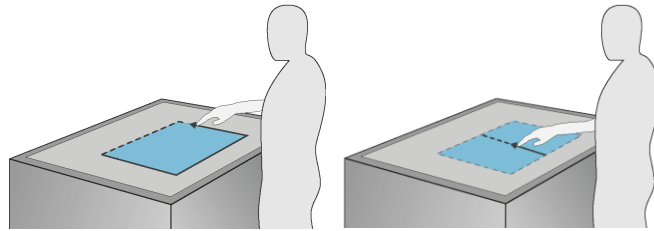

Drawing Shapes and Lines: Spawning Objects on Interactive Tabletops

**Tobias Hesselmann**

OFFIS Institute for IT
Oldenburg, Germany
tobias.hesselmann@offis.de

Volker Gollücke

University of Oldenburg
Oldenburg, Germany
volker.golluecke@uni-oldenburg.de

Benjamin Poppinga

OFFIS Institute for IT
Oldenburg, Germany
benjamin.poppinga@offis.de

Wilko Heuten

OFFIS Institute for IT
Oldenburg, Germany
wilko.heuten@offis.de

Susanne Boll

University of Oldenburg
Oldenburg, Germany
susanne.boll@uni-oldenburg.de

Abstract

In tabletop computing it is crucial to instantiate objects, such as documents or virtual containers, in an ergonomically convenient way for users. Particularly, objects need to be positioned within reach of users, need to be orientated properly, and need to be scaled appropriately for convenient interaction by touch. As the user's location at the device is usually unknown to the system, objects are typically spawned at a default position and with a default orientation and size in tabletop user interfaces. Thus, users typically need to manipulate objects after instantiation until they are properly aligned and scaled, which can be a cumbersome and time-consuming process. We designed two gesture-based interaction techniques to instantiate objects with a convenient orientation, size, and position, making further adjustments to these properties unnecessary. We describe the functionality of both techniques and discuss insights gathered during initial evaluations.

Author Keywords

Interactive Tabletops and Surfaces; Containers; Scaling; Orientation; Positioning

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces – Interaction Styles;

Introduction

The horizontal design of interactive tabletops typically enables users to approach and access such devices from different positions. For instance, in collaborative settings multiple users often interact with tabletop devices simultaneously, taking positions on different sides of the table (see Figure 1). This aspect has several implications for the design of user interfaces for such devices. For instance, user interface (UI) designers typically a) need to make sure that objects are positioned within arm's reach to enable users to properly interact with them, b) that they are orientated towards users for proper readability, and c) that they are scaled appropriately to prevent errors caused by imprecision induced by touch interactions [1].

As most available tabletop devices are unable to detect the positions of users at the system, objects typically cannot be appropriately scaled, orientated, and positioned by the system automatically. Instead, objects are typically spawned manually by users, e.g., by pressing a button [3], and are then instantiated by the system at a default size, orientation, and position. These properties need to be adjusted manually in follow-up steps - e.g. by performing touch gestures - until they conform to users' needs, which can be a cumbersome and time-consuming process.

To approach this problem, we contribute two gesture-based techniques to extract the proper position, orientation, and scale factor when spawning objects in tabletop systems. Both techniques rely on fairly simple touch interactions: With the *SHAPE-DRAWING* technique, users simply draw the outline of an object to instantiate it. With the *LINE-DRAWING* technique, users achieve the same effect by simply drawing a line on the surface.



Figure 1. Two users simultaneously interacting with an interactive tabletop (120 x 80 cm). Note that users stand at different positions at the table. Users interact with objects, which are orientated towards them, are placed in a reachable position, and are scaled appropriately to enable proper interaction by touch.

All necessary parameters, i.e., the rotation, position, and scaling of objects are computed from these gestures, making further manual adjustments of objects unnecessary. Using these techniques, contents can be instantiated in a properly aligned and scaled way from the start, saving time and effort for users.

We outline the functionality, advantages, and drawbacks of each technique in the following and provide qualitative results from initial studies that provide indications about the applicability of these approaches in practical settings. With our work, we hope to improve the design of tabletop user interfaces and seek to spark discussions regarding the handling of virtual objects in tabletop computing. As our work targets fundamental characteristics of interactive tabletop UIs, we believe it is relevant and inspiring for a large set of domains in surface computing.

