YouTouch!
Low-Cost User Identification at an Interactive Display Wall

Ulrich von Zadow[1] Patrick Reipschläger[1] Daniel Bösel[1]
Anita Sellent[2] Raimund Dachselt[1]
[1]Interactive Media Lab, [2]Computer Vision Lab
Technische Universität Dresden
Dresden, Germany
firstname.lastname@tu-dresden.de

Figure 1: a) user-specific interface in a drawing application, b) debug view showing image and skeleton-based user data, c) features used for re-identification (left: image histograms and right/top: floor-shoulder distance, shoulder width)

ABSTRACT

We present YouTouch!, a system that tracks users in front of an interactive display wall and associates touches with users. With their large size, display walls are inherently suitable for multi-user interaction. However, current touch recognition technology does not distinguish between users, making it hard to provide personalized user interfaces or access to private data. In our system we place a commodity RGB + depth camera in front of the wall, allowing us to track users and correlate them with touch events. While the camera's driver is able to track people, it loses the user's ID whenever she is occluded or leaves the scene. In these cases, we re-identify the person by means of a descriptor comprised of color histograms of body parts and skeleton-based biometric measurements. Additional processing reliably handles short-term occlusion as well as assignment of touches to occluded users. YouTouch! requires no user instrumentation nor custom hardware, and there is no registration nor learning phase. Our system was thoroughly tested with data sets comprising 81 people, demonstrating its ability to re-identify users and correlate them to touches even under adverse conditions.

AVI '16, June 07 - 10, 2016, Bari, Italy

ISBN 978-1-4503-4131-8/16/06...$15.00

DOI: http://dx.doi.org/10.1145/2909132.2909258

CCS Concepts

•**Human-centered computing** → *User interface toolkits;*

Keywords

Display wall; multitouch; RGBD sensor; user identification; multi-user interaction; re-identification; interactive surface

1. INTRODUCTION

Large interactive surfaces such as tabletops and display walls are increasingly popular and, by virtue of their size, invite multi-user interaction. In this context, distinguishing between users is important because it allows personalized interaction (demonstrated, e.g., in the DiamondTouch project [8]). Additionally, knowledge of users' positions in front of wall displays allows rich interaction, among others enabling proxemic [2] and body-centric interaction (e.g., proposed with BodyLenses [17]). However, current interactive displays wall do not provide this information, so most research work in this field uses instrumentation or marker-based tracking (e.g., OptiTrack).

While there are a number of tabletop-specific systems that provide the user's identity (e.g., [6, 14, 22]), there is very little work that is applicable to large vertical displays. In particular, display wall users tend to move around the room during collaborative work [16], so camera-based systems need to deal with occlusion issues. Many application scenarios – especially in casual or public settings – also require support for true walk-up-and-use interaction, without an explicit ID step, training phase, or user instrumentation.

Our system uses low-cost, off-the-shelf hardware (a consumer RGB + depth camera) to enable this. From this input, state-of-the-art tracking systems such as the one integrated into the Microsoft

Kinect can establish the locations of users [28]. However, these tracking systems lose user identification whenever a person becomes occluded or leaves and re-enters the interaction space. Additionally, state-of-the-art tracking is not tailored for display wall interaction, so no component that correlates persons with touch events on the wall exists.

Our main contribution is a low-cost and reliable method for tracking users at a large display wall and associating touches with the respective users. We adapt and extend methods used in surveillance to re-identify (ReID) users after tracking loss. To this end, we use person descriptors containing color histogram data and skeleton-based biometric measurements (Figure 1c). Our novel TouchProcessor component associates touches with users and uses the skeleton data to determine the hand the user is touching with. Using stored skeleton data of past frames, even touches of users currently occluded can be assigned in a large majority of cases.

As the task we put our system to is quite challenging, we thoroughly evaluated it with a considerable number of test scenes. We acquired RGB + depth (RGBD) video and tracking data of a total of 81 subjects, with 36 single-person scenes used for optimization of the ReID component. For evaluation, we recorded the remaining 45 users in multi-person scenes involving numerous position swaps to stress the system. We evaluated three configurations of our system, allowing us to judge the effectiveness of different components. In addition, we manually inspected the remaining touch identification failures to find their causes. Finally, we implemented a sophisticated development and test toolset able to record and play back the data interactively, and including pause and single-step facilities to allow pinpointing and debugging of issues.

2. INTERACTION USING YOUTOUCH!

YouTouch! enables numerous application scenarios and effortless multi-user cooperation. Since the system has knowledge of the positions of users in front of the wall, proxemic [2] and body-centric [17] interaction becomes possible. YouTouch! makes areas with personalized views feasible, and the display can show personal data in close proximity to the user. For example, menus can automatically appear when a user approaches the wall and be placed at appropriate positions with respect to her. Since touches are also personalized, user specific modes and settings can be applied, and seamless transitions between proxemic and touch interaction become possible. YouTouch further enables, e.g., personalized clipboards (e.g., proposed by Rekimoto [23]) and interaction in conjunction with personal devices such as the SleeD [32].

