Improved Tabletop-Interaction using nestable circular Tangibles

Christopher Weber, Sebastian Petsch, Daniel Cernea, and Achim Ebert
Computer Graphics and HCI,
University of Kaiserslautern,
Germany
{name}@cs.uni-kl.de

Figure 1: *TangibleRings* in a map application, each controlling different information layers.

CHI'13, April 27 – May 2, 2013, Paris, France.
ACM 978-1-XXXX-XXXX-X/XX/XX.

Abstract
The multitouch functionality of tabletop computers is often augmented by the use of tangible objects that offer an intuitive and haptic alternative to interaction and manipulation. However, employing tangibles can also lead to less desirable effects, such as occlusion or lack of precision. In this paper we highlight the design and implementation of ring-like tangible objects: *TangibleRings*. They do not occlude the objects underneath them and also support the detection of touch events inside their perimeter. Additionally, multiple rings may be nested within one another in order to combine ring functionality or produce more complex filters.

Author Keywords
Tabletop; combinable tangibles; memory-enhanced tangibles; occlusion management; markers; virtual lenses.

ACM Classification Keywords
H.5.2. [Information Interfaces and Presentation (e.g. HCI)]: Input devices and strategies, Interaction styles

General Terms
Design; Human factors.

Introduction

In recent years, the multitouch functionality of tabletop computers has been augmented by the use of tangible objects that can be detected when positioned on or close to the surface of the tabletop. While these tangibles have the advantage of providing the user with distinct physical objects that he can use to interact with the virtual space of the tabletop, there are also some known issues. One such problem is the occlusion generated by opaque or translucent tangibles positioned on the tabletop, which inhibits a clear detection of the underlying objects and the precise positioning of the tangibles. Furthermore, it is not possible to interact with objects occluded (hidden) by the tangibles.

In this paper we propose an alternate solution that tackles the above mentioned issues and adds both, flexibility and functionality to the tangible-enhanced interfaces. We propose a solution called *TangibleRings*, that consists of a number of circular tangibles that support filtering inside their radii, quite similar to the concept of "virtual lenses" from the visualization field, e.g. Magic Lenses [2], GeoLenses [11], or tangible views for information visualization [9]. As our tangible rings are hollow, users can still see the virtual objects that are displayed on the tabletop in the center of the tangible. This aids in improving the precision of positioning or selecting elements with the rings.

Furthermore, the rings are designed as small controllers that rest on the tabletop visualization. As such, the rings not only offer an enhanced view, but also an additional layer of information or a particular filter depending on the application. They also allow users to manipulate the filtered information by performing touch operations inside the rings and to store information through special gestures. Besides in-ring touch operations, users can access various application-specific commands that the ring supports by means of a menu that is strategically positioned along the outer rim of the rings. The selection of an item in the menu is achieved by turning the ring clockwise or anti-clockwise, similarly to a knob. All this functionality is supported by a ring-shaped marker that is attached to each tangible and that has been design specifically for the *TangibleRings* to allow precise and unique identification of each ring positioned on the tabletop.

Similarly to stackable tangibles [1], *TangibleRings* can be combined to offer more flexibility to the user and the ability of executing logical operations with views and filters. For this, the diameters of the rings are slightly different, allowing them to be positioned inside one another. In this case, the combined rings act like the overlapping knobs on old radio devices, allowing the user to still turn each of the individual rings separately.

In the following section we will highlight related research in the field of tangibles and multitouch. Additionally, we will describe the design process and the functionality of our ring tangibles, as well as discuss how they overcome limitations of opaque and translucent tangibles and highlight their applicability in an example. We then engage in a discussion about advantages and potential issues and finish by offering our conclusions.

Related Work

There have been many approaches in different application areas [8]. Good overviews on tangible user interfaces can be found in the works of Ishii [4] or Underkoffler [10]. Most of the presented activities rely on the classical approach of tangibles as solid, block-like input devices that are identified by markers on their bottom (e.g. [9]). A drawback of dealing with these kinds of tangibles is that only a few approaches go further in utilizing the space on the tangible to show additional information or are able to offer touch surfaces on the tangible [6, 7, 12]. In the following we will present some of the works that tried to overcome and expand this clasical approach to tangibles.