Related Work

The concept of rotatable, scalable, and movable objects to support changing orientation, size, and position in tabletop computing is well known and broadly employed in research and the industry. Two of the most recent systems utilizing this approach are "G-nome Surfer" by Shaer et al. and "Cambiera" by Isenberg and Fisher, who utilize interactive tabletops for the co-located, collaborative analysis of genomic data [4] and the exploration of large text document collections [3]. Both systems instantiate objects at a default position and with a default size and orientation, and allow users to reposition, resize, and re-orientate these objects in follow-up steps. Different approaches to enable these manipulations have been described by research. Shen et al. propose to place handles on objects for manual reorientation, to automatically align objects towards the nearest edge, or to rotate objects towards users using a "magnet-like" technique [5]. Hancock et al. present a survey of five techniques for the precise and coordinated rotation and translation of objects [2]. And Wobbrock et al. propose user-defined gestures to manipulate visualized objects on interactive surfaces [6].

We conclude that freely positionable, rotatable and scalable objects are commonly used in tabletop applications and that various techniques exist to manipulate these parameters after the instantiation of objects. At the same time, we cannot identify techniques that allow a correct alignment and scaling of objects at the time of instantiation. In contrast, objects are typically launched at a fixed position and with a default orientation and size, which demands that users manipulate these objects until they are properly aligned and scaled. In the following, we describe two techniques to approach this problem.

Shape-Drawing

The functionality of the SHAPE-DRAWING (SD) technique is illustrated in Figure 2. To instantiate an object, users simply draw the outline of an object on the surface in a "finger-ink" way. When the finger is lifted from the surface, the drawn outline is mathematically approximated to a desired shape, e.g., a rectangle. The size and position of the object can then be computed from the size and the center of the approximated shape. The computation of the object's orientation is more complex and depends on the shape of the desired object. As shown in Figure 2, we can often find rectangular objects in surface computing applications, which are useful for displaying rectangular shaped media, such as documents and photos, or for migrating Desktop PC applications to interactive tabletops by embedding them into rectangular containers. For such objects, we compute the orientation by determining the direction vector between the starting point and the first vertex of the drawing. For circularly shaped objects, the size and position can be calculated using the same algorithm, while the object is orientated towards the starting point of the shape.

Evaluation

We evaluated the SHAPE-DRAWING technique as part of a qualitative study of a visual analytics application, in which SD was used to instantiate charts for data analysis. 5 participants (2 female, age 17-25, students from our university and trainees at our lab) took part in the study. We asked participants to solve a number of analysis tasks during the evaluation for which they needed to create objects (in this case: rectangular charts containing visual data) using the SD technique. Participants were allowed to try SD before the evaluation started.

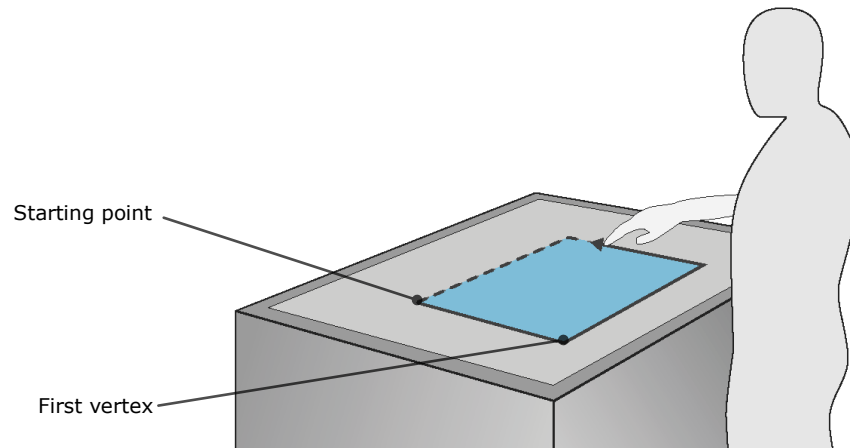


Figure 2. A user spawns a rectangular object by drawing its outline. The correct size and position are computationally derived from the drawn shape. The orientation is derived from the direction vector between the starting point and the first vertex of the drawing.

