Interaction Workspaces: Identity Tracking for Multi-user Collaboration on Camera-based Multi-touch Tabletops

KyungTae Kim, Tejas Kulkarni, and Niklas Elmqvist, Member, IEEE

Fig. 1. Mockup screenshot of a bus route map application with two users (left), each one with a personal interaction workspace (right).

Abstract—The ability to distinguish between individual users in collaborative visualization is important for avoiding interference and facilitating cooperation. However, identification can be difficult on camera-based tabletops, where we typically cannot distinguish which tracked touch points belong to which user. Additional mechanisms are needed, such as video input, capacitance, or social and software protocols. We propose a model for the identification process on tabletop displays, and then use this model to suggest a software-based mechanism called interaction workspaces for supporting user identification. An interaction workspace is a movable and resizable region on the full visual space that belongs to a particular user, and inside which only that user is assumed to interact.

Index Terms—Digital tabletop displays, multi-touch, identity tracking, single-display groupware, horizontal interaction.

1 INTRODUCTION

Collaborative visualization may often benefit from the system being able to distinguish between the identities of the different users collaborating on the system [14], both in order to avoid interference between their respective actions as well as to facilitate role-specific operations and cooperation between the users [3, 5, 15, 16]. Consider a scenario with two users working on developing a bus route map for an urban area on a single-display groupware [17] device (Figure 1). When user A tries to sketch a route from point P1 to P2, while user B sketches a route from point P3 to P4, a system without user identification might recognize these two gestures as coming from a single user and may perform an incorrect operation, such as zooming the whole map. Another situation where user identity is critical is when multiple users are working on developing unique routes in the same visual space. Without user identification, if one user wants to undo modifications to the routes, the system will not be able to distinguish between modifications by different users. The system may inadvertently affect other users’ work in that space, causing interference between the users.

Multi-touch [9] tabletop displays are becoming increasingly popular for this kind of co-located collaborative work [8] due not only to the natural user interface afforded by the user’s own fingers [6], but also by virtue of camera-based tabletops being relatively inexpensive and easy to build [7]. However, while tabletop displays that rely on capacitance tracking, such as DiamondTouch [2], do support user identity, the more expensive camera-based tabletops [7] (of which Microsoft Surface is an example) do not. Camera-based tabletops merely track multi-touch data in the form of “blobs”, which represent points of contact on the tabletop surface by objects such as fingers. In order to distinguish this multi-touch data as belonging to different users, these systems must be augmented with additional mechanisms. Examples of such mechanisms include simple distance-based heuristics, camera-based image processing [3], social protocols [10], or software mechanisms such as storage bins [12] or user territories [13]. However, all of these mechanisms are currently independent and cannot easily be combined, compared, or contrasted.

In this position paper, we propose to model the identification problem as a general transformation process which accepts multi-touch data and classifies this into multi-user multi-touch data. The identity classifier itself depends on the available hardware, software, and physical properties of the tabletop system. The model is general enough to describe all of the above known identification mechanisms for tabletop displays. Extending on previous work, we also propose a simple software-based classifier called interaction workspaces that permit fine-grained access control of the visual space by restricting users to personal work areas (Figure 1), but which relies on a social protocol for high-level access control (i.e., protecting your own workspace).

This position paper is structured as follows: First, we discuss the background of multi-user systems and the existing literature. We then describe the design of our process model that assigns unidentified blobs to their respective owners. We go on to present our extension to software-based identity classifiers: interaction workspaces. We close the paper with our conclusions and plans for future work.

