

RESEARCH

Open Access



# Design and implement chords and personal windows for multi-user collaboration on a large multi-touch vertical display

Ioannis Leftheriotis<sup>1,2\*</sup>, Konstantinos Chorianopoulos<sup>1</sup> and Letizia Jaccheri<sup>2</sup>

\*Correspondence:

iolef@acm.org

<sup>1</sup> Department of Informatics,  
Ionian University, Tsirigoti  
Square 7, Corfu, Greece  
Full list of author information  
is available at the end of the  
article

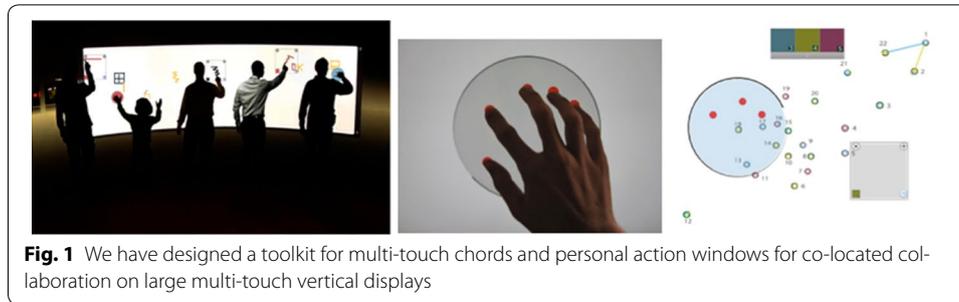
## Abstract

Co-located collaboration on large vertical screens has become technically feasible, but users are faced with increased effort, or have to wear intrusive personal identifiers. Previous research on co-located collaboration has assumed that all users perform exactly the same task (e.g., moving and resizing photos), or that they negotiate individual actions in turns. However, there is limited user interface software that supports simultaneous performance of individual actions during shared tasks (Fig. 1a). As a remedy, we have introduced multi-touch chords (Fig. 1b) and personal action windows (Fig. 1c) for co-located collaboration on a large multi-touch vertical display. Instead of selecting an item in a fixed menu by reaching for it, users work simultaneously on shared tasks by means of personal action windows, which are triggered by multi-touch chords performed anywhere on the display. In order to evaluate the proposed technique with users, we introduced an experimental task, which stands for the group dynamics that emerge during shared tasks on a large display. A grounded theory analysis of users' behaviour provided insights into established co-located collaboration topics, such as conflict resolution strategies and space negotiation. The main contribution of this work is the design and implementation of a novel seamless identification and interaction technique that supports diverse multi-touch interactions by multiple users: multi-touch chord interaction along with personal action windows.

**Keywords:** Chords, Multi-touch, Collaboration, Personal windows, Multi-user, Large screen

## Background

In recent years, there have been great advances in the accuracy and the available number of touches supported on large-scale multi-touch (MT) hardware technology (e.g., FTIR, laser-plane, DI, other combined installations). Although this allowed multiple users to interact with a relatively low-cost screen simultaneously, there is still limited user interface software technology support for group collaboration. For example, many MT systems (e.g., moving and resizing photos) assume that co-located users perform the exact same type of interaction on the screen, but there are applications (such as drawing on a shared canvas with different pens or working on maps) that require concurrent



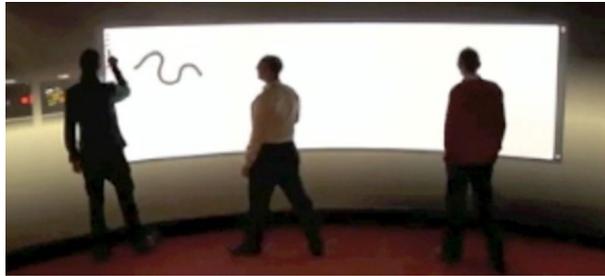
activity of diverse interactions. As a result, there is a need for user interfaces that support concurrent individual actions on a multi-touch screen without the need of special equipment.

A table/wall setting provides a large interactive visual surface for groups to interact together. It encourages collaboration and coordination, as well as decision making and problem solving among multiple users and therefore needs new kinds of interface [1]. Since most applications are developed for desktop computers/mobile devices and for single-user interaction, new interaction techniques that support seamless collaboration on larger MT screens are needed. Most conventional metaphors and underlying interface infrastructures for single-user desktop systems have been traditionally geared towards single mouse and keyboard-based WIMP interface design and might not be suitable for large MT screens. For example, Nacenta et al. [2] carried out an exploratory study to determine how several types of established interaction techniques (such as drag-and-drop, radar views etc.) affect coordination and awareness in tabletop tasks and proved that the choice of interaction technique does indeed matter, affecting coordination, performance and preference measures. Elliott and Hearst [3] proposed a touch-sensitive interface should be used as a more appropriate interaction technique for larger interaction surfaces.

Notably, the Reality MT screen at the University of Groningen<sup>1</sup> offers an impressive large screen, but there is no support for user awareness or appropriate menu selection technique. Using conventional pulldown or popup menus might require walking across the room to the appropriate button [4]. In the following figures, we demonstrate one such problem. In order for the user in the middle to change the colour of his drawing pen from black to red (Fig. 2), he has to literally walk toward the left part of the screen, where the appropriate menu can be found. When the user arrives at the menu, the user on the right has already started painting with the red colour (Fig. 3). User in the middle has again to walk back to his original position in order to draw his blue line. It is worth observing that the user on the right has been enforced to continue the rest of his painting with the new pen colour, which might have not been his intention (Fig. 4).

A flexible and scalable solution to the above practical issue has been one of the main motivations of our research. We considered that a chorded—simultaneous touch of more than one finger—input technique might be the solution. We design and develop an innovative method that makes use of chorded interactions and personal windows. Users

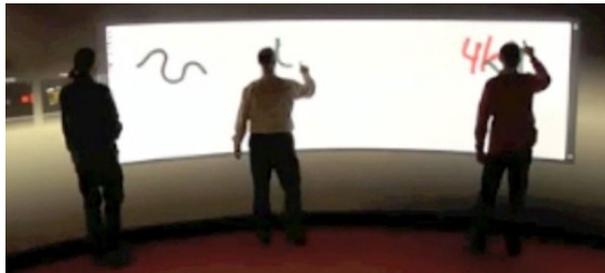
<sup>1</sup> YouTube demo video: <http://www.youtube.com/watch?v=AlWFtF06RFo> (Sep. 2015).



**Fig. 2** The user in the middle has decided to change the colour of the drawing pen from *black* to *red*



**Fig. 3** The user in the middle has arrived at the drawing menu and is making a colour selection. At the same time, the user on the *right* has already started drawing with a *red* pen



**Fig. 4** The user in the middle has already changed the colour of the drawing pen (from *black* to *red* to *blue*) and has walked back to his original position. Notably, the user on the *right* has been enforced to continue the rest of his painting with the new pen colour, which might have not been his intention

are able to select items from menus or execute different functions while the system is able to identify the user. In Fig. 5, we demonstrate a mock-up of the solution we propose.

