Hand Distinction for Multi-Touch Tabletop Interaction

Chi Tai Dang, Martin Straub, Elisabeth André
University of Augsburg
Institute of Computer Science
Multimedia Concepts and their Applications
Universitaetsstr. 6a, 86159 Augsburg, Germany
{dang, andre}@informatik.uni-augsburg.de, martin.k.straub@googlemail.com

ABSTRACT
Recent multi-touch multi-user tabletop systems offer rich touch contact properties to applications. Not only they provide touch positions, but also finger orientations. Applications can use these properties separated for each finger or derive information by combining the given touch contact data. In this paper, we present an approach to map fingers to their associated joined hand contributing to potential enhancements for gesture recognition and user interaction. For instance, a gesture can be composed of multiple fingers of one hand or different hands. Therefore, we present a simple heuristic for mapping fingers to hands that makes use of constraints applied to the touch position combined with the finger orientation. We tested our approach with collected diverse touch contact data and analyze the results.

Author Keywords
Input/Interaction, Multi-Touch, Tracking, Touch Properties, Tabletop Hardware.

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces–Input devices and strategies; I.4.9 [Image Processing and Computer Vision]: Applications

INTRODUCTION
Multi-touch is an increasingly emerging user interaction technology found in small display devices, such as the Apple iPhone, recent multi-touch notebooks as well as in larger form factors, such as the Microsoft Surface[1]. While the sensor capabilities of small display devices limit the kind of detectable touch properties, camera based systems as used for most interactive tabletop surfaces are more abundant and facilitate the detection of rich touch contact properties. These interactive tabletop surfaces have become widespread over the last years due to a cost-effective and reliable construction using Han's FTIR-approach [12] or another popular technology called diffused illumination as utilized in [19]. Along with the tabletop devices, developers create applications that make use of touch interaction. Examples include dual finger selection techniques to allow for greater precision at pointing tasks [4, 23, 3, 13], techniques to emulate graphical commands, such as deleting or moving groups of objects by using the complete hand [10, 22], techniques for fluid interaction, such as integrated control of rotation and translation [14], or techniques involving several fingers to emulate the functionality of computer mice [20, 17]. The easiest technique to recognize user interaction is to consider touch contacts without respect to their joined hand. However, on tabletop devices, user input can originate from different hands of one or several users and from multiple fingers of their hands.

Applications that are able to determine whether user input is provided by one or several hands may exploit the potential of a more natural and richer repertoire of input gestures. Often, a repertoire of gestures can only be extended at the expense of lower robustness, since new gestures distinguish from existing ones only by subtle variations. The inclusion of two-handed gestures would, however, not result into a significant loss of accuracy because information on handedness may be employed as a highly discriminative feature to classify input. In applications that emulate the functions of mouse buttons, techniques to distinguish one-handed from two-handed input could, for example, ensure that mouse actions are only triggered if touch points come from one hand and not from different hands that accidentally form a similar constellation.

Moscovich et. al. [18] present a study, which demonstrates that one-handed input and two-handed input is not interreplaceable, since handedness has an impact on the ease and accuracy with which gestures may be conducted. In particular, they showed that one-handed input is suitable for moving, stretching or rotating an object while two hands are more decent for tasks wherein separate control of points, such as selecting a region, is required. Their study advocates the use of both unimanual and bimanual gestures depending on the accuracy with which certain tasks can be performed. Here, distinguishing hands helps adapt the application behavior to offer possibilities to ease precise tasks in case of two hands, for example by changing the resolution or speed of object movements.

Even though many research projects exploit the potential of a

[1]http://www.surface.com

richer repertoire of input gestures resulting from two-handed input, current work usually does not consider cases where a collection of touch points should be interpreted differently depending on whether they originate from the finger of one hand or the fingers of several hands. For example, Benko and colleagues [4] present a dual finger stretch technique where one finger is used to select an area of the interface and a second one is used to scale the area. However, when evaluating their gestures, they start from the assumption that all gestures are executed with two fingers of two different hands belonging to a particular person. Such a gesture could also be performed by two fingers of the same hand as depicted to the left in figure 1. As a consequence, it might be difficult to

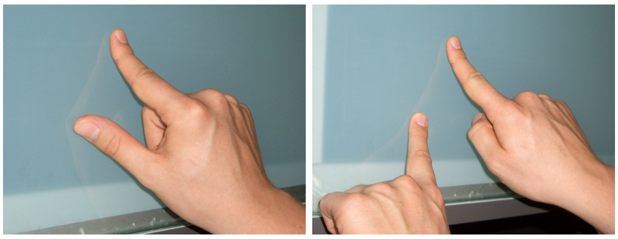

Figure 1. Gestures with one hand (left) and two hands (right) creating the same touch positions and tracks.

distinguish a dual finger stretch gesture performed by fingers of the same hand from a dual hand duplicate gesture where a copy of a visual object is produced by pulling the object apart with the fingers of two separate hands. Another example pertains to the application behavior for multiple objects when two objects are moved together. A one-handed gesture could result in merging those objects whereas a two-handed gesture would overlay one object with the other for a precise comparison task.