SLAPbook [13] introduced SLAP widgets, transparent silicone peripheral devices that represent input devices like keyboards, sliders, or turning knobs. However, SLAP widgets are not stackable/combinable and they are also do not feature distinct screen space that can be used as individual information views. The Sensetable approach of Patten [5] electromagnetically tracks multiple wireless objects on a projection-based tabletop. The projector is installed above the table, allowing not only to project on the table but also augmenting the tangibles themselves. However, the tangibles are quite complex and the user's hand can easily occlude the projected information.

One of the most relevant approaches for our work is Lumino, developed by Baudisch et al. [1]. They used block-like tangibles filled with glass fiber bundles that can transport light through the tangible. Hereby, multiple blocks can be combined and the surface of a block can be used for simple touch events. The main application of Lumino lies in the possibility to build and track 3D tangible arrangements on a tabletop. There are certain similarities between our research and Lumino, like the combination of tangibles, the see-through functionality, and the possibility to use tangibles for touch inputs. Although, there are also significant differences such as the technical complexity of the Lumino blocks, which makes them expensive to produce. Further, the glass fiber bundles in the blocks do not offer a clear view on the underlying screen, which is a severe drawback for many applications. Also interaction possibilities are restricted to only a simple touch/no-touch detection on the blocks. Since our approach just reuses the original touch surface of the tabletop in the interior of our rings, we do not loose functionality like image quality or touch capabilities. In addition with an appropriate size and design of our tangible, this enables us not only to use the interior of our tangibles for touch, but also let us continue using multitouch gestures in this region.

"CapStones and ZebraWidgets" [3] extends Lumino's concept of structured transparency to capacitive tangibles consisting of multiple parts. This approach is only targeting devices with capacitive touch screens and cannot be used on tabletop computers that are based on diffuse illumination.

Our Approach

In order to address the requirements for combinability as well as highly functional tangibles, our method is based on circular-shaped tangible objects. These rings offer a variety of advantages that can be exploited to enhance interaction with multitouch applications. In particular, they minimize the number of occlusions on interactive surfaces. If conventional solid tangibles are used for interaction, important information is likely be hidden underneath them. Users then have to remove a tangible from the surface before they can identify an occluded item. Naturally, this problem worsens the more tangibles are involved in a scenario and the bigger their size becomes. Our circular tangible user interfaces (TUIs) offer an elegant way to address this issue. Although the tangibles we use are 8-12 cm in diameter, occlusion is not an issue. Since the center area of the tangible is spared, it occupies only a small fraction of the display. Furthermore,

Figure 2: Prototyping with real-world objects and final ring tangibles.

the coding scheme we describe in the next section enables us to build rings with narrower borders. In addition to minimize occlusions, the display area covered by a ring tangible can be used for further visualization and interaction and thus features new ways to manipulate and display data. Conventional tangibles basically "block" the display area they occupy, i.e., they reduce the area on which data can be visualized and that users can interact with. With the help of our approach, however, tangibles remain active parts of the surface. Gestures can be performed both inside and outside the tangible and can be processed in different ways. Outside-the-ring events can be used to interact with context menus that surround the tangible. Inside-the-ring events can trigger actions whose influence is limited to the inner radius of the tangible; for instance, the inside can be used to examine different levels of detail of an data set, i.e., it can act like a semantic lens that can be moved across the display surface. That way, users can investigate data sets without affecting other collaborators.

Another important aspect of our approach is the ability to combine tangibles by placing them inside each other. Placing one tangible inside another tangible results in a compound object which, in the simplest case, integrates the functionality of both rings, i.e., if ring $R1$ is associated with a particular filter $f1$ and ring $R2$ is linked to another filter $f2$, then the combination of both rings implements $f1\&f2$. In the application section we will discuss this feature within a map-based collaborative scenario.

A major benefit of tangibles is their ability to provide haptic feedback to users. Although interaction with multitouch surfaces is very intuitive, certain basic operations are not typically supported through multitouch gestures. For instance, pinch-and-zoom has become the almost universally accepted gesture for manipulating the size of objects on mobile devices. However, an equally intuitive and scalable gesture for defining orientations does not exist. Consequently, many map-based applications on mobile device do not even support rotation, whereas their desktop counterparts do. Due to their shape and the haptic feedback, our ring tangibles can be employed to easily carry out certain operations by using them like a control button or knob. For instance, they can be used for quickly defining numerical values, angles, and orientations. Many features of the example we discuss in the application section make use of this idea. The feature becomes especially useful being combined with the ability to nest multiple tangibles.