2 BACKGROUND

As computers become more integrated in society, there has been increasing demands for co-located collaboration in multi-user environments [14]. However, traditional desktop computers typically use a standard keyboard and mouse setup that allow only a single active user at a time, rendering them unsuitable for multi-user collaboration [17]. Recent advances in both display and input technology has made a new generation of collaborative devices available for co-located collabora-
tion [14]. Among these, digital tabletop displays are particularly suitable for co-located collaboration due to their similarity to traditional tabletops, their natural interaction using fingers or stylus, as well as their obvious collaborative affordances [2, 18]. Furthermore, camera-based tabletop displays, such as FTIR [7], are inexpensive and relatively simple to build. However, a necessary—but, as we shall see below, not sufficient—requirement for these devices to support multiple concurrent users is that they first support multiple concurrent touch points on the display. This property is known as multi-touch sensing [7, 9].

2.1 Recognizing User Identity

Multi-touch systems are generally quoted as also being multi-user capable due to being able to distinguish between multiple points of contact, regardless of whether these points stem from a single user or multiple users. However, this is a qualified truth. While capacitance-based tracking, such as DiamondTouch [2], can trivially distinguish contact points for up to four users, camera-based tabletops generally cannot. They detect each point of contact (from a finger, hand, or other object) as a “blob” on the surface of the tabletop, but have no way of assigning a user identity to each individual blob [7].

However, the ability to recognize and track user identity in co-located collaboration may be critical to resolve interference and conflicts between users. For example, a system without identity tracking will not be able to tell the difference between a single user interacting with both hands, or two users each interacting with a single hand. On a higher level, the lack of identity tracking may cause conflicts for shared objects on the display, and does not support assigning roles to users. Groupware has typically relied on social protocols—spoken or unspoken conventions between users about object ownership—for managing these problems, but this is not sufficient for many situations [5, 10].

2.2 Classification of Touch Technologies

As argued above, there is clearly a distinction between supporting multiple concurrent touch points and supporting multiple concurrent users. In Figure 2, we classify existing touch technologies based on the number of concurrent users (single or multiple), as well as the number of concurrent touches they support per user (single or multiple).

![Fig. 2. Classification of existing touch technologies depending on their multi-touch and multi-user capabilities.](image)

As we can see, camera-based multi-touch systems (as well as and mobile touch screen devices such as the iPhone), fall under the category of multi-touch single-user systems. These types of system face the problem of user identification as described above. A select subset of touch technologies, such as primarily DiamondTouch [2], are not only multi-touch but also multi-user systems.

2.3 Multi-user Multi-touch Tabletops

Most single-display groupware [17] projects in the literature which need multi-touch and multi-user support utilize the aforementioned DiamondTouch [2] tabletop, developed by Mitsubishi Electric Research Laboratories (MERL). The device supports up to four unique users, each capacitively connected with a receiver in their chairs, and detects touch points as users close the circuit by touching the tabletop. Because of this design, the display needs to be front-projected. Despite the widespread success of the DiamondTouch device, both in research as well as industry applications, this tabletop is limited to a specific form factor, whereas camera-based tabletop technologies are more flexible in terms of size, and also more inexpensive to build [7].

2.4 Separating Multi-touch Data into Multiple Users

Clearly, the vast majority of multi-touch technologies are not true multi-user capable. At the same time, camera-based multi-touch tabletops of this kind have considerable potential for co-located collaborative work, given that the user identification problem can be solved. To achieve this, we propose a process model that transforms multi-touch data, in the form of touch points coordinates, into multi-user multi-touch data—separating blobs into their respective users—using an identity classifier (Figure 3). We define an identity classifier as a function to that maps unidentified finger blobs to its respective owners using different kinds of inputs.

![Fig. 3. Identity classification process for recognizing multiple users in a single-user multi-touch tabletop.](image)

In the following section, we suggest several methods of implementing identity classifiers, most of which will be heuristic-based in nature.

2.5 Identity Classifiers

**Distance-based:** Perhaps the most straightforward identity classification heuristic is based on calculating distances between blobs and then connecting them according to the minimum distances. The intuition is that finger blobs belonging to a single hand are typically clustered close together, and two hands belonging to a single user are correspondingly clustered close together. According Epps et al. [4], users use their index finger or spread hand for about 80% of the time while interacting with a multi-touch tabletop. This behavior could be exploited to further enhance the performance of the classifier. Of course, this simple heuristic breaks down in the face of complex cooperative interactions between users, such as crossing arms or performing very large-scale gestures.