Peltonen and colleagues [19] report on problems that arise when multiple users interact in parallel on large surfaces and unintentionally break territorial boundaries. Techniques that distinguish between one-handed and multi-handed input may help resolve such conflicts. Consider, for example, the case where a photo is accidentally zoomed because two users are trying to move it toward themselves as depicted in figure 2. If a system is, however, able to recognize that two users

Figure 2. Two different users executing a pulling away gesture.

are manipulating one and the same object in parallel, but with different intentions, it might respond to the users' behavior in a more appropriate manner. For example, it might lock the object until the users have agreed upon what to do with the object or create a duplicate of the object.

Terrenghi and colleagues [21] conducted a study to investigate whether people manipulate physical objects on a tabletop differently than digital objects. They found that users tended to use just one hand when interacting with digital objects. Distinguishing hands and offering interaction based on handedness might encourage users to use both hands. For example, participants were requested to spatially structure objects. In the physical tasks, users could move multiple pieces altogether to form a structure, but it was hard to imitate such an action in the digital tasks. However, a hand distinction could enable an adaption of the application behavior, so as to select all underlying and overlying objects for movement if both hands are touching an object, whereas one hand selects only the directly touched object.

In this paper, we contribute to potential enhancements of gesture recognition by explicitly distinguishing one-handed from two-handed gestures. Tabletop tracking components such as used in the Microsoft Surface provide a lot of touch properties, but almost no hand related information to applications. Therefore, we present an algorithm that maps touch contacts onto hands with a high level of precision and discuss observations from an empirical evaluation.

RELATED WORK

Various attempts have been made to distinguish one-handed touches from two-handed touches. The DiamondTouch [5] table makes use of modulated electric fields. When a user touches the surface of the table, the contact areas are capacitively coupled through the user to a receiver corresponding to that user. In this way, the DiamondTouch table is able to determine for each touch point to which user it belongs. However, DiamondTouch table is not able to distinguish between one- and two-handed gestures from one and the same user.

Echtler and colleagues [8] present an approach for FTIR-tables that features the possibility to map finger touches to the hands of a single or several users. It is based on a top-mounted infrared light source that let the hands and arms throw shadows onto the surface. However, they haven't explored their setup to explicitly distinguish hands yet. Another approach was employed by Dohse et al. [7] who enhanced the interaction on an FTIR-table with a camera placed above the interactive surface. This technology allowed them to track hands by means of computer vision techniques, thus assign a specific touch point to a user or his hand.

Malik and colleagues [15] and Do-Lenh et al. [6] present various one-handed and two-handed input techniques for a visual touch pad and an augmented tabletop. Both works distinguish single gestures based on positional information for the fingertips. Furthermore Malik [15] consider finger orientations. In their case, there is no need to disambiguate gestures based on handedness because the chosen gestures are sufficiently different. Nevertheless, their approaches to determine finger-features can also be used to distinguish between one-handed and two-handed input on infrared images captured by tabletop setups.

The idea behind our approach is to exploit information on the anatomy of human hands and fingers to distinguish between one-handed and two-handed gestures. We will show that positional information for the fingertips and finger orientations suffices to reliably map fingers onto hands even if not recognized with perfect accuracy.

FINGER POSITION AND ORIENTATION

Figure 3 sketches the position and orientation of a detected finger blob, wherein a finger contact area is approximated through an ellipse with a position (P_x, P_y) and major / minor axis distance. The major axis vector of the ellipse allows us to calculate the angle of x-axis and thus the orientation of the detected finger blob. Figure 3 assumes that position $(0, 0)$ is the top-left-corner and the y-coordinates increase downwards as commonly used with most operating systems.

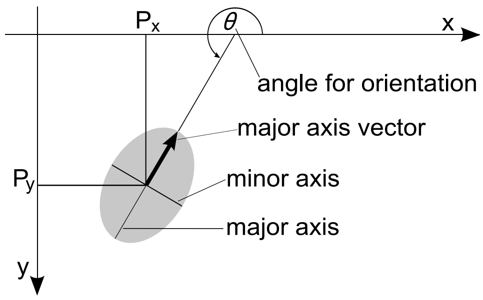

Figure 3. Finger blob represented as an ellipse with position and angle.