When designing the tangibles, a major requirement was that our rings must be easy to reproduce. Many methods propose complex setups in terms of hardware and materials which make it difficult to quickly build new tangibles. Furthermore, the more complex the design is, the less likely it can be reproduced by other research groups. Therefore, we avoid using any kind of electronics and exotic materials for building our tangibles.

Ring Hardware Prototyping

By choosing the form of a ring as the tangibles' shape for our approach, we had to start from scratch and prototype our own customized tangibles. As early prototypes, we used slightly modified, ring-like real-world objects, we found in supermarkets, home-improvement markets, or toy stores. These objects (see Figure 2) revealed rough requirements for the needed diameters and minimum rim width for placing our markers. Based on these findings, we went on to the final prototyping step and modeled the rings in Autodesk 3d Studio Max[1]. The resulting 3D data was plotted as a three dimensional solid object using a Dimension 1200 series 3D printer[2], see Figure 2. Instead

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

[2]http://www.dimensionprinting.com/

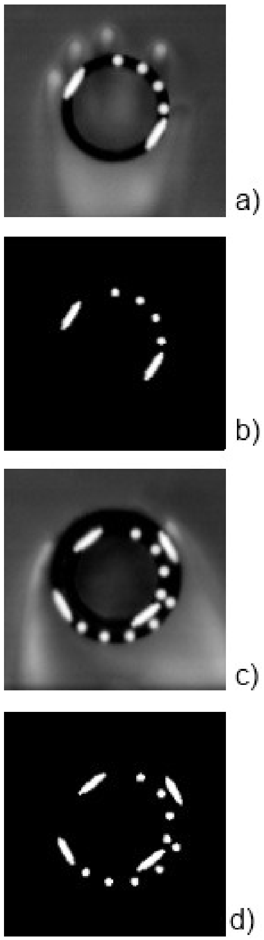

Figure 3: Ring detection using markers for single and concentric rings: (a),(c): raw image; (b),(d): image for blob detection (large blobs encode ring position and radius, small blobs encode id and orientation).

of modeling plain and smooth rings we decided to create a non-slip surface in order to improve the usability of the tangibles. We created three customized rings, which differ from each other in size and height and can be used standalone or stacked inside each other.

Tracking the TUIs
The tracking of our ring tangibles has to fulfill several requirements:

- Unique IDs for multiple tangibles
- Detectable orientation for each tangible
- Detectable position and radius of each tangible
- Interaction and gesture recognition inside the tangible
- Detection of concentric positioned tangibles

For the implementation of the *TangibleRings* concept we are using the Microsoft Pixelsense platform. It already comes with a working tangible recognition based on markers that would fulfill our requirements for unique id, position and orientation. The problem with these markers is, that they were designed for classical closed tangibles. They have a squared shape and require a minimal size of $2x2cm$ for a stable detection. Considering the relation in size and screen occlusion in contrast to our final marker approach, it is obvious that these tags are way to large for usage with *TangibleRings*, which rely on narrow ring rims to minimize occluded screen space on the tabletop. Thus we had to think of a new marker design that works for our approach. As the MS Pixelsense is very stable in recognition of fingers and gives us access to the raw image of the built-in IR-detectors, we decided to keep using the finger detection of the MS Pixelsense framework and use OpenCV [3] for interpretation of the raw-image and detection of our own markers.

Coding the ID
The basic idea for our markers is to keep them as simple as possible. A simple marker is most likely stable to detect and would give us the possibility to use smaller sizes. Furthermore, we do not need to code complex information inside the marker. Therefore our idea was to use a unary coding for the ID along the ring utilizing small blobs of maximal reflective white color, where the number of points along the ring encodes the ID of the ring. The geometric form is not of importance since we use the center of gravity of the detected blobs as position of the blob. In the end we used circles, but every kind of shape with an interior center of gravity would work well. To uniquely detect the rings, we detect circular shapes on the tabletop and white blobs representing ring marker. After this we combine these two findings.

