the teacher.

Both users keep practicing and Bootstrapper adds their achievements to their respective badges. (g) Caro personalizes her badge by adding a photo. Stephan can leave and come back anytime and is recognized. The next day, however, he is wearing a new pair of shoes. Bootstrapper does not recognize him and therefore (h) creates a placeholder badge for him. (i) Stephan taps on the placeholder badge which opens the list of existing badges. He selects his own badge from the list, which merges his accounts and adds his new shoes to his existing account. (j) Finally, their teacher accesses the table. Inspecting the performance of his students with the individual exercises helps him understand how to best support each one of them.

BENEFITS AND LIMITATIONS

Bootstrapper supports walk-up use and persists profiles across sessions. Unlike approaches based on hand recognition, users hands are free to interact. Unlike traditional biometric features, such as fingerprints or pupils, shoe recognition can be implemented with comparably coarse hardware. The reason is that shoes offer *large* visual features, such as distinct colors, seams, laces, logos, or stripes. These can be recognized from a distance using inexpensive consumer cameras.

The recognition problem solved by Bootstrapper is comparably simple: unlike other parts of the human body, such as hands and faces, shoes maintain direct contact with the ground most of the time. This constrains shoes to translation and rotation *in the plane,* which simplifies the recognition problem (see section "algorithms").

Bootstrapper is subject to three limitations. First, two users wearing the same shoes will be assigned the same profile. This makes identity spoofing possible by buying the same shoes as someone else, which is inherent to clothing-based recognition. Second, Bootstrapper currently recognizes users only when at least one shoe stands flat on the ground. Future versions may alleviate this with a virtual viewpoint transformation. Third, users can mislead Bootstrapper's touch-to-user association by contorting their hands.

Due to these limitations, we see Bootstrapper being used primarily for signing into non-critical accounts, such as learner or gamer profiles or for interface personalization. In contrast, Bootstrapper is not appropriate for applications that require true security, such as to work with computer accounts or for accessing banking information.

CONTRIBUTION

Our main contribution is the idea to reformulate the user recognition problem as a shoe recognition problem. Based on this, we demonstrate a hardware setup and an algorithm that associates shoes with touch input and that locates a user's shoes in the user database. Our recognition reaches 95.8%, which is sufficiently accurate for groups the size of a lab group, school, or kindergarten class (see user study). Bootstrapper successfully associates users with their touches in 92.3% to 100% for 5 to 1 concurrent users.

RELATED WORK

The work in this paper is related to user identification on multitouch systems and to shoe recognition.

Researchers have proposed identifying users with unique tokens, such as by combining optical recognition with RFID tags [12], using a ring that flashes a unique ID sequence to the table [16] or combining an electronic wristband with an analysis of hand orientation [11]. Instead of instrumenting users, *DiamondTouch* instruments the chair, i.e., an object that users touch only during the session [5].

A different track of research has explored the use of biometrics for user identification. Several researchers expect fingerprints scanning to eventually be integrated into multitouch hardware [6, 18]. Identifying users by their hands, in contrast, demands that the whole hand be placed on the surface [17]. Face recognition achieves high success rates, but requires users to directly face the camera [1].

Other researchers proposed to distinguish users based on shoes or gait. *Smart Floor* identifies users by observing the forces and timing of the individual phases of walking [14]. *Multitoe* [3] identified users by their sole patterns. Both require an instrumented floor.

Similar to *Bootstrapper, Medusa* locates users around a tabletop computer and associates touches with them [2], using a multitude of proximity sensors. This approach does not afford user identification, however.

ALGORITHMS: IDENTIFY USERS & PAIR UP TOUCHES

To recognize users and identify who touched where, Bootstrapper performs the following four steps.

Step 1: Extracting shoe textures from the camera image

First, Bootstrapper extracts shoes from the image as shown in Figure 3. (b) Bootstrapper retrieves the depth image, and (c) by thresholding it, removes the background. (d) Find-

ing *connected components* [13] separates bodies. Feet connected to the same hips are identified as belonging to the same user. (e) Thresholding depth at 6" above the background removes legs. (f) Masking the RGB image with the registered depth image extracts shoe textures.