To illustrate the possibilities, we implemented an example vector drawing program (Figure 1a) that integrates body-centric and touch interaction to deliver an intuitive and user-specific interface. The pen settings are user-specific; colors remain selected even if the user becomes occluded or leaves the room and comes back. The pen configuration dialog intelligently appears close to the user when she approaches the wall and follows her if her position changes significantly, thus always staying in an appropriate position. The dialog can be dragged, adding a user-specific offset to the position. We additionally implemented a user-specific undo; this is possible because the system knows who drew what. Finally, the hand recognition integrated into YouTouch! allows us to support hand-specific modes. In case of the drawing program, the right hand draws, while the left hand erases. In summary, YouTouch! allows us to support not only a user-specific pen configuration, but also user-specific menu positioning and a user-specific undo stack; furthermore, transitions between interactions in front of and on the wall are integrated seamlessly.

3. BACKGROUND AND RELATED WORK

There is significant work regarding the identification of users touching an interactive display, which we review in the following section. In addition, our work builds on previous research in person re-identification, also summarized below.

3.1 Touch User Identification

A large majority of work on touch user identification is in the context of interactive tabletops – summarized in Table 1.

A number of works require explicit registration followed by a machine learning step. Among these is Ramaker et al.'s Carpus [22], which identifies users via an overhead camera that tracks registered users' hands. Schmidt et al.'s HandsDown [27] uses hand contours for this purpose. Similarly, finger positions of 5-finger touches can be registered and used for identification [4]. Harrison et al. [13] distinguish registered users using the raw signal of a capacitive touch screen, while Richter et al.'s BootStrapper [24] uses a depth camera pointed at users' feet to recognize registered users. In all of these cases, the system first captures user information in a registration step. This information is then used to learn a classifier, which generally requires lengthy computation. The result is usually a very high identification accuracy, but the registration and learning steps prevent spontaneous interaction with unknown users.

It is also possible to distinguish users using additional worn or carried equipment. Holz et al.'s Biometric Touch Sensing [15] uses a biometric sensor armband for this purpose. With optical touch recognition, additional equipment can use pulsing LEDs to send ID data to the touchscreen. This is used in Meyer et al.'s IdWristbands [19] and Roth et al.'s IR Ring [25]. Schmidt et al. identify mobile phones touching a screen by correlating the time of the phone's touch with its accelerometer data [26]. While these systems have uses in planned situations with a limited number of users, the requirement for additional equipment is a hinderance in walk-up-and-use situations.

Conversely, most approaches that are usable in walk-up-and-use scenarios restrict the movement of the user. These include a number of works that use an overhead camera: Clayphan et al. [6], Murugappan et al. [20] and Thelen et al. [29] all track the user's

Paper	No Explicit Registration	No Worn/Carried Equipment	Commodity Hardware	Supports User Movement
Biometric Touch Sensing [15]	-	-	-	Y
BootStrapper [24]	-	Y	Y	Y
Capacitive Fingerprinting [13]	-	Y	-	Y
Carpus [22]	-	Y	-	Y
CollAid [6]	Y	Y	Y	-
DiamondTouch [8]	Y	Y	-	-
Extended Multitouch [20]	Y	Y	Y	-
Fiberio [14]	Y	Y	-	Y
Hand Tracking [9]	Y	Y	Y	-
HandsDown [27]	-	Y	-	Y
IdWristbands [19]	Y	-	-	Y
IR Ring [25]	Y	-	-	Y
Large Display Interaction [29]	Y	Y	Y	-
Medusa [1]	Y	Y	-	-
MTi [4]	-	Y	Y	-
PhoneTouch [26]	-	-	Y	Y
YouTouch!	Y	Y	Y	Y

Table 1: Comparison of tabletop-centric related work for touch user id.

hands using an RGBD camera but lose her ID when she leaves the tracked area. Similarily, Dohse et al. [9] use a conventional camera and hand color segmentation for the same purpose. To extend the interaction area, Medusa by Annett et al. [1] uses additional proximity sensors mounted at the side of a tabletop, but still loses the user's ID when she leaves the sensor range. The DiamondTouch system [8] instruments chairs, relying on the user forming a conduit between her chair and the tabletop at each touch.

Holz et al.'s Fiberio [14] is a touchscreen that is able to identify users biometrically using fingerprints. It is an exception in that it allows both spontaneous interaction by unregistered users and supports user movement. Unfortunately, it requires custom display hardware and a very high-resolution camera to support a small display area. There is also no support for interaction in proximity to the display.

In summary, shortcomings of these works with respect to our goals include a) an explicit registration step that prevents usage in walk-up-and-use scenarios, b) worn or carried equipment, c) the use of custom and/or costly hardware, and d) the assumption that users will not leave the tracking area or switch places.

We found only two works that cover user identification specifically tailored to wall displays, both Kinect-based and leveraging the ability of the Kinect to track users. Turnwald et al. [30] place the camera in front of the screen. While similar to our setup, their system loses the user's ID once the camera loses tracking (i.e., when users switch places), since there is no ReID component or occlusion handling. Chen et al. [5] place the Kinect at the side of a vertical screen and re-identify users using color histograms when tracking is temporarily lost. However, with larger displays, this setup suffers from occlusions. In contrast, we minimize occlusions through camera placement behind the users. Additionally, our rearward view allows for more robust skeleton fitting and tracking, which allows us to use skeleton data for ReID. Chen's paper focuses on interaction and therefore omits algorithm details; furthermore, there is no evaluation of their ReID performance.

In addition to a tracking component, the Kinect system [18] also provides a user recognition component that can make up for failures in skeleton tracking. Unfortunately, it relies on face recognition, and in the case of larger display walls there is no camera position that has a reliable view of the face, since interacting users stand close to the wall (Figure 1a).