In this paper, we reflect on the need of such interfaces for multi-touch screens and propose a technique in order to improve group work on a MT screen: the combination of chord interaction along with personal action windows for multiple users. Previous research has highlighted the need for a novel set of MT programming toolkits [5] being reusable [6]. Thus, we have designed and developed a novel technique in an open-source library and evaluated its quality for group collaboration with a novel experimental task.

In summary, the main contribution of this research is a) the design and b) the development of multi-touch chords interface along with personal action windows in a



**Fig. 5** The solution we propose: a multi-user chording interaction technique. More people are able to perform different actions simultaneously on a large MT surface

collaborative environment as a seamless identification and interaction technique for large vertical MT displays.

#### **Chords and personal action windows**

In the following subsections, we firstly describe the related work concerning the chord interaction technique and the personal windows interface and then, we demonstrate the need for a toolkit that can handle these multi-user multi-touch techniques and describe what experimental task is needed in order to evaluate this multi-user multi-touch interaction techniques.

#### **Chorded input on multi-touch screens**

Previous multi-touch research has focused on improving single user performance with chorded menus. Lepinski et al. [7] found that directional chords for marking menus performed significantly faster than traditional hierarchical menus. Bau et al. [8] proposed the Arpege contextual technique, in order to make it easy for users to learn multi-touch chord gestures. Wagner et al. [9] propose that even more complex posture chords with multiple fingers can be learned and memorized. Bailly et al. [10] found that the finger-count shortcuts perform better in menu selection, especially with expert users. Kin et al. [11] proposed a finger registration technique that can identify in real-time which hand and fingers of the user are touching the multi-touch device. In this way, they have introduced the Palm Menu, which directly maps commands or operations to different combinations of fingers and they have found that using finger chords has significant performance advantage.

According to a research conducted by Wobbrock et al. [12], when users were asked to propose their own gestures in a participatory design experiment, they claimed that they rarely care about the number of fingers they employ on a MT surface. This seems to contradict the theory behind chorded input we propose in this work. However, in that experiment, users had no previous experience with any MT surface and were novice users. In another research, Bailly et al. [10] prove that finger-count shortcuts can be learned faster than stroke shortcuts, confirming that people also easily learn to “express numbers with their fingers”. According to Kin et al. [11], chorded interaction techniques might be more suitable for users who have already been trained and as Kin et al. [11]

demonstrate, using finger chords has significant performance advantage (compared to popup buttons).

### **Personal multi-touch areas**

There have been many studies investigating territoriality in co-located MT tabletop installations [13] or in remote tabletop settings [14]. According to observations, users usually prefer working on their own personal spaces and even partition the screen in such a way each user has its own private area to work in (as in Morris et al.'s replicated control widgets [15]). Additionally, in a tabletop environment, users tend to interact mostly in the area near where they are sitting [16]. Based on these observations and due to the experience of users in traditional desktop environments, a personal area similar to those of a window was considered during the design of the proposed multi-touch interaction technique.

### **Multi-touch toolkits**

As both our experience and the taxonomy of multi-touch frameworks discussed in Krammer [17] shows, a lot of different multi-touch SDKs and Toolkits have been developed. Some of them are device-related (e.g. Microsoft Surface SDK or Diamond-Touch SDK). On the other hand, there have been presented multi-touch Toolkits such as Python Multitouch (PyMT) or Multi-touch for Java, TouchScript (Unity), which are open-source, and platform-independent systems. There is no doubt that the Multitouch community is vivid and new toolkits are being developed constantly either by practitioners and hobbyists (e.g. Kivy) or researchers (e.g. uTableSDK).

All these multi-touch SDKs/toolkits support multiple touches. However, it seems that developers who designed the toolkits, were not really focused on one of the main characteristics of multi-touch surface, multi-user interaction. Developers did not build tools/widgets that can be used by multiple users simultaneously and thus augmenting collaboration. They relied on other developers for building their own tools by extending the toolkits. Indeed, some really interesting widgets such as multi-touch menus or pie menus etc can be found in the literature. However, once more, these widgets have been primarily developed for single user use and were evaluated accordingly.

Based on the studied literature, there is a need for more generic toolkits that can be used in various situations for co-located collaboration.

### **Experimental tasks in related work**

Apart from the toolkits, there is a need for tasks that evaluate collaborative technologies [18]. In our work, we are more focused on a task aimed mainly at examining the physical performance of the users instead of developing a decision-making or intellectual type of task (such as the job-scheduling task proposed in [18]). There are some experimental tasks in the literature such as the jigsaw collaborative puzzle [19]. Based on the relative literature, drawing stands out as a relatively representative task of a collaborative application for multiple users, either in a co-located environment (e.g. [20] as in our case), or in remote environments (distant drawing, e.g. [21]). But we finally consider using a more simplified drawing task, like that on Dillon's et al. [22] experiment, because researchers need to gather more data on user behaviour, preferences and strategies. Especially for

a multi-user multi-touch interaction technique, researchers need a task that (a) would allow for simultaneous use of multiple users, (b) would urge users to constantly interact and select items from a hypothetical menu (as in the collaborative photo tagging task of Morris et al. [15] but without using any special equipment) and that (c) could be used on a large vertical MT screen and not being restricted for tabletop use.

### **ChordiAction toolkit and interaction design**

In this section, we discuss the proposed interaction technique: we give the algorithm we have implemented and the interaction design of a non-intrusive software user-identification technique which we propose as a solution for simultaneous multi-user interaction on a multi-touch screen.

### **Algorithm**

Our main aim was to promote the diverse and simultaneous use of multi-touch screen by multiple users. Additionally, our chord-interaction toolkit was designed to be configurable and reusable. Developers or researchers can customize the toolkit to adjust it to their own needs or experiments.