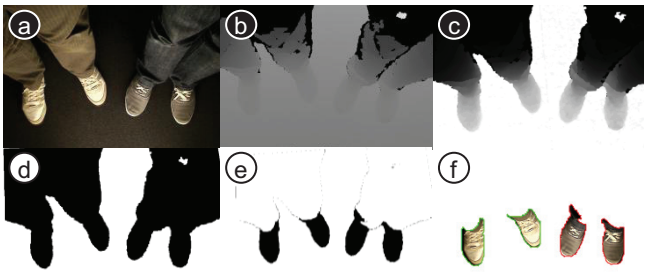

Figure 3: (a) RGB image, (b) raw depth image, (c) background subtracted, (d) bodies located, (e) leg removed. (f) RGB masked with depth image.

Step 2: Looking up matching shoes in user database

To identify users, Bootstrapper compares the RGB shoe image with all shoe images in its database. To obtain a better quality color image than provided by Kinect, we complemented Bootstrapper with a separate webcam. We currently employ two separate matching functions: SURF [4] and *color histograms* matching using *Bhattacharyya* distance. Our system uses the SURF implementation provided by OpenCV 2.3 [13] with hessian threshold set to 200 for additional features and default parameters otherwise. We refine the feature extraction using clustering and a probabilistic verification [9]. Earlier versions of Bootstrapper used *SIFT* and *Earth movers distance* on color and depth histograms, but were too slow for real-time use.

If none of the shoes in the database reach an empirically determined threshold for a period of 5 frames, Bootstrapper creates a new user and displays a placeholder badge.

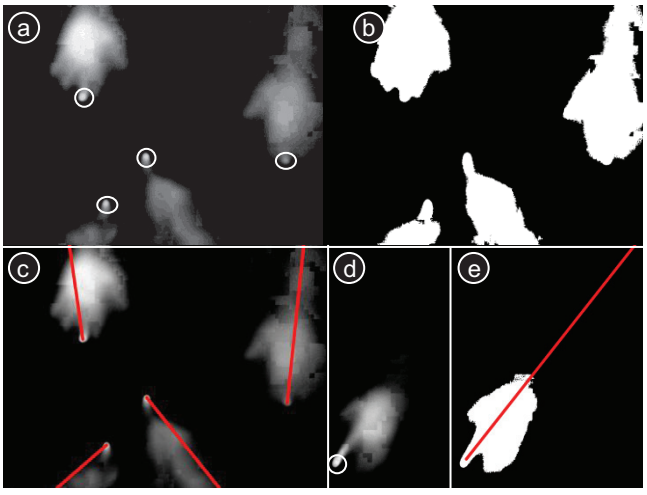

Figure 4: Bootstrapper traces touches to an edge of the table by following the image of the user's arm.

Step 3: Tracing touches to the table edge & to the user

If Bootstrapper sees only one user, it associates all observed touches with this user. If it detects several users, it locates hovering arms in the tabletop image and traces them back to the edge of the table.

Figure 4 illustrates the processing pipeline. (a) Bootstrapper obtains the raw image of the touch surface and a list of all touch locations from the tabletop system. (b) Bootstrapper locates hovering hands and arms by thresholding brightness with a low adaptive threshold [8]. (c) If the arm extends all the way to the edge of the table, Bootstrapper uses the intersection between arm and edge. (d) If an arm does not extend all the way to a table edge, Bootstrapper extrapolates linearly from the farthest point inside the blob (e).

Step 4: Applying offsets for left and right hands

Associating an observed arm with a user may still be ambiguous if two users stand side-by-side (Figure 5). *Bootstrapper* resolves this by determining whether the observed hand is a left or a right hand (as in [8]) and applying a hand-specific offset. This resolves the ambiguity.