Results

We observed that all participants were instantly able to use the SD technique correctly and were able to spawn objects in the way expected. Also, verbal feedback from participants suggested that they found SD intuitive. Nevertheless, one participant found the technique cumbersome to use after a while, probably due to the physical demand induced by drawing the outlines of the charts, and suggested to only draw half of a rectangle (i.e., an "L" shape) instead of drawing the shape completely.

We found this an intriguing approach, as drawing half of a shape is still sufficient to extract all necessary parameters, i.e. the size, orientation and position of symmetrical objects, while both the physical effort and time needed to draw shapes could be largely reduced.

Line-Drawing

Inspired by the ideas brought up in the SD study, we sought to further minimize the physical demand and time needed to instantiate objects. Our efforts resulted in the LINE-DRAWING (LD) technique illustrated in Figure 3, which is described in the following.

If only one type of object (e.g., a rectangle) is used in an application and if its aspect ratio is fixed, a single line is sufficient to instantiate correctly positioned, orientated and scaled objects on the screen. To instantiate objects, users simply tap on the screen in front of them, then draw a line pointing away from them. The object is then spawned at the center point of the line, with its height corresponding to the length of the line drawn. As the aspect ratio of the object is known, its width can be calculated accordingly. By computing the direction vector between the starting point and the end point of the line, we can derive the orientation of the object easily.

Evaluation

We evaluated the LINE-DRAWING technique as part of a collaborative visual analytics application, in which LD was used to instantiate containers that contained the visualizations of the application. We invited 20 participants (5 female, age $M=25.5$) and split them into groups of two persons each. All of them were students at our university, and none of them had taken part in the SHAPE-DRAWING study.

Participants were asked to instantiate containers in order to solve simple analysis tasks. As we were interested in the immediate usability of the LD technique, we did not explain how to create containers in the system but asked participants to find out themselves.

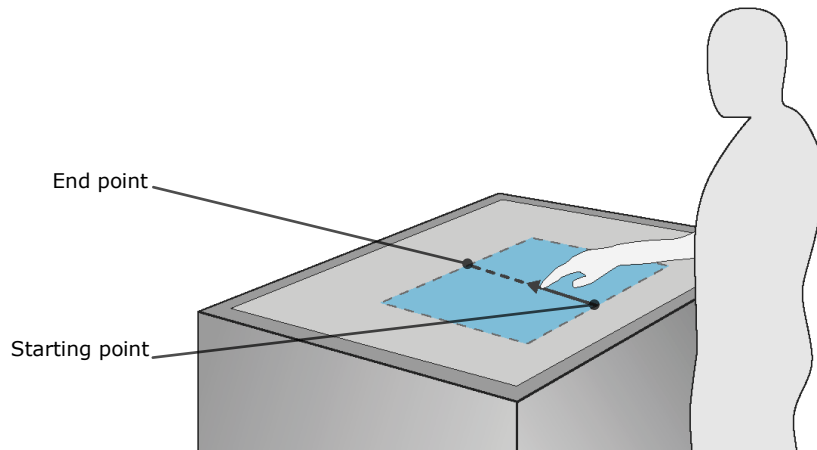


Figure 3. A user creates a new object by tapping the screen in front of her, then drawing a line directed away from her position. The position, size and orientation of the object are computationally derived from the vector of the drawn line.

Results

Participants tried different techniques to open containers at first, such as pressing their whole hand on the surface, drawing arbitrary shapes, performing single- and double-clicks, and drawing horizontal, vertical, and diagonal lines on the surface of the system. Nevertheless, after a short experimentation phase, all participants figured out how to open containers using the LD technique.

However, for some of them, the correlation between the length of the drawn line and the size of the resulting object was unclear at first. Although we displayed the line drawn on the screen visually, some participants did not recognize it as a line, but thought they created objects by performing a stroke gesture. Consequently, they created too small or too large objects at first.