In this subsection, we describe an abstract algorithm of what we have implemented:

```

1  configure chord_layer;
2  enable chord_layer;
3  ....
4  @Event chord_triggered{
5    ( X, Y ) = location(chord_triggered)
6    show chord_interaction_area_at( X, Y );
7    reserve_space_around( X, Y );
8    do{
9        I = count( #touches inside reserved space );
10   }until no_touches for S seconds;
11   perform_action( I );
12 }
```

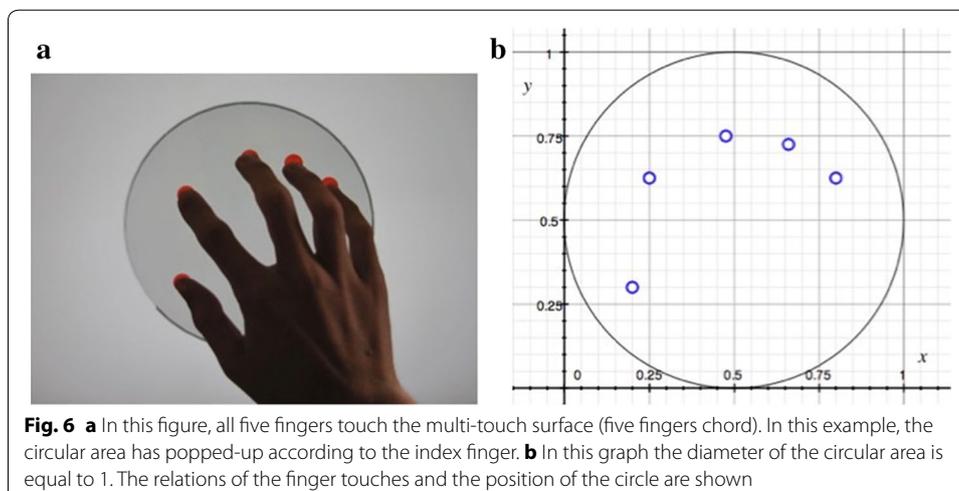
In the beginning, we have to define how the area in which we will apply our chord will be triggered. There are different options such as a double-tap or a long-tap event. In addition, we have to define other details such as the number of seconds that the system will wait in order to receive the chord or where the chord interaction area will be placed in relation to the interaction. In line 2, developer creates an event handler that monitors interactions and when the interaction that triggers the chord area takes place the event is triggered (line 4). Then, the system locates the place where the event took place (line 6) and reserves the space (line 7) in order to let the user perform his/her chord. Depending on the number of fingers inside the reserved area (line 9) the system performs the appropriate action (line 11).

### Interaction design and application development

Our goal is to allow user to work (performing actions e.g. selecting an option from a menu) together in parallel, independently or sequentially, without the need to negotiate turns. Initial experiments [23] proposed the transition from a fixed selection technique, where user simply clicks/touches an item in a static menu in order to select it, to multi-user chorded selection where user makes use of a circular chording area that is temporarily (for a number of seconds) reserved whenever he/she touches the MT screen. In a multi-user environment, users should dynamically reserve multiple small circular areas, which could be the size of ones' palm (diameter is 15 cm). In that small area, user has to perform a chord for the selection of the appropriate menu item or function. With the support of a status indicator on the menu bar, users are able to understand which menu item must be chosen and how many fingers they have to touch on the surface in order to select it.

Multiple users are able to touch different parts of the screen and then different small areas will be temporarily reserved for chorded modifiers accordingly. The reserved area is a circular area around the first touch of the user and it is about the user's hand size, being easy for the user to touch the appropriate number of fingers and thus applying the chord responsible for the selection of the menu item he/she desires. We have designed a multi-user MT component that allows users to touch multiple fingers on any place of the display. Each time a user makes a selection, the appropriate action/function can be activated.

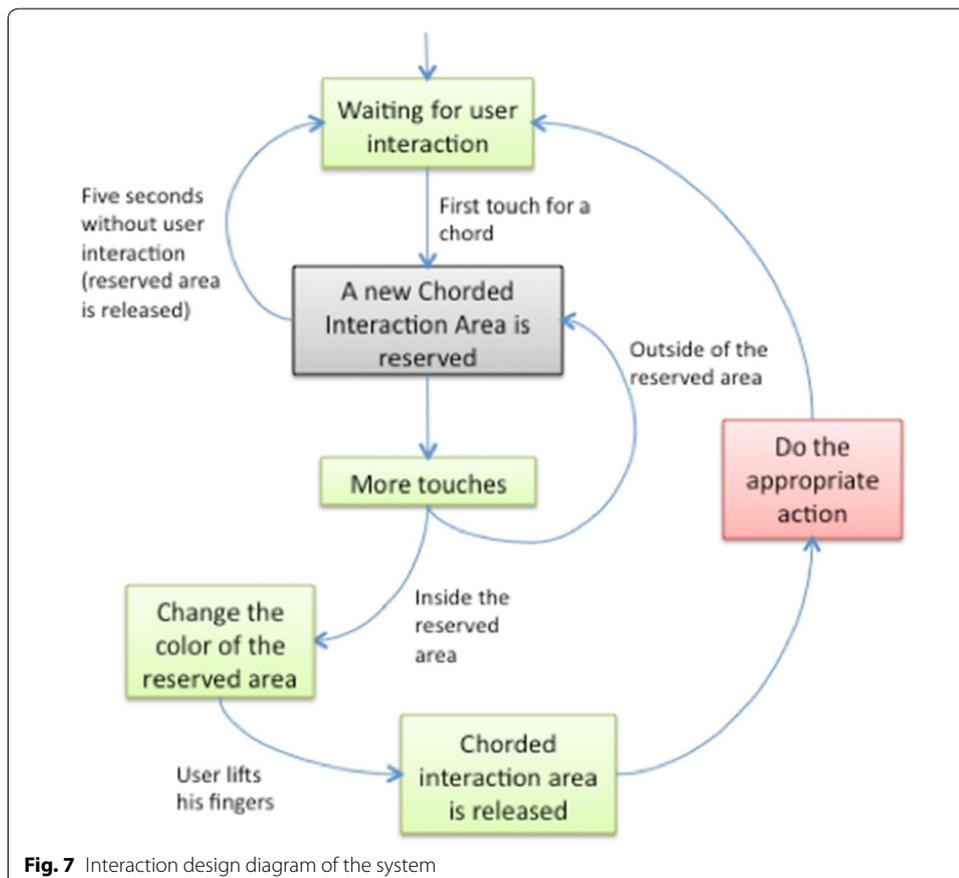
Our first approach was to define the circular area having the point of the first touch as its center. But this proved to be ineffective as none of the users used initially their middle finger and thus the circular area was misplaced. According to Epps et al. [24], the index finger is the most common hand gesture and it is used in more than 70 % of the times a user interacts with touch screen devices. In a drawing application this percentage reaches 90 % of interaction. Based on our experience and on Epps et al. study, we finally positioned the circle to the right and below the first point of touch. In Fig. 6a the circular area inside which chords can be articulated is depicted. In this case, the circular area pops-up according to the index finger of the user. Additionally, in Fig. 6b all possible