3.2 Person Re-Identification

In computer vision, identifying people across different cameras (or if they exit and reenter one camera's field of view) is known as person re-identification (or ReID). This is an important issue in surveillance and therefore a large research field, with several overview articles available (e.g., [3, 7, 10, 12, 31]).

ReID algorithms generally work with appearance-based *features*, relying, e.g., on the ability to distinguish different persons' clothing. Features are extracted from segmented images and combined to construct *descriptors* that discriminate individuals. People are re-identified by matching *probe* descriptors from current camera images to a *gallery* of previously scanned descriptors using a model-based matching procedure. However, in contrast to our scenario, realtime re-identification and database building is not a requirement in most surveillance settings. Challenges in general ReID settings include varying lighting conditions, camera color calibration and the need to work with low-resolution images containing significant clutter. We leverage and repurpose these algorithms, adapting them to the specifics of our setup and to our realtime setting.

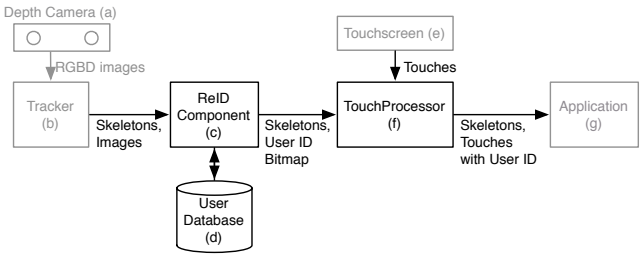

Figure 2: System Architecture, with pre-existing parts shown in grey. The main new parts are the ReID Component (c), which re-identifies users newly tracked and the TouchProcessor (f), which associates touches with users.

4. SYSTEM ARCHITECTURE

Our proposed system (Figure 2) relies on an RGBD camera (Figure 2, a) with an associated tracking component (b) to provide RGB and depth images as well as person tracking data. To maximize the tracked area while minimizing mutual occlusions between users, we place the camera several meters from the wall, facing the wall, and above head height.

The RGBD camera's tracking component delivers segmented images and user skeleton data (see Figure 8), but loses the user's ID whenever she is occluded or leaves the tracking area. The YouTouch! system consists of two main components that build on this and communicate using a simple network protocol:

- A *ReID Component* (Figure 2, c) that re-identifies users entering the camera's view using a User Database (d).
- A *TouchProcessor* (f) that uses the data provided by the ReID component to correlate touches (e) with users and delivers the results to the application (g).

Additionally, the system includes full recording and playback functionality to allow for efficient testing and evaluation of both components.

4.1 ReID Component

The ReID Component processes the tracker's images (including RGB and depth data as well as a segmented bitmap) and user skeleton data and uses it to build a user database in realtime. When the Tracker reports a new user, the ReID component attempts to correlate this user with the users in the database. An additional occlusion handling step remembers users that become untracked without leaving the tracking area, allowing optimized re-identification in this common case.

4.1.1 Algorithm

Re-Identification of users relies on a database of *person descriptors* that is generated and updated while tracking. Our person descriptors consist of *anthropometric features* (human biometric measurements such as height calculated from the skeleton data) and *color features* (histograms of person-specific image regions such as the torso). The similarity S between two person descriptors is calculated from the feature *correlation coefficients* $coeff_f$ and corresponding *weights* w_f (with w_f determined in an optimization step as detailed in the following section):

$$S = \sum_f w_f \cdot coeff_f$$

The anthropometric features we use are based on a candidate set used by Pala et al. [21] (Figure 5). From this set, we chose those

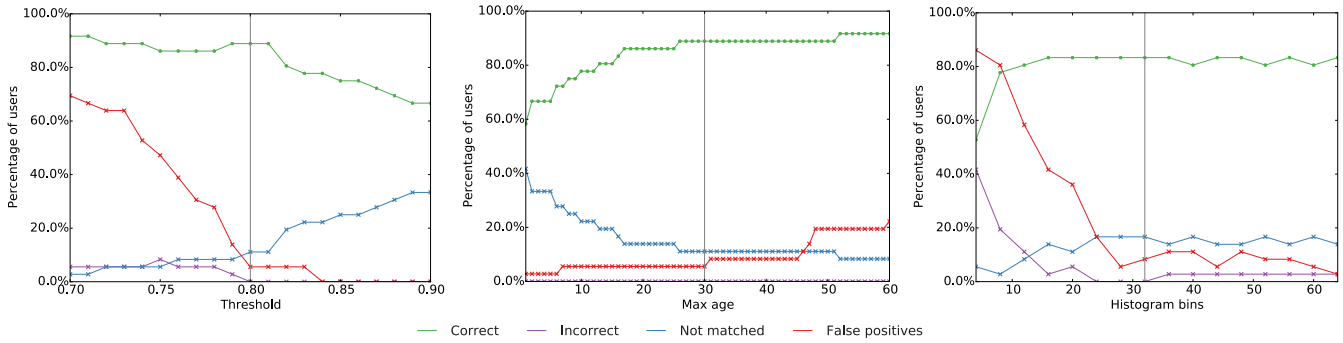

Figure 3: ReID performance for differing detection thresholds, ReID time limit in frames, and number of histogram bins. The vertical lines indicate the values used in the prototype.