**Camera-based:** A powerful hardware-based classification heuristics rely on augmenting the camera-based tabletop with additional cameras—typically placed above or below (in this case, seeing
through the surface to the participants hands above) the tabletop—and performing image processing on the live video feed. One possible implementation is to add an infrared camera above the surface and an infrared light source with different wavelength than that used in the acrylic of the FTIR tabletop, allowing the camera to track the movement of the users. Once rough coordinates of user hands are acquired, the classifier calculates the distance between blobs and these hands, assigning them to the right owner.

Dohse et al. [3] present a similar hardware setup based on a visible spectrum camera mounted above a FTIR tabletop that tracks hands using skin color segmentation techniques. This way, both the robustness of the multi-touch functionality can be increased, as well as user identities be tracked and used to differentiate touch points. The approach has the added capability of being able to detect hover gestures that are executed without actually touching the tabletop surface.

**Territory-based:** A software-based identification heuristic is to divide the surface of the tabletop into discrete territories [13], and to regard any interaction occurring in a territory as belonging to the owner of that territory. This relies on social protocols enforced not by the system but by the collaborating users themselves, something which may not be sufficient in many situations [10]. Furthermore, the strict and unflexible division of the visual space detracts from the multi-user experience and may decrease the efficiency of the collaboration.

### 3 Interaction Workspaces

In this section, we propose a novel software-based identity classifier called interaction workspaces that supports fine-grained access control of the visual space by assigning movable and resizeable regions to individual users. We first give a motivating scenario and then describe the details of the technique.

#### 3.1 Scenario

Suppose Pete is working as part of a team redesigning the bus routes for Greater Lafayette, Indiana. John and Mary, Pete’s co-workers, are familiar with the area and are already working collaboratively on a multi-touch tabletop to design the bus routes. Pete arrives, and proceeds to place his palm on an unoccupied part of the tabletop space for two seconds. In response, the system brings up a window to register his name. After registering himself, Pete defines his region of space—his interaction workspace—by using two fingers and dragging across the tabletop to specify the borders of the workspace. Once the workspace is instantiated, a simple tap on his name tag associates the interaction workspace with his identity.

At this point, the interaction workspace is now owned Pete and any interaction inside its borders will be classified as belonging to him. However, Pete is also now responsible for physically protecting his workspace in order not to come into conflict with John and Mary.

In the middle of process, Pete makes a mistake and wants to undo his recent changes. By using the workspace toolbox for accessing the workspace history, the system will revert his work to any specified previous state. Then, after finishing work in his workspace, John needs Pete’s help to improve his route, so John releases his workspace so that it is free. Pete can now destroy or release his current workspace and move to where John was working. By tapping on the released workspace, the system will ask Pete about his identity and then allow him to acquire the workspace to continue the collaborative editing.

#### 3.2 Basic Concept

An interaction workspace is a well-defined subarea of a visual display—its extents indicated using visual borders—that is considered to be owned by a particular user (Figure 1). Access to the workspace is assumed to be enforced by participants using a social protocol, so all interaction within the boundaries of the workspace is regarded as belonging to the workspace owner. Workspaces can be moved and resized. Because an interaction workspace can essentially be regarded as a mutual exclusion lock of the visual space, different workspaces are not allowed to overlap since this may cause ambiguities.

Beyond move and resize operations, interaction workspaces can be created and destroyed. They can also be released and acquired, allowing for sharing among users (see Section 3.3). Furthermore, each workspace may have an additional tool interface with operations local to the workspace (Section 3.4), as well as may maintain an interaction history to support undo and redo operations (Section 3.5).

Interaction workspaces are not independent desktops—they are intended to be used on full-display groupware applications (see scenario). Furthermore, by virtue of being resizeable and movable, interaction workspaces are generally more flexible and provide more fine-grained control than many other territory-based mechanisms [13].