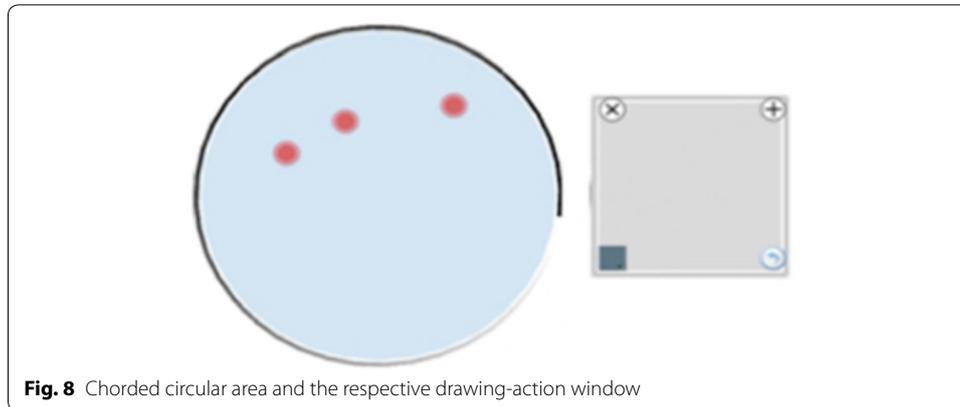


finger-touches and their appropriate positions in relation to the circular chorded area are shown. According to our updated implementation and despite the fact that we propose the use of the index finger as a trigger for the position of the chord interaction area, the circular area pops-up relatively to the finger designated by the developer/application. We did not choose an approach such as calculating the convex hull of the touches and choosing the centroid of the very bounding box as the center of the circular area because we wanted to avoid a prior hand registration session (as in Kin et al.'s study, [11]).

According to the interaction design diagram in Fig. 7, the system waits initially for the first finger touch. When this occurs, a new chorded area is reserved. User has to touch the appropriate number of fingers in order to articulate the corresponding chord. When the user has lifted all his fingers from the screen, the chord interaction area is released, the system processes the chord and the appropriate action window or function is enabled. As it is shown, multiple chording areas are allowed and thus multiple users can interact simultaneously.

In Fig. 8, the chorded area in which the user touched three fingers along with a personal window that pops-up when user has finished his chord is displayed. This type of interaction is in accordance to Kurtenbach and Buxton's [4] suggestion that even complex interactions should be popped up at any location in larger screens. The window has also a close button (x button), an add button (+button) and an undo button. The pop-up





windows are dynamic: (1) they can be moved anywhere, and (2) users can change their sizes. Based on Bier et al.'s [25] toolglass widgets (see-through user interface), these windows are transparent and permit specific action, i.e. drawing with blue colour due to the fact that the three-fingers-chord was chosen.

The following lines of code demonstrate the use of ChordAction toolkit in an example application:

```

1   from pymt import *
2   from chordAction import chordAction
3   if __name__ == '__main__':
4       w = MTWindow()
5       cA_object = chordAction(interaction_style='double_tap')
6       w.add_widget(cA_object)
7       @cA_object.event
8       def chord_done(position, selection):
9           print "Interaction at position", position
10          print "The selection was:", selection
11          #do whatever you want with the specific chord
12  runTouchApp()

```

In order to make use of the ChordAction toolkit we have to import Pymt library and the toolkit module as in line 2. In the fifth line a new ChordAction object is created. The `interaction_style` is the only parameter that is needed in order to create a ChordAction object. In this example the interaction style is 'double\_tap'. That means that in order for the user to enable the chord interaction technique the user has to double tap the screen. Other possible choices are 'single\_tap' and 'long\_tap' (where user has to continuously touch the screen for more than 0.5 s to enable chord interaction).

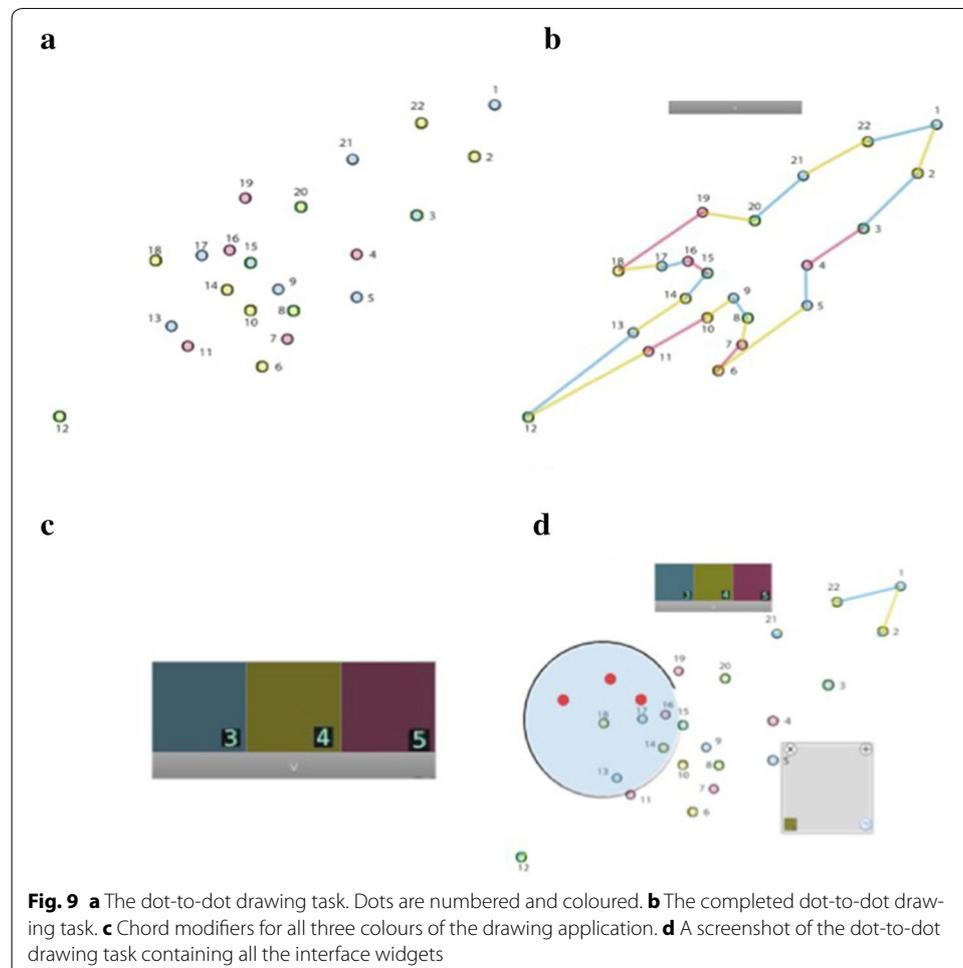
In line 6 the ChordAction is added in the widget tree. From this point on, whenever a user double-touches the screen a circle within which user has to articulate the desired chord pops up. When user lifts all his fingers from the circular area where he/she screen

can articulate a chord, the ChordiAction toolkit creates an event (line 7). In order to catch the event and do the appropriate actions the chord\_done function is used as in line 8. This event returns two variables, the position where the chord was articulated and the selection that has been made (number of fingers). In this simple example, in line 8 and 9 the appropriate values are just printed on the screen for every chord made by the user.