that posessed the smallest intra-person variance (e.g., discarding arm length in this step) to maximize feature stability. Furthermore, in the case of multiple features that described a similar human measurement (e.g., height and floor-neck distance), we discarded all but one, maximizing feature independence. This left the floor-neck and floor-hip distances as well as the shoulder width as features (Figure 1c, right and top). Given descriptors i and j, we calculate the correlation coefficient $coeff_f$ of these features from the feature values d_i and d_j and the range of the feature $range_f$ in the training data set:

$$coeff_f = 1 - \frac{|d_i - d_j|}{range_f}$$

As color features, we use histograms calculated from the segmented images of different body parts. Candidate body parts were the legs, the torso, the head and the feet (Figure 1c, left). While related work (e.g., [11]) uses multiple color components, we obtained best results using exclusively the hue component of the HSV color space for the histograms. The number of *histogram bins* was determined in an optimization step (see the following section). Correlation between two histograms is determined using normalized cross correlation.

We introduce *descriptor states* to manage the lifetime of active users' descriptors (Figure 4). For each camera frame, all descriptors of people visible to the camera are processed. If the person is new, the descriptor is initially put in the UNKNOWN_TRACKED state. If not, the descriptor is updated with the current feature vector using a sliding average with a window of 100 frames (3 seconds) to accomodate for changes in illumination (e.g., due to new wall content being displayed). If a descriptor is in UNKNOWN_TRACKED state, it is matched with all descriptors in the database. If a sufficiently similar descriptor is found, both are merged and the descriptor state is set to KNOWN_TRACKED. Conversely, if a person with an UNKNOWN_TRACKED descriptor has been seen for a maximum number of frames (*time limit*), we add it to the database as a new person. Again, *descriptor similarity threshold* and *ReID time limit* were determined in an optimization step described in the following section.

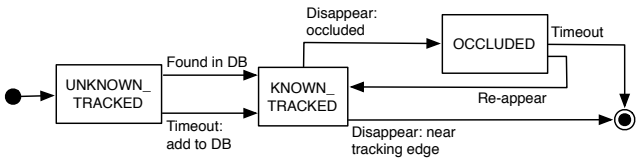

Figure 4: Lifetime of descriptors for currently active users in the ReID component. Users in the KNOWN_TRACKED and OCCLUDED states are also in the database.

Additional processing covers the common case of people becoming occluded by others and reappearing shortly after. While a KNOWN_TRACKED descriptor that disappears at the border of the tracked area is removed from the list of active descriptors, occlusion is assumed if a person disappears in the center. Corresponding descriptors are put into the OCCLUDED state. When checking for descriptor similarity, a distance- and time-based *occlusion_term* is added to the similarity for descriptors in this state, significantly increasing the chance that ReID is successful. The magnitude of *occlusion_term* is based on the number of frames *time* the descriptor has not been tracked and the distance *dist* between the occluded and the current descriptor:

$$occlusion\_term = \begin{cases} 0, & \text{if } time > time_{max}, \\ 0, & \text{if } dist > dist_{min} + \frac{time}{time_{max}} \cdot dist_{factor}, \\ \left(1 - \frac{time}{time_{max}}\right) \cdot val_{max}, & \text{otherwise.} \end{cases}$$

Using test videos, we heuristically determined good values for the constants to be $time_{max} = 60$, $val_{max} = 0.3$, $dist_{min} = 0.5$, and $dist_{factor} = 1.0$.

In a final step, the ReID component generates a low-resolution *User ID Bitmap* (in our prototype: 256x128 pixels) in screen coordinate space from the tracker's segmented bitmap and the tracked user IDs. This bitmap contains only those users close enough to the wall to touch it and is sent to the TouchProcessor. Additionally, skeleton data of all currently active users is sent to allow correlation of touches with occluded users and enable applications to react to user movements.

Feature	Weight
HEIGHT	-
FLOOR_NECK_DIST	0.15
FLOOR_HEAD_DIST	-
ARM_LENGTH	-
SHOULDER_WIDTH	0.10
HIP_WIDTH	-
FLOOR_HIP_DIST	0.00
TORSO_LEGS_RATIO	-
LEG_LENGTH	-
LEG_HISTOGRAM	0.30
TORSO_HISTOGRAM	0.40
HEAD_HISTOGRAM	0.05
FOOT_HISTOGRAM	0.00

Figure 5: Features used for re-identification with weights determined through optimization. Grey rectangles group features pertaining to the same human measurement. Features without weights were discarded due to high intra-person variance, features with weight 0 discarded during optimization.

4.1.2 Optimization

We optimized the feature weights (w_f above) using a training data set consisting of 36 subjects (9 female, ages 21-49), with two interaction sequences recorded per person. Each interaction sequence consisted of the person entering the tracking area, touching a series of 20 targets that successively appeared at random screen locations, and leaving the tracking area again.

For optimization, we built the database using the first set of 36 interaction sequences, then attempted re-identification using the second set. The optimizer then performed an exhaustive search on the solution space at 5% intervals, determining how many users fell into the categories *correct*, *incorrect*, and *not_matched* for each combination of weights. Additionally, we were interested in minimizing the number of unknown users incorrectly matched with a database entry. To this end, we removed one user from the database and probed with this user (again repeating for all users and feature weights). If a user was reported as found in the database, this was recorded as *false_positive*.

We determined optimal weights by minimizing the function

$$f_{min} = 0.5 \cdot not\_matched + incorrect + false\_positive$$

This prioritizes reported match failures over incorrect matches. Resulting feature weights are shown in Figure 5. For this combination of weights, the results were *correct* = 30, *incorrect* = 0, and *not_matched* = 4, while $false\_positive = 2$.