The determination of reliable finger orientations and finger positions is a fundamental requirement for mapping fingers to their joined hand. The finger position or rather the center of the detected blob is determined as the center of the approximated blob ellipse or the calculated mass center of the blob. While the finger position can be detected by any sensor techniques without problems, the finger orientation cannot always be detected.

Tabletop or large display interfaces mostly employ infrared camera based tracking systems and thus provide rich options for obtaining finger properties by leveraging computer vision approaches. Although there are several multi-touch techniques to date as described in [1], we consider only the FTIR and DI technique here, since these two techniques are widely used. Images captured via these techniques are most commonly passed through blob detection algorithms, such as used in [2, 9], in order to determine the finger contact area. After some more object recognition tasks, the finger properties as outlined in figure 3 are available to applications or for further analysis. The captured image on the left hand side of figure 4 stems from a DI-table. What makes such images ideally suited for detecting the finger orientation is the fact that the user's hand reflects infrared light that gets captured by the camera. Depending on the lighting situation and the particular DI-setup, only the fingers are visible or even the whole hand. This enables us to detect the finger orientation reliably by analyzing the proximity of each finger. A simple method, therefore, is to find a bright pixel along the four axis directions of the detected finger blob ellipse and choose the longest distance. The captured image on the right hand side of figure 4 is typical of FTIR-tables. What makes such images characteristic is that the finger blobs are very bright and thus can be detected easier than those occurring in DI-images. In contrast to the DI-method, the hand belonging to a finger cannot be recognized reliably with the captured image only. However, there are approaches to track hands with FTIR-tables by means of auxiliary enhancements as discussed before [8, 7]. Overall, cases where identifying finger orientation is erroneous rarely occur. For instance, if a finger approaches the surface from a steep or even perpendicular angle [11, 16], the near proximity of the contact area often does not consist of sufficient finger or hand pixels. Another situation appears when finger contacts of different hands are placed too close together. In this case, the finger contacts are merged to one contact area and detection is not reliable, since there is more than one finger leading away.

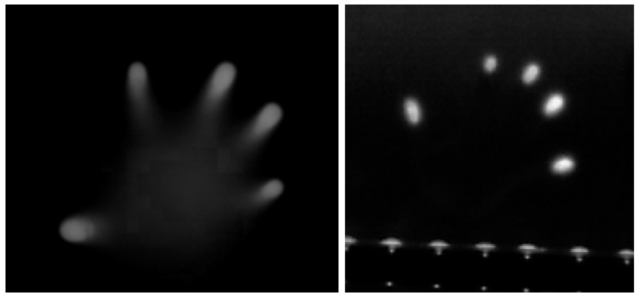

Figure 4. Image captured by a camera from the DI-table based table Microsoft Surface (left) and an FTIR-table (right).

MAPPING FINGER CONCEPT

Our objective is to map a set of finger contacts to a set of hands, wherein each hand is given a unique hand id. The finger contacts are assigned to a hand id based on its associated joins to a hand. To achieve this task, we take advantage of the fact that the constellation of touch contacts of one hand to touch a surface is limited naturally. By means of a heuristical parametrized approach, we are able to classify each finger. We identified two features that are sufficient to achieve correct assignments for most of the cases. For these features, constraints are formulated, which enable us to reliably distinguish fingers of different hands.

Such a mapping can also be performed through computer vision methods by tracking components, for example in case of the DI-method when the whole hand reflects infrared light. Our approach, however, enables applications to create a hand mapping for a set or a subset of finger contacts, thus calculate this information on demand or in case the tracking component does not provide a hand mapping. For example, applications can determine the hand ids for the touch contacts on a particular object only and ignore the remaining touch contacts.

Analysis of Finger Arrangements

When considering possible finger arrangements of human hands, we only take arrangements for touching a surface into account. Figure 5 exemplarily depicts the layout of the thumb and all fingers of a right hand with its oriented lines represented as dashed lines. While looking at two contact areas only, we can identify for all combinations the distance between their center position, their oriented lines and the interior angle between them. Any other finger contact composition of one hand can be looked upon as a subset of the depicted contacts. The relative orientation to each other varies just a little due to the anatomy of human hand skeleton. That is normally for two fingers abreast, the distance between their contact positions can reach a maximum value. The thumb represents an exception because in combination with a finger, it enables a higher distance and interior angle between their oriented lines as sketched in figure 6. Upon these observations, we deduce constraints based on geometrical properties.