### Stimulating interaction with chords in an experimental task

Based on our literature review, Dillon's et al. [22] experiment is close to our needs for gathering more data on user behaviour and preferences. By extending this experiment, we were led to a dot-to-dot type of drawing task. The type of application we developed allows for collaboration along with interference among users during simultaneous interaction on the MT screen, the ideal combination for our experiment. Moreover, users are familiar with this kind of task and they can focus more in the interaction technique and not in trying to understand the task.

As it is depicted in Fig. 9a, the dots of the dot-to-dot task are coloured and numbered. We asked from the users to connect the dots sequentially. Moreover, one more rule was added in order to complete the task (Fig. 9b). Each line should have the colour of the dot with the higher number. For example, if the user has to connect the blue dot with the



**Fig. 9** **a** The dot-to-dot drawing task. Dots are numbered and coloured. **b** The completed dot-to-dot drawing task. **c** Chord modifiers for all three colours of the drawing application. **d** A screenshot of the dot-to-dot drawing task containing all the interface widgets

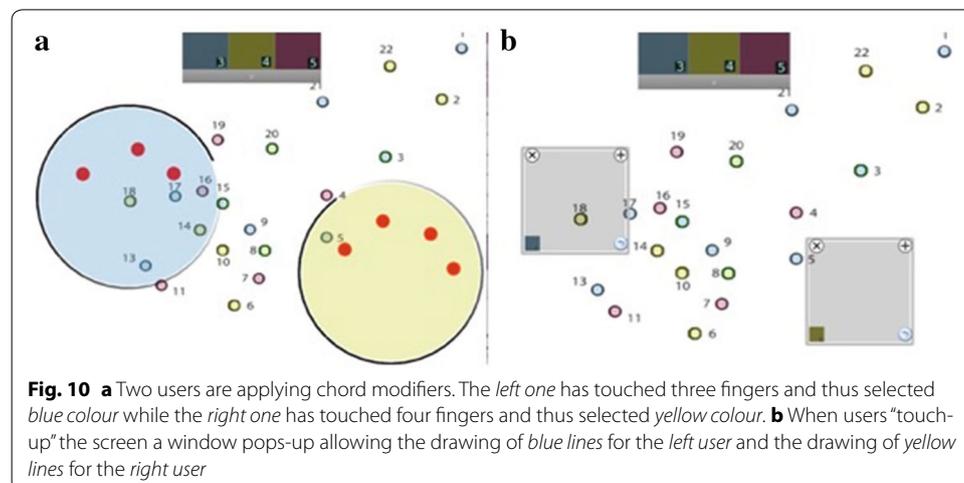
number one to the yellow dot with the number two, the line between them should be painted yellow.

In Fig. 9c the available colours are depicted. There is a status indicator that is presented on the upper part of the MT screen (Fig. 9d) in order to help users to remember chord modifiers and the respective colours. As it is shown in the Fig. 9c, with three touches users are able to draw a blue line. With a four-finger chord users can draw a yellow line, while using all five fingers allows them to draw a purple line. Even though the status indicator seems to be obsolete, it may be useful for new users that have not been familiarized with the chord modifiers technique.

The main requirement for the first experimental user task was to enforce users to perform several chords, as well as to negotiate the interaction over shared screen spaces and tasks. Even though this is a simplified drawing application with only one type of shape, the aspect of using chord modifiers in a collaborative environment is sufficiently represented, due to the fact that users must constantly use chords to change the colour of the line to be drawn inside the personal windows. Additionally, as Fig. 9a demonstrates, two neighboring dots are always of different colour and thus users are forced to articulate chords each time they want to draw a new line, since two consecutive lines must have different colours.

In Fig. 10a, a screenshot of the drawing application where two chorded interactions take place is depicted. In the left circle, the first user touches only three fingers on the screen. Each touch is presented with a red circle. The circular area turns into blue colour to notify him that by touching his three fingers he has chosen the blue colour. On the other hand, the second user touches down four fingers. The system notifies him accordingly by turning the circle into yellow colour. Notably, users are always informed about the number of the fingers they touch and thus their choice. Moreover, the system permits transitions between the available choices in real time. Note that the system works no matter how many users interact with the screen. Additionally, we expect that an experienced user could use his both hands to draw simultaneously, the way an experienced pianist or typist plays/works with both hands on the keyboard.

Figure 10b shows what happens when user lifts all his fingers from the MT screen. Users should have their own private pop-up action window in which they can perform



the action they selected using the chorded modifier. That is, user can only draw in such pop-up windows. Even this seems as a restriction for the user, it makes it feasible for more than one user to work simultaneously performing different actions on the screen. As it is also shown in the figure, in the left window, only blue lines can be drawn (as indicated from the small icon in its low-left corner). Alternatively, the right window is a drawing-yellow-line one based on the chorded selection of the user (Fig. 10a).

The task described above makes sure that users are going to make different actions (use different chords) since, for each line to be drawn, a different action is needed (use different windows). Furthermore, users should coordinate their actions in order to effectively integrate the task as long as (in this case, both chording circular areas and windows require space and thereby) negotiation is needed so as to avoid a cluttered work-area for all users. This type of task fulfills all our criteria being an effective instrument to evaluate chord-interaction as long as other techniques for exploring users' collaboration on a MT surface.

### Exploratory study

The exploratory study was divided into two sections. In the first part, five males and one female postgraduate informatics students with average age 25 were recruited from the local university and were trained on pairs on the MT drawing application until they could not improve their time more than 5 % from their best time. The training along with the familiarization of the experimental task took 1 h to participants. They were asked to be *as fast as possible* during the dot-to-dot drawing task and were observed and videotaped using a handheld camera. In the second part, twelve users (two males and ten females) with an average age of 18 were recruited in order to participate and complete the dot-to-dot task without any restrictions concerning time. In addition, users were not enforced to articulate chords for every action they performed as with the first part of the study in order to work together in a more friendly, creative and playful atmosphere. Qualitative results gathered by researchers, such as notes and observations and were incorporated into the final results. The interactions on videos along with users' conversations while completing the task were analyzed and manually coded using the grounded coding theory [26] and each finding is described in the following sections. We are mainly focused in how users behaved during interaction in order to avoid collisions and whether they worked in parallel or not, being aware of what others were doing. Moreover, we discuss their conflict resolution techniques and whether chording interaction technique is difficult to learn. Thus, the coding was connected to behaviours/interaction style, strategies, user participation, and user awareness. In Table 1 the codes along with the categories that occurred as a result of our exploratory study are presented. In the following subsections they are further analyzed.