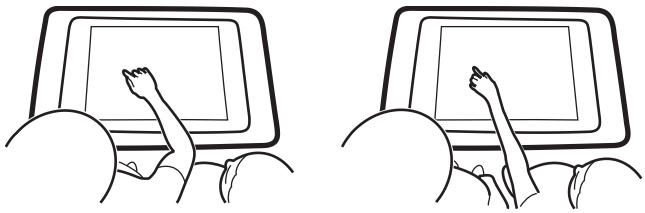

Figure 5: Ambiguity: which user caused this touch?

Finally, Bootstrapper associates the user closest in terms of radial distance to the adjusted edge point to the touch.

EVALUATION 1: 95.8% ACCURACY WITH 18 USERS

We conducted a brief technical evaluation to determine the recognition rates of the different algorithms. We recruited 18 random participants (with random shoes, Figure 6) from our institution. Each participant interacted with a puzzle application for two minutes and *Bootstrapper* captured and stored 20 samples of their shoes at random times. A week later, we brought the same participants, wearing the same shoes back to the lab and repeated the procedure.

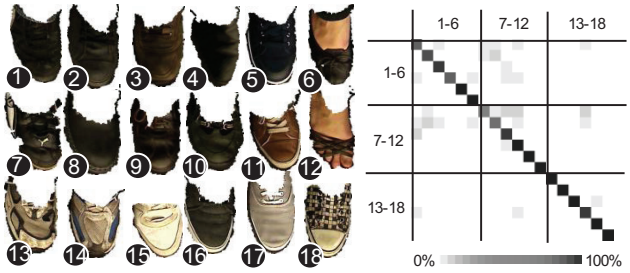

Figure 6: (left) Participants' shoes in the study. (right) Confusion matrix of *combined* matching.

Results: We performed a two-fold cross-validation of the two matching algorithms and their combination with all 720 captured images, extracting features from a single frame. In addition, we evaluated a majority voting across ten frames, which achieved the highest accuracy (Figure 7). Regarding speed, our implementation took 47ms for *histogram matching,* 2.5s for *SURF,* and 0.74s for *combined* on average. *Combined* is faster than just SURF, because we invoke SURF only when *histogram* returns a low confidence score. All results were calculated on a 2.2GHz Intel Core-i7 2720QM with 4GB RAM.

As expected, error rates varied across different types of shoes and dark shoes were the most error prone (#2 and #8 in Figure 6). Overall, however, recognition rates of 93.3% for a single frame and 95.8% across multiple frames are appropriate for a wide range of applications, such as the aforementioned scenario of tracking learning progress.

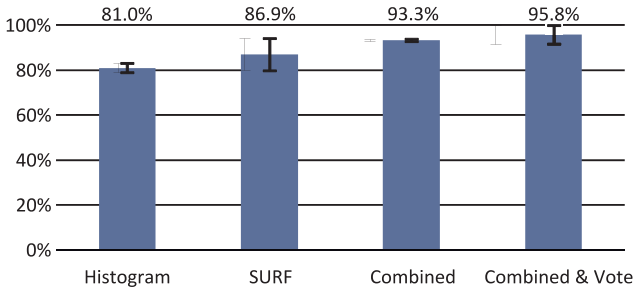

Figure 7: Recognition rates for SURF, color histogram matching, combined, and combined with majority voting over 10 frames on 18 users, based on a single observed frame

EVALUATION 2: TOUCH-TO-USER ASSOCIATION

To determine the accuracy of the touch-to-user association, we conducted a second study and recruited 13 new participants. During each trial, the participant stood at one of the 4 positions shown in Figure 8. They now acquired one of 5 different targets located on the screen using the right hand. The experiment comprised 3 blocks, resulting in 60 trials per participant and 780 overall trials. All targets and standing positions were selected randomly. We measured the angular distance between where our algorithm predicted the user and the participant's actual position.

Figure 8: Participants acquired 5 different targets on the table from 4 positions, repeating each combinations 3 times.