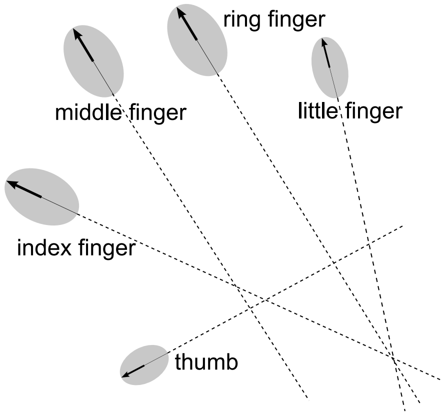

Figure 5. The five fingers of a right hand with the intersection points of its backward oriented lines.

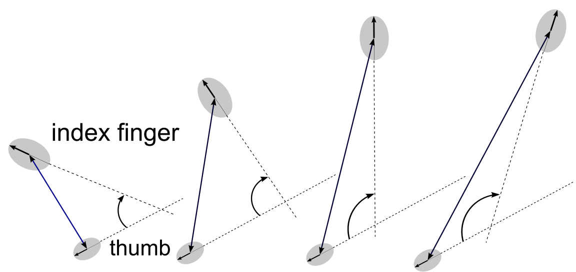

Figure 6. Variation in distance and angle of thumb to a finger. Combinations are from left with low distance and acute angle to right with far distance and obtuse angle.

Constraints and Conditions

The task of mapping a set of fingers to a set of hands can be decomposed to compare only two finger contact areas for each contact area with each other in the set. Figure 7 delineates such a comparison and the geometrical data that can be calculated. Typically, the center positions $P1(x, y), P2(x, y)$ and the orientations as angle of x-axis γ_1, γ_2 are given, which enables us to calculate the distance d between the center positions, the interior angle Θ between the oriented lines, the intersection point $I(x, y)$ of the oriented lines and the distance n of point $I(x, y)$ to line d by means of the normal line. Furthermore, we have to calculate a pair of gradient vectors v_1, v_2 using the given orientation angles, which are definite as compared to the ambiguity of slope values.

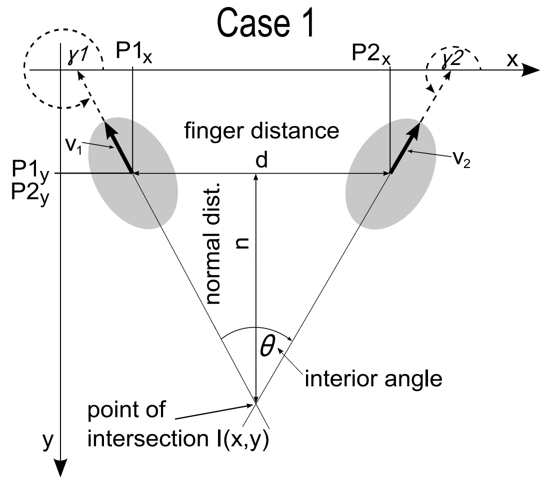

Figure 7. Oriented lines of two fingers with intersection point in back, the distance between two fingers and the normal distance to the intersection point.

Condition 1: Distance and Proximity

The distance between two touch contacts is treated as a first indicator for being from the same hand or not. For this purpose we empirically determined a maximal distance $D_{max} = 10,55$" for the farthest distance a thumb can have to the other fingers. If the distance d exceeds D_{max}, then the considered two touch contacts are from different hands. We also empirically determined a distance $D_{adj} = 3,5$" for adjacent fingers whose touch contacts are less than D_{adj} distant from each other as can be seen in figure 5.

Condition 2.1: Intersection Point

The next step is to consider the intersection point $I(x, y)$, which has to be behind of both touch contacts. This condition can be checked by means of the intersection point together with either both center points $P1(x, y), P2(x, y)$ or their gradient vectors v_1, v_2. A special case appears for in parallel-oriented contact areas, which have no intersection point. This case has to be caught by means of the gradient vectors and will be handled with condition 2.2. Figure 8 illustrates a situation with case 2, where the intersection point is in front of both touch contacts, thus violates the intersection point condition twice. This arrangement cannot be produced by fingers of one hand, whereas the thumb in combination with another finger of the same hand is able to create such an arrangement. To handle this case, we employ temporal information, that is hand ids previously assigned to touch contacts. For instance, when the thumb creates contact areas as in case 2, it performs a grab gesture on the surface that usually starts with correctly assignable touch contacts. Constellations as in case 2 are then fused with the