### Divide and conquer—partitioning the screen or sharing the workload

During our exploratory experiments with the dot-to-dot drawing task, we observed that despite the fact that users were able to work on a private area (since, by articulating the appropriate chord modifier, they were given a specific window in which they could draw), they tended to *partition the screen* notwithstanding. Most of the couples used *verbal communication* before the task in order to either divide the screen or the

**Table 1 Categories and codes that occurred during the experimental task**

Categories	Codes
Behaviours/interaction style	Verbal communication Partitioning the screen
Strategies	Divide the screen Divide the workload
User participation	Parallel and synchronous interaction In-turn type of sequential interaction
User awareness	Articulate the chords in the corner Articulate the chords in the main interaction area

workload of the drawing task so as to complete it as fast as possible. Thus, there were users followed the “*divide the screen*” strategy and therefore said “I connect all the dots in my area (in the right half of the screen) and you connect the dots in the left half.” and users that chose the “*divide the workload*” strategy: “There are 22 dots, you connect the first 11 and the rest are mine.” However, as it can be seen in Fig. 10a, a lot of dots are accumulated in the center of the pattern purposely, not allowing efficient partitioning of the screen, in an effort to observe users’ interaction during colliding situations. There was no couple interacting without a previously developed plan. We suppose that by asking the users to be as fast as possible, we were led to these two different techniques in order for users to improve their performance. Of course, as soon as they embarked on the task different behaviours were observed.

#### Space negotiation and conflict resolution

Users became aware that entering into others’ windows could lead to mistakes or confuse them and they tried to avoid this from happening. Additionally, they tried to *articulate the chord modifiers in the corner of the screen* near them and then they preferred to drag the respective action window in the area to be drawn instead of *articulating the chord exactly where they wanted to draw* from the beginning (Fig. 11).

Users were reluctant to move their window to others’ personal space. When both of the users had to draw in the center of the screen they almost changed their strategy. For instance, when one user was drawing a line, the other one was in the corner of the screen trying to form the appropriate chord (as depicted in Fig. 11c). And when he had his window popped-up, he was moving it while the first user was articulating his own chord in his own corner. That said, interaction seemed to change from *parallel and synchronous*



**Fig. 11** The user in the *right* articulates the chord in the low *right corner* of the screen and then moves the action window in the appropriate position to draw the line

drawing to an *in-turn type of sequential interaction* even though there was constant input from both of the users simultaneously.

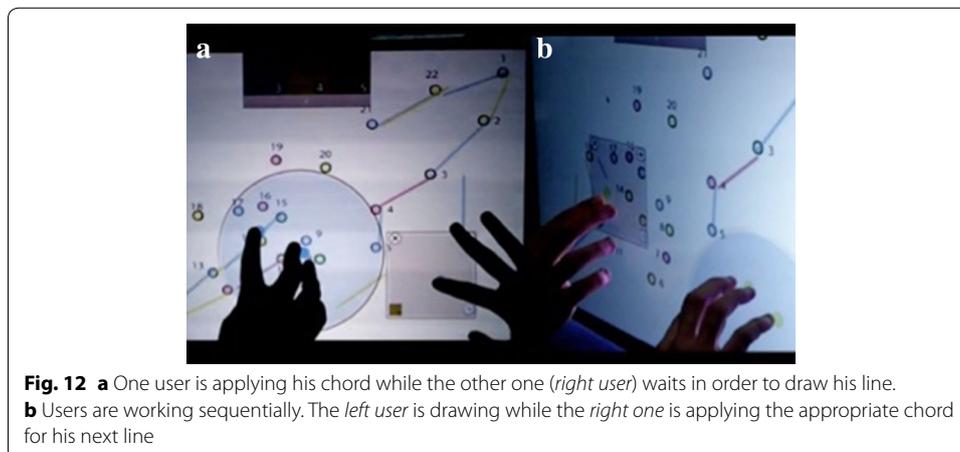
Moreover, there were times when one user had completed all his/her work (for example connecting all the dots, from 1 to 11) and then he/she was just waiting for the other user to connect the rest of the dots (an example can be seen in Fig. 12a). User was reluctant to help the other user because the dots were not in his own space and he would not like to penetrate to other's territory.

Users were hesitant to simultaneously touch the shared controls. For example, in some cases one user could have enlarged his window more than normally expected, breaking the territory rules. In these situations, users avoided to close others' window and withdrew waiting for others to close it up or continued drawing wherever there was enough space for them. Even if Peltonen et al. [27] claim that these situations prove to be funny and produce enjoyment for users interacting with an entertainment installation, we are convinced that in a more businesslike environment they could lead to frustration. In one of these moments during our experimental task, one of our users said "This is not working!".

## Discussion

### Collocated collaboration with chords and personal windows

Developing tools and applications for a multi-touch surface is considered to be a complicated procedure due to the limitations and challenges of a larger multi-touch screen. We propose the use of ChordiAction, a collaborative user interface toolkit that can be used in various situations for co-located large-scale MT screens. According to Elliott and Hearst [3], larger multi-touch screens need novel interaction techniques in order for the users to interact in larger work-areas. Wall-sized larger displays can be used up close by several users at a time, they offer high resolution for working up close, and they provide sufficient space for varied collaboration styles [28]. The proposed technique in this work shortens the distances and thus can improve selection time or help in avoiding possible user conflicts (as in Figs. 2, 3, 4). Chord interaction techniques have been also used in previous studies (e.g. [7] or [8]), but the main focus of the researchers was on single-user mode, while we aim at collaboration among users while interacting in parallel on a MT screen.



In addition, by employing personal windows, the system we propose is able to identify the user, or alternatively the user is able to perform simultaneously different actions in parallel with other users. We have also employed a transparent layer as a see-through interface such as the one in Bier et al's [25] (toolglass widgets) that lies between the application and the fingers of the user. This type of window makes it feasible for the system to identify the user (personal window) or the appropriate action (action window) unobtrusively, since it is a result of the appropriate chord articulated previously by the user. Moreover, in the case of using our system for menu selection, chord interaction along with action windows can be considered as a virtually replicated menu interface, since every user can select from a (virtually positioned) menu in a position that is useful for him/her. In our system, users instead of interacting with a centralized menu that (being static) cannot be shared efficiently (e.g. the example discussed in introduction section), they are able to perform different actions simultaneously without interference among them.

To conclude, the interaction technique we propose (a) allows for simultaneous diverse interactions from multiple users, (b) shortens the distances and thus can improve selection time in larger multi-touch screens (in the case of a menu-selection technique), (c) helps avoiding possible user conflicts by identifying the user or the action due to personal action windows, (d) is a low-cost software solution that works unobtrusively without any training sessions or additional equipment. Researchers and developers can easily use the proposed interaction technique by using the ChordiAction toolkit in their multi-touch applications.