### 3.3 Workspace Management

**Creating** a new interaction workspace is the first step for a user to be able to collaborate in a multi-user environment (such as the bus route design scenario above). During this process, the user will be asked to authenticate himself and then indicate the position and extents of the new workspace. Analogously, **destroying** a workspace will remove it from the visual display (although it can be recovered again by a global undo operation, see Section 3.5). Beyond these basic management operations, users can also **release** a currently owned workspace to share it for others to use, and they can correspondingly **acquire** a workspace with no current owner.

Creation, authentication, acquisition, release, and destruction are all global operations that rely on a social protocol for enforcement. To augment this protocol and to decrease the burden on the participants, additional rules can be enforced on the visual display, such as limiting the number of workspaces that one user can own, as well as the maximum allowable size for a single workspace.

#### 3.4 Workspace Tool Interface

Most groupware applications that support more than just basic navigation tasks require an application-specific tool interface. For example, in our bus route design scenario, there should be tools for drawing and deleting routes, as well as adding bus stops. Instead of having an interface toolbox associated with each user, every interaction workspace has its own toolbox. This will help both the users and the system. On the user side, this allows for customizing the toolbox of a particular workspace to fit the current task (Figure 4). It also provides a natural interface to accessing the interaction history of a workspace for undo or redo operations (see Section 3.5).

![Fig. 4. Example of each interaction workspace containing a unique and independent tool interface.](image-url)
3.5 Interaction History

The main purpose of interaction workspaces is to improve coordination on multi-touch tabletops. However, sometimes even our software and social protocols may not be sufficient to avoid conflicts. For such situations, we propose to add interaction histories that support redo and undo operations. Because of the structure of our workspace mechanism, we propose a two-tiered history mechanism:

1. Global history: Global event log, such as for workspace creation, deletion, release, and acquire operations (see Section 3.3).
2. Workspace history: Interaction log for operations conducted inside individual workspaces (see Section 3.4).

In other words, the system stores a history of workspace management operations since startup, and each interaction workspace in turn keeps track of changes made to the objects within that workspace. Accessing these histories can be done using redo/undo buttons placed on the global display, for the global history, and on each workspace tool interface, for the workspace history.

A problem might arise in situations when objects within different interaction workspaces are interdependent. If an outcome of an action within a workspace affects the outcome of an action within another workspace, user interference might restrict certain changes within those workspaces. We suggest developing a global dependency graph, compromising of objects irrespective of their parent interaction workspaces, to warn users in case of interference between changes being made. A dependency check flags a warning anytime changes to inter-related objects within different workspaces are performed. This mechanism provides a robust technique to resolve dependency issues between objects and prevents user from violating each other’s work.

Furthermore, interaction histories in groupware systems are clearly susceptible to conflicts arising from interference between the actions of different users [1, 11]. Beyond the static dependency graph of domain objects described above, this will require maintaining a dynamic dependency graph for individual user operations in the interaction history. We anticipate exploring this issue further in our future work.

4 Conclusions and Future Work

We have proposed a general model to describe the user identification process for camera-based multi-touch tabletop displays. This model enables us to describe a wide variety of identification mechanisms in terms of an identity classifier that accepts input from sources such as overhead cameras, distances between touch points, and territories, and then utilizes this data to distinguish interactions made by different users collaborating on the tabletop. We have also proposed a straightforward classifier based on non-overlapping interaction workspaces that are owned by a particular user.

This is mostly a conceptual paper, and we anticipate implementing and evaluating many of the ideas put forth here in the future. For example, we are interested in replacing our “soft” social protocol for high-level access control of workspaces by a more strict hardware-based protocol based on image processing and an additional overhead camera, similar to the method employed by Dohse et al. [3]. We are also interested in studying conflict resolution and avoidance for our workspaces, particularly in the presence of conflicting histories.

References