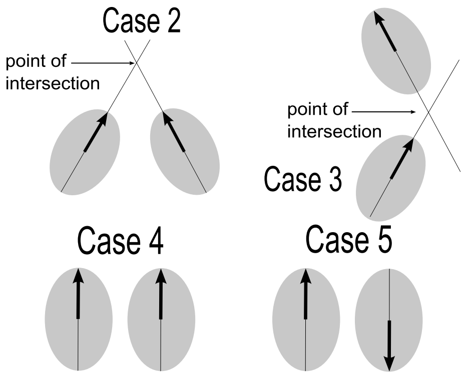

Figure 8. Intersection point in front of at least one line (1,2), Quasi parallel-oriented finger (4) and quasi-reverse oriented finger (5).

previously assigned touch contacts and hence get assigned correctly. Case 3 in figure 8 also violates the condition because the intersection point is in front of one touch contact. Such an arrangement can be produced, for example, through the pump gesture as proposed in [20]. The pump gesture is performed by describing an arc movement with the thumb under another finger. As for case 2, this case can be handled by considering temporal information.

Condition 2.2: Adjacency and Parallels

Adjacent touch contacts can be arranged in parallel or quasi parallel as sketched in case 4 of figure 8. In these cases the interior angle between their oriented lines is very small and enables us to stabilize adjacent contacts. Therefore, we define an angle Θ_{adj} and assume that two touch contacts belong to two adjacent fingers if their distance d is less than D_{adj} (distance condition) and the interior angle Θ is less than Θ_{adj}. We have found $45°$ to be a good value for Θ_{adj}. Not directly adjacent fingers, for instance the index finger and the ring finger, can also be placed in parallel. For those cases, the assignment depends on the value of D_{adj}, which is set to include a proximity of 3 fingers. The touch contacts of case 5, however, cannot be produced by only one hand, when using normal touch interaction. Hence case 5 gets excluded by examining the gradient vectors.

Condition 3: Distance Relationship

Condition 1 and 2.2 are sufficient to recognize adjacent finger touches. However, condition 1 and 2.1 are not sufficient because case 1 in figure 7 for distances $D_{adj} < d < D_{max}$ can still be formed by fingers of two different hands. Therefore, we adopt the limitations imposed by the limbs and joins of human hands to deal with that situation. Figure 9 depicts the distance between two touch contacts and the normal distance to their intersection point. As becomes apparent, the farther two touch contacts are, the higher the interior angle between their oriented lines is and the shorter the corresponding normal distance becomes, for example $d_1 < d_2, n_1 > n_2$ in figure 9. Condition 3 exploits this relationship to constrain the constellation and distances of touch contacts, thus exclude or recognize touch contacts from different hands. Therefore, we empirically determined a value $N_{max} = 14,06$" for the maximum normal distance and allow only normal distances n that are lower than N_{max}. Normal distances higher than N_{max} are handled by condition 2.2 or indicate touch contacts from different hands.

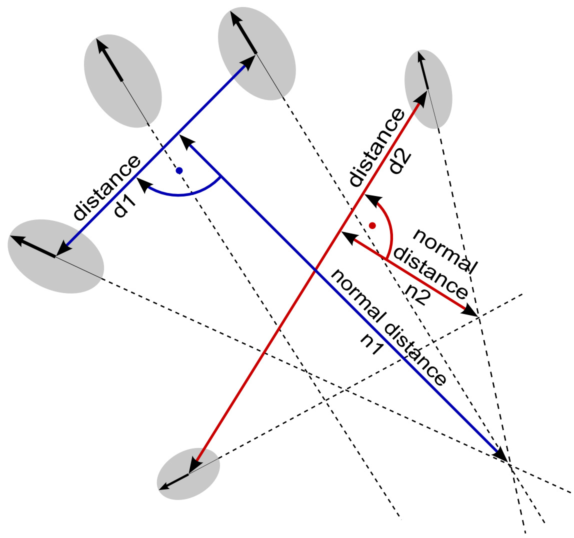

Figure 9. Two examples for the relationship between the distance of two fingers and the normal distance to the intersection point of its backward oriented lines.

$$score = (D_{max} - d) - \frac{n * D_{max}}{N_{max}} \quad (1)$$

In equation 1, we scale the normal distance n to the proportion of the maximum distances D_{max} with N_{max}. Since the distance d is in inverse proportion to n, we subtract the result of the previous adaptation from the difference between D_{max} and d to get the $score$. If the $score$ is positive, then the touch contacts are from the same hand. Otherwise the $score$ is negative and the intersection point is too far away. One more point is that figure 9 shows touch contacts of one hand with the fingers outstretched, but gestures are composed of constellations in which fingers are placed close together as well. Our approach handles such constellations through the distance and adjacency condition.