#### **Dot-to-dot collaborative task**

In addition to the advantages of the proposed method we described in previous sections, we introduced an experimental task in order to evaluate multi-user interaction toolkits-like the one proposed in this work- on multi-touch surfaces. Based on our observations, the dot-to-dot collaborative task we chose while developing the evaluation strategy of the proposed interaction techniques proved to be a valid decision. Despite the fact of being mainly a physical performance task—a task that involves physical behaviour as opposed to symbolic, or mental manipulations, [18] or [28]—it produced valuable results and shed light to users' strategies while collaborating on the screen. Users tried to improve their time, used verbal communication, pointed others what to do, helped each other, worked in parallel or isolated in a partition of the screen and tried to resolve conflicts. Users were mainly focused on the interaction and on completing the puzzle-style task and saw it as a battle between them and the other teams that had completed the task previously with a better time. Based on our experience with the dot-to-dot collaborative task, its main advantages are: (a) it is a simple to administer task, (b) it is a traditionally played game and thus does not demand knowledge from the users, (c) it is fast and enhances competitiveness among different teams or cooperation among teams' members, (d) it demands coordination skills and it also demands from the users to be aware of what other users do, (e) it allows for either working in parallel or performing joint work (as the task used by [28]) and (f) it is effective in producing reproducible results.

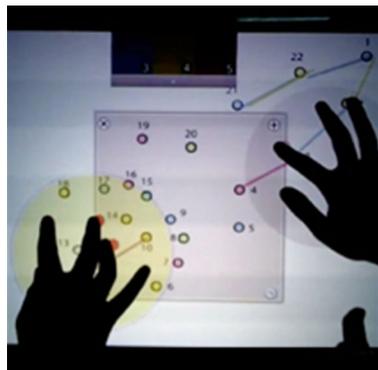
### Limitations

Despite the fact that our system with chords and personal windows was evaluated on a 24 inches it was designed for large-scale MT systems. Users employed verbal communication or withdrawal mechanisms, such as being reluctant to interfere with others' actions or entering their territory in order to avoid collisions. As in Peltonen et al's [27] large interactive display, there were times when something unexpected could happen. For example, some windows would accidentally blow up (as in Fig. 13). A collaborative task in which users are asked to perform as fast as possible demand higher system robustness compared to an entertainment installation where users tend to have fun in the case of an unexpected event or error. In order to deal with window overlapping problems we considered the window lastly touched to be the active one. We did not employ any restrictions on where the window should be moved or window collision techniques in order to explore how users would interact in such cases. However, as Hornecker et al. [29] notice, fluidity of interaction and switching of roles between co-located users is more preferable than enforced sequential interaction or predetermined territories in MT surfaces. As they propose, "instead of trying to eliminate conflicts, simply aim to increase the resources for dealing with and negotiating interference".

One more limitation of our menu selection technique is that it allows up to eight different menu items considering that we have ten fingers and we must use at least two fingers for basic interaction. But, as it is shown by Kiger [30] eight different items in a menu is an effective number of menu elements and as far as the depth of the menu is concerned, a MT application could reserve a suitable chord of fingers, which could permit it. Instead of directional chords that were used by other researchers [7, 8], we propose the use of static chords as a much easier technique for the majority of users.

### Future work

In our future work, we are going to evaluate our work quantitatively and compare it to other interaction techniques designed for large MT surfaces. We plan to evaluate the proposed interaction techniques in an educational context as a case-study. Additional work is planned in order to measure the effects of using the toolkit by advanced versus trained users. Additionally, we would like to go one step further on the chorded interaction allowing the use of either interchangeable interaction or bimanual chords [11] for



**Fig. 13** A window bothers both of the users to articulate their chords

multiple users in a larger MT screen and thus giving more free space to users in order to articulate the chords and furthermore increase users' selectable space. Chorded interaction would also function in network-connected tabletops as a synchronous collaboration technique for multiple users interaction since the essential guidelines by Tuddenham and Robinson [31] for effective collaboration between distributed tabletops were followed during the design of the chord interaction technique.

## Conclusion

In this work, we proposed and implemented a chording technique that enables higher levels of multi-user diverse interactivity, collaboration and awareness when used along with personal action windows. Users' interaction techniques were investigated and issues such as conflict resolution strategies were discussed. The main contribution of this work is the design and implementation of this novel seamless identification and interaction technique that is scalable and supports diverse multi-touch interactions especially on larger MT surfaces.

We evaluated this technique in vertical multi-touch surfaces, but it can be used in tabletop systems as well, since it was designed bearing in mind the general characteristics of MT surfaces (multi-user interaction, user orientation, user movements etc.) and how users work on them no matter their setting.

In this research, we examined the idea of using chords along with personal/action windows in a MT collaborative environment for menu selections as a non-intrusive technique, as well as we designed and implemented an easily repeatable synthetic experimental dot-to-dot task that demonstrates the potential of this technique and can be used as a tool to evaluate other techniques on large MT surfaces by other researchers. This research also demonstrates the need for designing and implementing toolkits and applications that are dedicated to MT interaction style and take advantage of the unique characteristics of a MT surface. For instance, although a chording system was absent from MT Toolkits, we believe that MT dedicated interaction techniques like this, should be integrated in future MT Toolkit updates for being: (a) a simple ad hoc solution, (b) fast in comparison to traditional interaction techniques, (c) atomic and thus suitable for multi-user interaction and (d) flexible and thus scalable.

In this paper, we reflected on the need of such interfaces for multi-touch screens and demonstrated that a combination of chord interaction along with personal action windows for multiple users can be a technique suitable for group work on a larger MT screen.

### Authors' contributions

IL had the main idea and wrote most of this paper. He has also developed the toolkit and conducted the experimental task. KC contributed in structuring of the paper and polishing the introduction section. He also participated in discussions and provided corrections and extended feedback. LJ helped with the experimental task and valuable discussions. IL was also the main responsible for the data collection and analysis. All authors read and approved the final manuscript.

### Author details

<sup>1</sup> Department of Informatics, Ionian University, Tsirigoti Square 7, Corfu, Greece. <sup>2</sup> Department of Computer and Information Science, Norwegian University of Science and Technology, Sem Sælands vei 7-9, 7491 Trondheim, Norway.

### Acknowledgements

We would like to thank our pilot users who participated in our experiments.

### Competing interests

The authors declare that they have no competing interests.