Fixing feature weights, we additionally evaluated sensitivity to varying descriptor similarity thresholds, the ReID time limit, and the number of histogram bins, with results shown in Figure 3. Several tradeoffs become apparent. First, a smaller similarity threshold results in more *correct* matches, but also increases the number of $false\_positive$ matches. An optimum can be found around 0.80. Increasing the ReID time limit also improves the number of correct matches. At the same time, the number of $false\_positive$ matches increases with a larger time limit. Additionally (and not visible in the graph), the time limit directly determines how fast new users are identified, prompting us to set it to 30 frames (= 1 second). Regarding histogram bins, there is a large good interval starting at around 30 bins. Our prototype uses 32 bins.

4.2 TouchProcessor

The TouchProcessor takes the User ID Bitmap and the skeleton data generated by the ReID Component and uses these to correlate touches on the wall to users. In a first step, the User ID bitmap is used. If the first step fails, skeleton data – including historical skeleton data of occluded people – is used, making the process more robust. Hand association is done using skeleton data as well. Note that the user may be in UNKNOWN_TRACKED state. In that case, the identification has failed. Figure 6 visualizes these processing steps.

To associate a touch with a user and hand, a number of operations are performed. The image-based mapping step begins by removing noise in the User ID Bitmap using a morphological closing operation. The touch position is then projected onto this bitmap and a small window around the touch (in the prototype: 5 pixels, corresponding to around 10cm) is searched for users to assign to the touch, with the closest user being selected if there is more than one user in the window (Figure 8a shows an example). If this does not succeed, the closest skeleton neck joint within 80cm is determined and its user associated with the touch (example in Figure 8b). In the event that this does not succeed either, the touch is left without an assigned user.

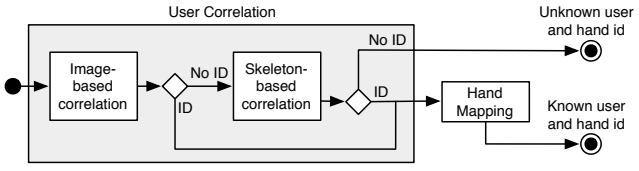

Figure 6: TouchProcessor steps for associating a touch with user and hand.

For touches with associated users, the TouchProcessor attempts to determine the touching hand. This algorithm uses a number of heuristics that depend on the number of hands tracked with high confidence by the RGBD camera. If both hands are tracked, we simply choose the one with the closest distance. If only one hand is tracked, and this hand is within 25cm of the touch, this is the touching hand. Otherwise, we assume that the untracked hand is touching. This covers the common case a user touching the wall in front of her body while the other hand rests at her side, e.g., visible in Figure 8a. As fallback, we use the body center as dividing line to determine the hand.

Finally, the TouchProcessor sends touch events enriched with User IDs to the application. The skeleton data received from the ReID component is forwarded to the application as well, thus allowing it to support body-centric interaction.

5. DEVELOPMENT SETUP, TOOLS, AND METHODOLOGY

We developed YouTouch! using a display wall with total dimensions of 5x2 meters and 24 megapixels resolution, consisting of twelve 55" touch-sensitive displays. The RGBD camera was placed at 2.4m height and a distance of 4.4m from the wall, maximizing viewable area while minimizing occlusions.

To allow for efficient iterative development, we put significant effort into a versatile test toolset. At the heart of the toolset is a recording and playback application (Figure 7) that is able to handle the full set of image and tracking data (RGB, depth, and segmentation images as well as skeleton data) in addition to touch data. The tool supports fast-forward playback and includes pause as well as single-step functionality. To avoid any issues with compression artifacts influencing the system, videos were encoded losslessly using the *huffyuv* codec. Complementing this playback tool, we implemented a debug view application that shows the output of the

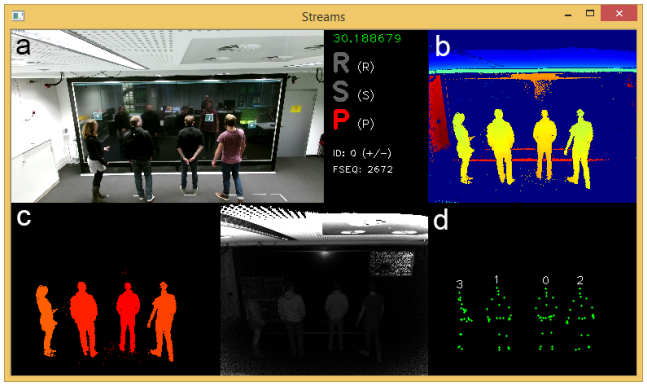

Figure 7: Screenshot of the Recorder and Playback application used for development and evaluation. We show a) RGB image, b) Depth image, c) User ID Bitmap, and d) Skeleton joints of users with associated IDs, and enable playback, pause and single-step of recorded tracking data.