Temporal Adaptation

Tracking systems usually assign an incremental finger id to each new touch contact of a finger and keep track of the touch contact while retaining the finger id for the same finger. By examining frames over time using the finger id, we are able to capture and carry along dynamic hand mapping information. For a new frame, the cached hand ids are taken over from the previous frame. After that, the conditions are applied to the new frame, so as to correct the adopted mapping or handle new touch contacts. To cope with situations where fingers of different hands are close to each other or entangled, our implementation maintains the number of fingers for each hand id from the previous frame. When two hands move close together, then their finger ids should have already been mapped to hand ids in a previous frame. Therefore, we considered the number of fingers when deciding whether to change a hand id or not. The temporal adaptation is an important component of our approach and improves classification enormously.

Diversity of Finger Anatomy

The dimensions of the hands and the extent of the fingers vary between humans and their age but the geometrical constraints as described with condition 3 and 2.1 still remain valid. Depending on the size of users' hands, the maximum distance D_{max} might be too small and has to be increased. On the contrary, the maximum distance for adjacent fingers D_{adj} could be too big for children and thus require an adaptation. We have not tested our approach for children, however a method to adjust the parameters dynamically is to take the physical mass of the considered finger touches into account.

EMPIRICAL EVALUATION

Apparatus

For collecting finger contact data and evaluating our approach, we utilized a Microsoft Surface Developer device. It is a horizontal direct-touch tabletop that employs the DI-method and provides reliable finger orientation. The tabletop device rear-projects images onto a surface measuring 24" x 18" with a resolution of 1024 x 768 pixels.

Experiment

The experiment aimed at collecting touch contact data of diverse hand dimensions in order to verify our finger mapping concept and provide quantitative results. We even went one step further and tested the performance of our concept for conflict situations where fingers of two hands crossed each other or were even beneath other hands.

Participants and Tools

We recruited 10 volunteers for the experiment, 4 males and 6 females aged from 21 to 60. One of them was left-handed, and all of them, but two had already experience with multi-touch interaction through the Apple iPhone. We implemented a tool to record all finger touch contacts with their contact properties and timestamps, that is to spy and log them into an xml-file. We also wrote a tool called contact-player to step through each recorded frame and to visualize the contact data in the manner of a movie showing the flow of the touch contacts. The contact-player implemented a prototype of our approach to map fingers to hands and offered the possibility to annotate fingers in case they were falsely mapped. After manually verifying all recorded frames, the contact-player calculated statistics about the amount of fingers and frames and correctly or falsely mapped fingers, so as to provide recognition rates. For these statistics, only frames with two or more touch contacts were considered.

Methodology

The experiment used a within-subject design and consisted of five tasks, each deliberately designed to verify the approach step by step. The first task validated the geometry assumptions for only one hand on the surface and for different hand dimensions. The second task did the same for two hands beneath on the surface. Task 3 addressed the geometry assumptions when the involved fingers exploited their scope to vary angle and distance. Tasks 4 and 5 served to simulate conflict situations with finger arrangements that are often used for zoom or grab gestures. All participants were told to approach the surface with their fingers as if they would manipulate objects while performing the following tasks:

Participant	**Geometry** (1, 2)	**Variance** (3)	**Conflicts** (4, 5)
Average overall	97.5% $\sigma = 0.48$	96.82% $\sigma = 0.79$	90.55% $\sigma = 1.05$
Tallest	98.1%	98.5%	92.3%
Smallest	95.0%	92.0%	90.1%
Best	99.5%	99.6%	95.4%
Worst	94.9%	92.0%	84.2%

Table 1. Recognition rate for all participants, the tallest, the smallest, the best and the worst.

1. They touched the surface with one hand and then with the other hand. For each hand, they used 25 previously defined finger combinations. The combinations consisted not only of the fingers, but also the thumb and were chosen to cover up arrangements that are frequently used to resize, move or grab objects.
2. They performed the same combinations as in (1) but simultaneously with both hands.
3. They performed the same combinations as in (1) and shrinked up and stretched the involved fingers once to their possible extent.
4. They placed the following combinations with two hands apart from each other on the surface, moved the fingers close together, moved the fingers to cross the fingers of the other hand and then moved the fingers beneath the other hand for each hand. The combinations were:
   (a) Both index finger.
   (b) Both index finger and thumb.
   (c) One hand with index finger and the other hand combined with thumb.
   (d) The same as in (4.c.) with thumb and middle finger.
   (e) All fingers with thumb for both hands.
5. They performed the same task as in (4) while the second hand stemmed from a different user at the left and afterwards at the right side.