Received: 26 November 2015 Accepted: 22 June 2016  
 Published online: 10 September 2016

## References

- Haller M (2008) Interactive displays and next-generation interfaces. In: *Becta emerging technologies for learning*, vol 3, p 91–101
- Nacenta MA, Pinelle D, Stuckel D, Gutwin C (2007) The effects of interaction technique on coordination in tabletop groupware. In: *Proceedings of Graphics interface*. ACM Press, New York, p 191–198
- Elliott A, Hearst M (2002) A comparison of the affordances of a digital desk and tablet for architectural image use tasks. *Int J Hum Comp Stud* 56(2):173–197
- Kurtenbach G, Buxton W (1993) The limits of expert performance using hierarchic marking menus. In: *Proceedings of the INTERACT'93 and CHI'93 conference on human factors in computing systems*. ACM, New York, p 482–487
- Wigdor D, Fletcher J, Morrison G (2009) Designing user interfaces for multi-touch and gesture devices. In: *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems—CHI EA'09*: 2755
- Luyten K, Vanacken D, Weiss M, Borchers J, Izadi S, Wigdor D (2010) Engineering patterns for multi-touch interfaces. In: *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS'10)*. ACM Press, New York, p 365–366
- Lepinski GJ, Grossman T, Fitzmaurice G (2010) The design and evaluation of multitouch marking menus. In: *Proceedings of the 28th international conference on human factors in computing systems—CHI'10*, ACM Press, New York, p 2233–2242
- Bau O, Ghomi E, Mackay W (2010) Arpege: design and learning of multifinger chord gestures, CNRS-Université Paris Sud—LRI. *Rapport de Recherche* (1533)
- Wagner J, Lecolinet E, Selker T (2014) Multi-finger chords for hand-held tablets: Recognizable and memorable. In: *Proceedings of the 32nd annual ACM conference on human factors in computing systems*. ACM Press, New York, p 2883–2892
- Bailly G, Lecolinet E, Guiard Y (2010). Finger-count and radial-stroke shortcuts: two techniques for augmenting linear menus on multi-touch surfaces. In: *Proceedings of the 28th international conference on Human factors in computing systems—CHI'10*. ACM Press, New York, p 591–594
- Au OKC, Tai CL (2010) Multitouch finger registration and its applications. In: *Proceedings of the 22nd conference of the computer-human interaction special interest group of Australia on computer-human interaction, OZCHI'10*, ACM Press, New York, p 41–48
- Wobbrock JO, Morris MR, Wilson AD (2009) User-defined gestures for surface computing. In: *Proceedings of the 27th international conference on human factors in computing systems (CHI'09)*. ACM Press, New York, p 1083–1092
- Scott SD, Sheelagh M, Carpendale T, Inkpen KM (2004) Territoriality in collaborative tabletop workspaces. In: *Proceedings of the 2004 ACM conference on computer supported cooperative work*. ACM Press, New York, p 294–303
- Tuddenham P, Robinson P (2009) Territorial coordination and workspace awareness in remote tabletop collaboration. In: *Proceedings of the 27th international conference on human factors in computing systems*, ACM Press, New York, p 2139–2148
- Morris MR, Paepcke A, Winograd T, Stamberger J (2006) TeamTag: exploring centralized versus replicated controls for co-located tabletop groupware. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM Press, New York, p 1273–1282
- Ryall K, Forlines C, Shen C, MR Morris (2004) Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In: *Proceedings of the 2004 ACM conference on computer supported cooperative work*. ACM Press, New York, p 284–293
- Kammer D, Keck M, Freitag G, Wacker M (2010) Taxonomy and overview of multi-touch frameworks: architecture, scope and features. In: *Proceedings of the EICS'10 workshop on engineering patterns for multi-touch interfaces*
- Tan DS, Gergle D, Mandryk R, Inkpen K, Kellar M, Hawkey K, Czerwinski M (2008) Using job-shop scheduling tasks for evaluating co-located collaboration. *Personal Ubiquitous Computing* (2008), 12, p 255–267
- Kraut RE, Gergle D, Fussell SR (2002) The use of visual information in shared visual spaces: informing the development of virtual co-presence. In: *Proceedings of the ACM conference on computer-supported cooperative work 2002*. ACM Press, New York, p 31–40
- Zhang H, Yang XD, Ens B, Liang HN, Boulanger P, Irani P (2012) See me, see you: a lightweight method for discriminating user touches on tabletop displays. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM Press, New York, p 2327–2336
- Ishii H, Kobayashi M (1992) ClearBoard: a seamless medium for shared drawing and conversation with eye contact. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM Press, New York, p 525–532
- Dillon RF, Edey JD, Tombaugh JW (1990) Measuring the true cost of command selection: techniques and results. In: *Proceedings of the SIGCHI conference on human factors in computing systems: empowering people*. ACM Press, New York, p 19–26
- Leftheriotis I, Chorianopoulos K (2011). Multi-user chorded toolkit for multi-touch screens. In: *Proceedings of EICS'11, ACM SIGCHI symposium on engineering interactive computing systems*, ACM Press, New York, p 161–164
- Epps J, Lichman S, Wu M (2006) A study of hand shape use in tabletop gesture interaction. In: *CHI'06 extended abstracts on human factors in computing systems*. ACM Press, New York, p 748–753

25. Bier EA, Stone MC, Fishkin K, Buxton W, Baudel T (1994) A taxonomy of see-through tools. In: Adelson B, Dumais S, Olson J (eds) Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence (CHI '94). ACM Press, New York, p 358–364
26. Glaser BG, Strauss AL (2009) The discovery of grounded theory: strategies for qualitative research. Transaction Books, Piscataway
27. Peltonen P, Kurvinen E, Salovaara A, Jacucci G, Ilmonen T, Evans J, Oulasvirta A, Saarikko P (2008) It's mine, don't touch!: interactions at a large multi-touch display in a city centre. In: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. ACM Press, New York, p 1285–1294
28. Jakobsen MR, Hornbæk K (2014) Up close and personal: collaborative work on a high-resolution multitouch wall display. *ACM Trans Comp Hum Interact (TOCHI)* 21(2):11
29. Hornecker E, Marshall P, Dalton NS, Rogers Y (2008) Collaboration and interference: awareness with mice or touch input. In: Proceedings of the 2008 ACM conference on computer supported cooperative work. ACM Press, New York, p 167–176
30. Kiger J (1984) The depth/breadth trade-off in the design of menu-driven user interfaces. *Int J Man Mach Stud* 20(2):201–213
31. Tuddenham P, Robinson P (2007) Distributed tabletops: supporting remote and mixed-presence tabletop collaboration. In: Second annual IEEE International workshop on horizontal interactive human-computer systems (TABLETOP'07). IEEE, New York, p 19–26
32. Leftheriotis I, Chorianopoulos K, Jaccheri L (2012) Tool support for developing scalable multi-user applications on multi-touch screens. In: Proceedings of ITS' 2012, ACM international conference on interactive tabletops and surfaces, ACM Press, New York, p 371–374

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---