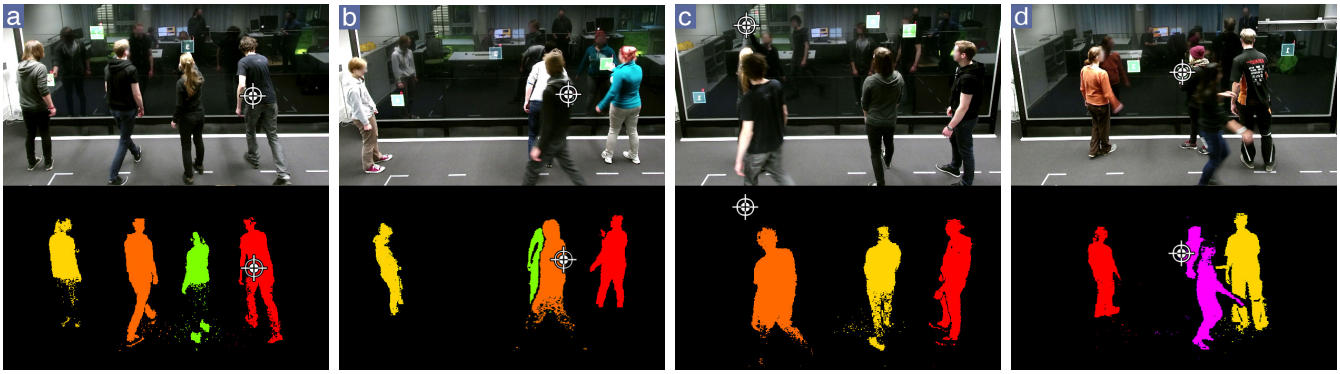

Figure 8: Typical scenario images (top: RGB and bottom: segmented, with touch positions marked by cross hairs in both images) showing successful a) image-based touch ID, b) skeleton-based touch ID and c) touch ID using occluded skeleton, as well as d) unsuccessful Kinect tracking (two people tracked as one) causing touch ID failure.

TouchProcessor, including User ID bitmap, skeletons with IDs, and touch event data (Figure 1b). Both the playback tool and the debug view application run without a connected camera or touch-sensitive wall. In combination, they allow swift reproduction and pinpointing of issues on development workstations; interesting situations can be replayed at will and the effects of algorithm changes judged quickly.

6. EVALUATION

Employing the development setup described in the previous section to record and evaluate videos of users, we measured the performance of the complete YouTouch! system – including ReID, touch association, occlusion handling, and hand determination. The main study goals were to determine the effectiveness of the user-touch association as a whole as well as the effectiveness of different system components. In addition, we wanted to estimate the potential for further improvements by analyzing the causes of the remaining identification errors.

6.1 Procedure

Users participated in two different group interaction scenarios. In each scenario, we successively presented 20 sets of touch targets to the users at random positions on the screen. Targets were user specific and marked with a User ID as well as the hand to use. After all targets of a set had been touched, a new set appeared.

The first scenario was designed to maximize user movement as well as short-term occlusions. Groups were composed of 4 users, with one touch target per user in each set. The second scenario additionally required participants to leave and re-enter the tracking area regularly to further stress the ReID component. In this scenario, groups were composed of 5 users each, with touch targets for 3 of them displayed in each set. Users without targets were asked to leave the tracking area and re-enter when a corresponding target appeared. A total of 45 users in 9 groups (15 female, ages 22-29) participated in the scenarios. Thus, our data set contains 9 groups · (20 sets · 4 targets + 20 sets · 3 targets) = 1260 touches.

Note that the evaluation videos and the videos used for ReID optimization were recorded using different users to prevent skewed results due to possible overfitting in the optimization step.

The scenarios were designed for maximum stress to the system. Since the positions of the targets were random, participants needed to exchange places often: We calculated a minimum of 528 (Scenario 1) and 450 (Scenario 2) position switches from the touch data, causing large amounts of occlusion. Difficult situations like the one in Figure 8b, where three users interact in very close proximity, are common. In comparison, Jakobsen et al. [16] found a total of 53 position switches on average in two-person wall interaction scenarios lasting 90 minutes.

6.2 Results

We used these data sets to evaluate the system's performance in several configurations and recorded the number of correctly and incorrectly as well as unmatched users for each of them. We evaluated the following configurations:

- No ReID: A baseline configuration with ReID turned off (i.e., a new ID assigned to each newly tracked user), using skeleton-based touch correlation.
- Raw ReID: A configuration that performed full ReID but used only basic (i.e., image-based) touch correlation. Occlusion tracking was turned off.
- Full System: All components enabled for maximum efficiency.

The results (Figure 9) show the major impact of the ReID algorithm. Although not statistically significant, we can see some additional improvements through the occlusion tracking and touch correlation heuristics. Considering the amount of movement and occlusions, the performance for Scenario 1 was very good and points towards usability in general application cases. Performance in Scenario 2 was still good, with some additional issues caused by participants constantly leaving and re-entering the tracking area.

Figure 8 shows several typical example frames from the evaluation that highlight the capabilities of the system. Frame a) shows

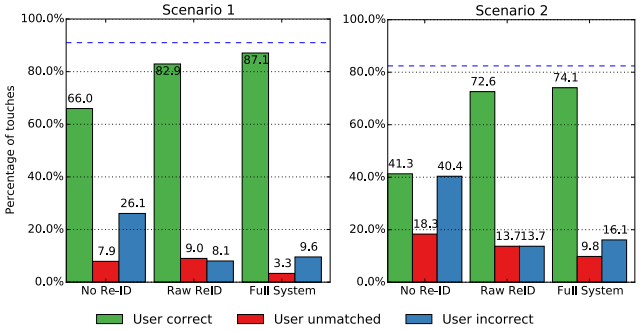

Figure 9: Touch User ID performance by algorithm. The dashed horizontal line shows the maximum performance possible without improving the RGBD camera's tracking system.

a simple case: Users are clearly segmented and there is no occlusion, so image-based touch correlation succeeds. In frame b), the touching user is partially occluded, causing image-based correlation to fail. Skeleton-based correlation, however, succeeds. Owing to occlusion, the touching user is completely untracked in frame c). However, there is a descriptor of the user in OCCLUDED state, and skeleton-based correlation using this descriptor succeeds. Finally, in frame d), Kinect segmentation has failed and is reporting two persons as one. In this case, touch correlation fails as well and reports the wrong user.