Results

Table 1 shows the recognition rates for all participants with mean deviation, for the tallest (height 1.98 m), the smallest (height 1.52 m) participant and recognition rates for the best and worst. We annotated 84001 frames with 335561 touch contacts, that is an average of 3.99 contacts per frame.

Geometry Assumptions

Although most of the participants had no experience with multi-touch interaction, they touched the surface in such a way that the finger orientation could be recognized correctly.

For that reason the recognition rate was over 92% in the worst case for the geometry (1, 2) and distance variance (3) tasks. The average recognition rate of about 96%/97% confirms our geometry assumptions and the efficacy of the formulated constraints. For these tasks, false classification occurred mostly due to arrangements with the thumb in combination with ring finger or little finger, where the angle between the fingers was around 180°. That is when the fingers were oriented quasi antiparallel as illustrated in figure 10. This observation suggests to extend the approach for handling the thumb separately, since it can adopt constellations with fingers that the fingers alone cannot form. However, finger combinations with three and more touch contacts could be recognized with a high accuracy. This is due to the stabilization effect of adjacent fingers of one hand, where our adjacency distance includes up to three fingers. Also worth mentioning is that the recognition rates should be considered under the circumstance that finger orientation was not detected with perfect accuracy through the tracking component. Certainly this could be improved, but overall the detection was accurate enough to handle the wrongly detected finger orientations as little noise within the results.

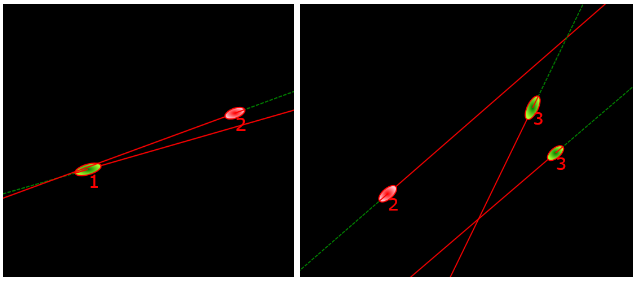

Figure 10. False classification with thumb and ring-finger / little-finger. An ellipse depicts a touch contact, the lines depicts the orientation and the numbers denotes the recognized hand ids.

Conflict Situations
Tasks 4 and 5 lasted between 30 seconds and 60 seconds each and the participants moved their fingers into a conflict situation at least twice following the instructed combinations, for example as illustrated in figure 11. As expected, the recognition rate for conflict situations was worse than for tasks 1-3 but were still better than 84% in the worst case and about 95% in the best case. Here, the recognition benefits greatly from the temporal adaptation and there is still potential for improvements. If one hand moves beneath the other hand, then the mapping creates two clusters of fingers, which move on the 2D-surface. We could observe that the clusters sometimes merge to one big cluster, hence counting to a lot of falsely classified fingers. To cope with that issue, strategies could be applied that take the clusters over multiple frames into account. For example, two clusters can be assigned a confidence value for being distinct hands by virtue of the euclid distance between their cluster-centroids as sketched in figure 12. Once distinct clusters have been identified, these confidence values adhered to particular finger ids then influence the decision whether or not to correct an assigned hand id. Another cause for falsely classified fingers was the recognition of finger orientation because users were beware of getting in contact with the fingers of other users, they sometimes moved their fingers to touch the surface under a perpendicular angle, which in turn led to wrong recognized finger orientations. In all, conflict cases like these are rather rare and are mostly unintentional, therefore the accuracy for these cases is not decisive for real world multi-user interaction.

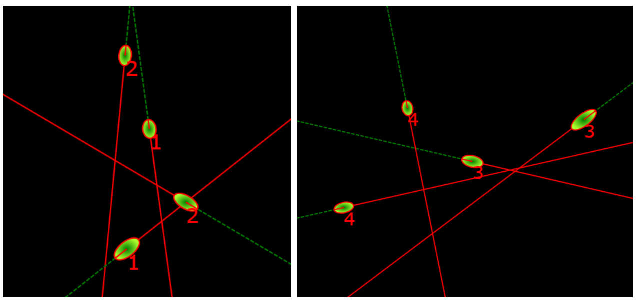

Figure 11. Conflict cases where one hand is beneath the other hand.

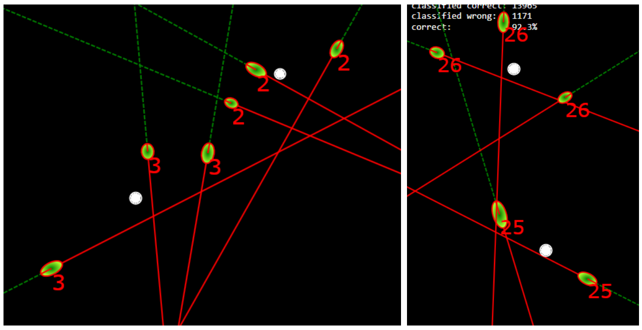

Figure 12. Each image consists of two hands with several fingers and their centroids. Centroids provide information to mark the fingers of two different hands.