Results: The median angular error for our predictions was 32 degrees across all trials. Whether a touch is associated with the correct user depends on how tightly users are huddled around the table. By snapping the predicted location to the closest of 1, 2, 3, 4, or 5 users distributed evenly around the table, we obtained accuracy rates of 100%, 99.7%, 99.1%, 96.1%, 92.3%, respectively.

CONCLUSIONS AND FUTURE WORK

In this paper, we presented a new approach to recognizing users of a tabletop system. Our main contribution is the idea to reformulate the user identification as a problem of recognizing shoes. Unlike hands or faces, shoes can be observed for a constant perspective and they offer large visual features, which we can recognize reliably from a distance with consumer hardware.

For future work, we plan to adapt Bootstrapper to different form factors, including the traditional coffee table shape, to match shoes based on depth data, and to explore our approach in non-tabletop scenarios, such as to profile customers while shopping. We also plan to explore user recognition based on users' clothing using a single overhead camera, which will facilitate touch-to-user associations [7].

ACKNOWLEDGMENTS

We thank Paul Hoover and Caroline Fetzer for discussions.

REFERENCES

1. Abate, A.F., Nappi, M., Riccio, D., Sabatino, G. 2D and 3D face recognition: A survey. *Pattern Recognition Letters Vol.28, Issue 14*, 1885-1906.
2. Annett, M., Grossman, T., Wigdor, D., Fitzmaurice, G. Medusa: A Proximity-Aware Multi-touch Tabletop. *Proc. UIST'11*.
3. Augsten, T., Kaefer, K., Fetzer, C., Meusel, R., Kanitz, D., Stoff, T., Becker, T., Holz, C., Baudisch, P. Multitoe: High-Precision Interaction with Back-Projected Floors Based on High-Resolution Multi-Touch Input. *Proc. UIST'10,* 209-218.
4. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L. Speeded-up Robust Features. *Proc. Computer Vision and Image Understanding (2007)*, 346-359.
5. Dietz, P. and Leigh, D. DiamondTouch: a multi-user touch technology. *Proc. UIST '01,* 219-226.
6. Holz, C. and Baudisch, P. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. *Proc. CHI '10,* 581–590.
7. Holz, C. and Baudisch, P. Understanding Touch. *Proc. CHI '11*, 2501-2510.
8. Lepinski, J.G., Grossman, T., Fitzmaurice, G. The design and evaluation of multitouch marking menus. *Proc. CHI '10*, 2233-2242.
9. Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (2004)*.
10. Matsushita, N. and Rekimoto, J. *HoloWall*: Designing a Finger, Hand, Body, and Object Sensitive Wall. *Proc. UIST'97*, 209-210.
11. Meyer, T. and Schmidt, D. IdWristbands: IR-based User Identification on Multi-touch surfaces. Poster at *ITS 2010*.
12. Olwal, A. and Wilson, A. SurfaceFusion: Unobtrusive Tracking of Everyday Objects in Tangible User Interfaces. *Proc. GI '08,* 235-242.
13. *OpenCV*. http://opencv.willowgarage.com
14. Orr, R.J. and Abowd, G.D. The smart floor: a mechanism for natural user identification and tracking. In *CHI'00 Extended Abstracts*, 275-276.
15. Piper, A., O'Brien, Ringel Morris, M., and Winograd, T. SIDES: a cooperative tabletop computer game for social skills development. *Proc. CSCW'06,* 1-10.
16. Roth, V., Schmidt, P., and Güldenring, B. The IR Ring: Authenticating users' touches on a multi-touch display. *Proc. UIST '10*, 259-262.
17. Schmidt, D., Chong, M., and Gellersen, H. HandsDown: Hand-contour-based user identification for interactive surfaces. *Proc. NordiCHI '10,* 432-441.
18. Sugiura, A., and Koseki, Y. A user interface using fingerprint recognition: holding commands and data objects on fingers. *Proc. UIST '98,* 71-79.