In addition, we were interested in the causes of the remaining errors. To this end, we stepped through the videos frame by frame in the vicinity of each error using our development system and manually categorized the failures according to the part of the system that failed. As can be seen in Table 2, more than 2/3 of errors were directly caused by the Kinect's tracking system. Many were related to touches that happened when the person was not tracked at all by the Kinect, with additional errors due to mis-segmentation of several people as one person and Kinect IDs moving from one person to another. The remaining major causes for errors were ReID failures (generally resulting in a new ID being assigned) and ReID that was still in progress (i.e., descriptor in UNKNOWN_TRACKED mode). The number of Kinect tracking failures also give us a maximum attainable performance when using the Kinect's tracking component, shown as a horizontal bar in Figure 9. Finally, they highlight the effectiveness of the secondary components of our system: In Scenario 1, half of the errors not caused by Kinect tracking failures are corrected going from the Raw ReID configuration to the Full System configuration.

7. DISCUSSION

The main result of the evaluation is very encouraging. Given an error rate of under 13% in a demanding data set, we believe that YouTouch! should be usable in a number of general application scenarios. Informal tests using our sample drawing application confirmed this: We had only few ReID and touch correlation failures. Human territorial behavior will probably prevent most tracking failures in serious contexts [16]. However, we assume that the system will not be accurate enough in cases where users move quickly in close proximity, such as in movement-based games. Since identification is largely based on clothing worn, it is also clear that we can not expect IDs to outlast clothing changes – and that it is not suitable for groups of users wearing the same clothing (e.g., uniforms). On the positive side, this should also alleviate any privacy concerns: Users are not identified permanently, nor are IDs unique enough to distinguish more than a few dozen users.

In crowded situations, YouTouch! will likely fail because of the large amount of occlusion (and in the current prototype, because the Kinect's tracking component is limited to six simultaneous users). In these cases other (e.g., marker-based) methods must be used as fallback. On the other hand, while we did not test this formally, YouTouch! should be somewhat resilient to lighting changes, since the person descriptor is based on a sliding average and will hence adapt automatically. Furthermore, our analysis of User ID errors shows that we are reaching the point of diminishing returns with the current Kinect-based tracker. We have not formally evaluated our hand detection method; this is left for future work.

While we are confident that the camera angle we chose allowed for efficient recognition, experiments with different camera angles should be easy. Generally (unless the user base or the camera angle is significantly different), it should not be necessary to repeat our optimization step. Finally, YouTouch! is easy to set up and deploy and only requires mounting the Kinect and following a simple calibration procedure to establish the position of the wall.

	Scenario 1	Scenario 2
Correct	87,1%	74,1%
Kinect tracker error	8,9%	17,6%
ReID error	2,5%	7,5%
TouchProcessor error	1,5%	0.8%

Table 2: Causes of remaining User ID errors. Note that the Kinect's tracking component is responsible for the majority of un- or misattributed touches.

8. CONCLUSION

We presented YouTouch!, a low-cost and reliable method that enables user-specific interaction at a large display wall. We track users by means of a commodity RGB + depth camera placed facing the wall. Person descriptors containing both color histogram data and anthropometric measurements allow re-identification of users after tracking has been lost, and specialized handling ensures high performance in the case of short-term occlusions. Touches are associated with people using both image and skeleton data, allowing even touches by users that are not currently tracked to be handled. We thoroughly optimized and evaluated the system using video and tracking data with a total of 81 subjects, showing good performance even in demanding conditions.

9. ACKNOWLEDGEMENTS

We would like to thank the study participants for their time and our colleagues at the Interactive Media Lab Dresden for valuable input. This article is supported by funding of the Excellence Initiative by the German Federal and State Governments (Institutional Strategy, measure "support the best").

10. REFERENCES

[1] M. Annett, T. Grossman, D. Wigdor, and G. Fitzmaurice. Medusa: A Proximity-aware Multi-touch Tabletop. In *Proc. UIST '11*. ACM, 337–346.

[2] T. Ballendat, N. Marquardt, and S. Greenberg. Proxemic interaction: designing for a proximity and orientation-aware environment. In *Proc. ITS '10*. ACM, 121–130.

[3] A. Bedagkar-Gala and S. Shah. 2014. A Survey of Approaches and Trends in Person Re-identification. *Image Vision Comput.* 32, 4 (April 2014), 270–286.

[4] B. Blazica, D. Vladušič, and D. Mladenić. 2013. MTi: A Method for User Identification for Multitouch Displays. *Int. J. Hum.-Comput. Stud.* 71, 6 (June 2013), 691–702.

[5] Y. Chen, Z. Liu, P. Chou, and Z. Zhang. VTouch: Vision-enhanced interaction for large touch displays. In *Proc. ICME '15*. 1–6.