Regarding the concept of deriving the object's orientation from the direction vector, we observed an interesting behavior of participants. We expected that participants would draw the line towards them, as the corresponding finger movement corresponds to a deictic gesture expressing "put the object here, right in front of me". But contrary to our expectations, many participants opened containers by drawing a line pointing away from them instead. We believe that this effect might be caused by the fact that it is physically more convenient to start a finger movement at a relatively low distance to one's body.

Conclusions and Future Work

Comparing both presented techniques, we can conclude that both the SHAPE-DRAWING and the LINE-DRAWING technique can be effectively used to instantiate appropriately orientated, scaled, and positioned objects on interactive tabletops. Both techniques were well received by participants in our evaluations, and both our observations and feedback by most users suggests a sufficient intuitiveness of both techniques.

A particular strength of the SD technique is that it can support different types of objects, such as circular, rectangular, or even asymmetrical ones. Further, the aspect ratio of objects does not need to be fixed – objects can be created in any shape users desire. As a drawback, we found indication that the physical demand induced by the SD technique could cause discomfort to users. Thus, we believe that an interesting direction for future research is to seek for methods attenuating this effect, e.g., by spawning objects using a bimanual "stamping" technique, in which the form of objects is derived from the posture of users' hands. For instance, users could form an "o" with their hands to create

round objects in the system, which is likely to speed up the instantiation of objects significantly.

The LD technique, while only able to create objects with a fixed aspect ratio thus far, induces less physical demand than SD and allows users to create objects much faster, as less finger movement is necessary to instantiate objects. Also, the correct orientation of objects can be elegantly derived from the line drawn, as users intuitively draw lines pointing away from their position in the correct angle. Regarding the intuitiveness of LD, our evaluation showed that it might be feasible to provide explanatory visual cues to achieve a steeper learning curve and to increase the affordance and the understanding of the technique. We implemented such a method in later implementations and displayed an animated finger that spawns objects in the system. Similarly to SD, it could be also interesting to experiment with bimanual input using the LD technique. For instance, one finger of each hand could be used to indicate opposing corners or sides of a rectangle, which is likely to induce less physical strain on users and increase performance.

Another direction for future research is to investigate how to appropriately distinguish between gestures for spawning objects and gestures for manipulating objects in tabletop applications. In our implementation, we made this distinction either based on the spatial context of a gesture (objects could only be launched in the interstitial space between objects) or by switching the system to a specific "spawn mode", which was triggered by touching one of the corners of the surface. The exploration of drawbacks and advantages of these or other approaches demand for more dedicated research in the future.

Probably the most manifest direction for future research is to conduct a comparative study of both approaches, gathering insights regarding user preference of the presented techniques. Also, we see potential in using the presented techniques not only to instantiate new objects, but also to reshape existing objects.

We believe our work is relevant for a wide range of applications and hope that it will contribute to improving the user experience of tabletop UIs in the future.

Acknowledgements.

We thank the student project groups TAP and TOAD for their valued help.

References

- [1] Benko, H., Wilson, A.D. and Baudisch, P. Precise selection techniques for multi-touch screens. In Proc. CHI '06, pp. 1263–1272. ACM, 2006
- [2] Hancock, M.S., Vernier, F.D., Wigdor, D., Carpendale, S., and Shen, C. Rotation and translation mechanisms for tabletop interaction. In Proc. Tabletop '06, pp. 79-88. IEEE, 2006.
- [3] Isenberg, P. and Fisher, D. Collaborative Brushing and Linking for Co-located Visual Analytics of Document Collections. In Proc. EuroVis '09, pp. 1031-1038. IEEE, 2009
- [4] Shaer, O., Kol, G., Strait, M., Fan, C., Grevet, C., and Elfenbein, S. G-nome Surfer: a Tabletop Interface for Collaborative Exploration of Genomic Data. In Proc. CHI '10, pp. 1427-1436. ACM, 2010
- [5] Shen, C., Vernier, F., Forlines, C., and Ringel, M. DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction. In Proc. CHI '04, ACM, 2004
- [6] Wobbrock, J.O., Morris, M.R. and Wilson, A.D. User-defined gestures for surface computing. In Proc. CHI '09, pp. 1083-1092. ACM, 2009