CONCLUSIONS AND FURTHER WORK

We presented an approach to distinguish hands using only finger position in combination with finger orientation. Even though without perfect accuracy of the hand mapping approach, the results show that the presented concept suffices to distinguish hands for two-handed input on interactive tabletops. False recognition happened mostly because of the thumb being more flexible as for the constellation with fingers. Therefore, enhancing the approach with respect to the thumb would improve the approach. We also found that the temporal adaptation of our prototype was an important part and still bears a lot of potential for improvement, for example through keeping statistics of more than one frame. Another approach is to identify fingers as being from different hands by means of the centroid of the identified hands. The conducted experiment mainly focused on verifying the geometry assumptions and distinguishing hands from one and the same user. Further studies will address the recognition rates when multiple users interact in parallel within a collaborative environment. Of interest are also the possibilities to determine different users based on hand mappings.

REFERENCES

1. Nui group community, getting started with multitouch - multitouch techniques. http://nuigroup.com/forums/viewthread/1982.

2. Touchlib, a multi-touch development kit. http://www.nuigroup.com/touchlib.

3. P.-A. Albinsson and S. Zhai. High precision touch screen interaction. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 105–112. ACM, 2003.

4. H. Benko, A. D. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272. ACM, 2006.

5. P. Dietz and D. Leigh. Diamondtouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226. ACM, 2001.

6. S. Do-Lenh, F. Kaplan, A. Sharma, and P. Dillenbourg. Multi-finger interactions with papers on augmented tabletops. In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 267–274. ACM, 2009.

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 *ACHI '08: Proceedings of the First International Conference on Advances in Computer-Human Interaction*, pages 297–302. IEEE Computer Society, 2008.

8. F. Echtler, M. Huber, and G. Klinker. Shadow tracking on multi-touch tables. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 388–391. ACM, 2008.

9. F. Echtler and G. Klinker. A multitouch software architecture. In *NordiCHI '08: Proceedings of the 5th Nordic conference on Human-computer interaction*, pages 463–466. ACM, 2008.

10. J. Epps, S. Lichman, and M. Wu. A study of hand shape use in tabletop gesture interaction. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 748–753. ACM, 2006.

11. C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan. Direct-touch vs. mouse input for tabletop displays. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 647–656. ACM, 2007.

12. J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM, 2005.

13. W. A. König, J. Gerken, S. Dierdorf, and H. Reiterer. Adaptive pointing: implicit gain adaptation for absolute pointing devices. In *CHI EA '09: Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, pages 4171–4176. ACM, 2009.

14. R. Kruger, S. Carpendale, S. D. Scott, and A. Tang. Fluid integration of rotation and translation. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 601–610. ACM, 2005.

15. S. Malik and J. Laszlo. Visual touchpad: a two-handed gestural input device. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 289–296. ACM, 2004.

16. E. Mansor, A. De Angeli, and O. De Bruijn. Little fingers on the tabletop: A usability evaluation in the kindergarten. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 93–96, 2008.

17. J. Matejka, T. Grossman, J. Lo, and G. Fitzmaurice. The design and evaluation of multi-finger mouse emulation techniques. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1073–1082. ACM, 2009.

18. T. Moscovich and J. F. Hughes. Indirect mappings of multi-touch input using one and two hands. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1275–1284. ACM, 2008.

19. P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko. It's mine, don't touch!: interactions at a large multi-touch display in a city centre. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1285–1294. ACM, 2008.

20. C. Sousa and M. Matsumoto. Study on fluent interaction with multi-touch in traditional gui environments. In *TENCON 2007*, pages 1–4, 2007.

21. L. Terrenghi, D. Kirk, A. Sellen, and S. Izadi. Affordances for manipulation of physical versus digital media on interactive surfaces. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1157–1166. ACM, 2007.

22. E. Tse, C. Shen, S. Greenberg, and C. Forlines. Enabling interaction with single user applications through speech and gestures on a multi-user tabletop. In *AVI '06: Proceedings of the working conference on Advanced visual interfaces*, pages 336–343. ACM, 2006.

23. D. Vogel and P. Baudisch. Shift: A technique for operating pen-based interfaces using touch. In *CHI '07: Proceedings of the conference on Human factors in computing systems*, pages 657–666. ACM, 2007.