[6] A. Clayphan, R. Martinez Maldonado, C. Ackad, and J. Kay. An Approach for Designing and Evaluating a Plug-in Vision-based Tabletop Touch Identification System. In *Proc. OzCHI '13*. ACM, 373–382.

[7] M. De Marsico, R. Distasi, S. Ricciardi, and D. Riccio. 2014. A comparison of approaches for person re-identification. In *Proc. ICPRAM (ICPRAM)*. 189–198.

[8] P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In *Proc. UIST '01*. ACM, 219–226.

[9] 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. Conference on Advances in Computer-Human Interaction '08*. 297–302.

[10] G. Doretto, T. Sebastian, P. Tu, and J. Rittscher. 2011. Appearance-based person reidentification in camera networks: problem overview and current approaches. *Journal of Ambient Intelligence and Humanized Computing* 2 (2011), 127–151.

[11] Y. Du, H. Ai, and S. Lao. Evaluation of color spaces for person re-identification. In *Proc. ICPR '12*. 1371–1374.

[12] S. Gong, M. Cristani, C. Loy, and T. Hospedales. 2014. The Re-identification Challenge. In *Person Re-Identification*. Springer London, 1–20.

[13] C. Harrison, M. Sato, and I. Poupyrev. Capacitive Fingerprinting: Exploring User Differentiation by Sensing Electrical Properties of the Human Body. In *Proc. UIST '12*. ACM, 537–544.

[14] C. Holz and P. Baudisch. Fiberio: A Touchscreen that Senses Fingerprints. In *Proc. UIST '13*. ACM, 41–50.

[15] C. Holz and M. Knaust. Biometric Touch Sensing: Seamlessly Augmenting Each Touch with Continuous Authentication. In *Proc. UIST '15*. ACM, 303–312.

[16] M. R. Jakobsen and K. Hornbæk. 2014. Up Close and Personal: Collaborative Work on a High-resolution Multitouch Wall Display. *ACM TOCHI* 21, 2 (Feb. 2014), 11:1–11:34.

[17] U. Kister, P. Reipschläger, F. Matulic, and R. Dachselt. 2015. BodyLenses: Embodied Magic Lenses and Personal Territories for Wall Displays. In *Proc. ITS '15*. ACM, 117–126.

[18] T. Leyvand, C. Meekhof, Y. Wei, J. Sun, and B. Guo. 2011. Kinect Identity: Technology and Experience. *Computer* 44, 4 (April 2011), 94–96.

[19] T. Meyer and D. Schmidt. IdWristbands: IR-based user identification on multi-touch surfaces. In *Proc. ITS '10*. ACM, 277–278.

[20] S. Murugappan, Vinayak, N. Elmqvist, and K. Ramani. Extended Multitouch: Recovering Touch Posture and Differentiating Users Using a Depth Camera. In *Proc. UIST '12*. ACM, 487–496.

[21] F. Pala, R. Satta, G. Fumera, and F. Roli. 2015. Multi-modal Person Re-Identification Using RGB-D Cameras. *IEEE transactions on circuits and systems for video technology* (2015), 788–799.

[22] R. Ramakers, D. Vanacken, K. Luyten, K. Coninx, and J. Schöning. Carpus: a non-intrusive user identification technique for interactive surfaces. In *Proc. UIST 2012*. ACM, 35–44.

[23] J. Rekimoto. Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proc. UIST '97 (UIST '97)*. ACM, New York, NY, USA, 31–39.

[24] S. Richter, C. Holz, and P. Baudisch. Bootstrapper: Recognizing Tabletop Users by Their Shoes. In *Proc. CHI '12*. ACM, 1249–1252.

[25] V. Roth, P. Schmidt, and B. Güldenring. The IR Ring: Authenticating Users' Touches on a Multi-touch Display. In *Proc. UIST '10*. ACM, 259–262.

[26] D. Schmidt, F. Chehimi, E. Rukzio, and H. Gellersen. PhoneTouch: A Technique for Direct Phone Interaction on Surfaces. In *Proc. UIST '10*. ACM, 13–16.

[27] D. Schmidt, M. Chong, and H. Gellersen. HandsDown: hand-contour-based user identification for interactive surfaces. In *Proc. NordiCHI '10*. ACM, 432–441.

[28] J. Shotton, R. Girshick, A. Fitzgibbon, T Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. 2012. Efficient Human Pose Estimation from Single Depth Images. *Trans. Pattern Analysis and Machine Intelligence* (2012), 2821–2840.

[29] Deller M. Eichner J. Ebert A. Thelen, S. Enhancing Large Display Interaction with User Tracking Data. In *Proc. CGVR '12*. 3–8.

[30] Nolte A. Ksoll M. Turnwald, M. 2012. Easy collaboration on interactive wall-size displays in a user distinction environment. In *Workshop Designing Collaborative Interactive Spaces for e-Creativity, e-Science and e-Learning*.

[31] R. Vezzani, D. Baltieri, and R. Cucchiara. 2013. People Reidentification in Surveillance and Forensics: A Survey. *ACM Comput. Surv.* 46, 2 (Dec. 2013), 29:1–29:37.

[32] U. v. Zadow, W. Büschel, R. Langner, and R. Dachselt. 2014. SleeD: Using a Sleeve Display to Interact with Touch-sensitive Display Walls. In *Proc. ITS '14*. ACM, New York, NY, USA, 129–138.