Coding the orientation
The recognized markers are then assigned to the rings. For this assignment we use the center of gravity of the blobs as position of the marker. Then we compute the distance of each marker to the center of each ring and assign it to the ring with the best matching radius i.e. the difference between the radius of the ring and the distance of the blob to the center of the ring has to be within a given range defined by a threshold. Due to the maximal reflective white color of the ID-blobs we are able to eliminate influences by fingers and hands of the users, shadows or bad lightning conditions by filtering of the raw image. This is illustrated in Figure 3 where a) and c) show the raw image while b) and d) show the thresholded

[3] http://www.opencv.org/

image for the blob detection. The noise and hands of the users can be filtered out completely, making the ID detection very stable. What remains for the design of the ID-markers is the coding of the orientation of the rings. Let C be the center of the ring. We use one of the blobs B as indicator and compute the angle γ of the vector $\vec{CB}$ to the x-axis via the scalar product. This way we do not need a special type of blob for the detection of the orientation (e.g. by different color). This keeps the coding simple, robust, and more failsafe.

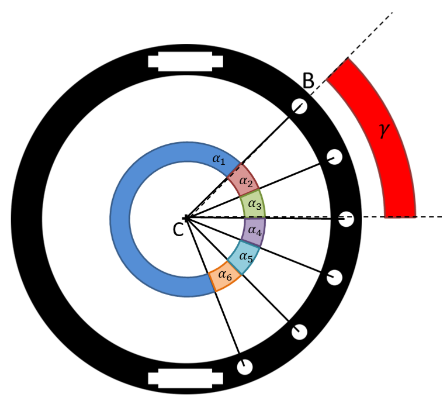

Figure 4: Computation of ring orientation γ: to identify blob B as marker for orientation, we compute the angles α_i between the markers on the ring and use the marker with the largest α in mathematical positive sense. The orientation γ is finally given as the angle between $\vec{CB}$ and the x-axis.

For B we need a unique identifiable blob on the ring. Therefore, we decided to position the blobs non-equidistantly on the ring and then use the 'first' blob in mathematical positive sense i.e. the blob with the largest gap in counter-clockwise direction. It can be fast determined by computing the angles between the blobs (α_i in Figure 4) on the ring and choosing the blob with the biggest angle to its next neighbor in mathematical positive sense. Figure 4 shows the principle of determining the orientation in this way.

Circle detection
To implement the coding as proposed, we need to know the center and radius of each ring. During development we first tried a classical circle detection based on Hough transformation. For better stability and performance we switched to a marker based circle detection method.

Pure marker-based circle detection
The method we developed is a pure marker-based circle detection: Here, we add a second type of marker to the tangible. The markers for the circle detection are about double the size than the markers for ID-coding and thus easy to separate from the the ID-markers by the detected blob size. We used size and not color since this allows a more robust detection. This also has the advantage that when thresholding the image for the blob detection, the upper threshold parameter is not needed anymore as the blobs are white. Due to the markers having maximal white color intensity values, the detection is very robust against changes of lightning conditions and delivers good results over a wide range of environmental conditions without the need of parameter tweaking. The two large circle-markers for each ring are positioned face to face on the ring. These two points on the rings are enough to determine the center and the radius of the ring and offer a quick and stable detection of a single ring. Although, for the detection of multiple (and later also concentric) rings, we need some more information to identify the correct pairs of circle blobs. Fortunately, this information can be gathered from the existing ID-markers. After the blob detection we have a list of 'large' blobs for the circle

detection and 'small' blobs for the ID-detection.

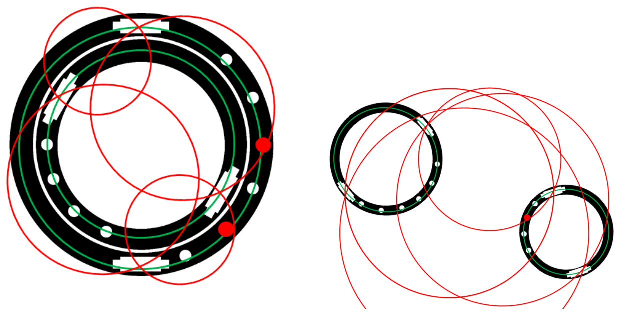

Figure 5: The figure shows the detection of the correct pairs of ring markers. The green rings are the correct rings, the red rings are candidates that are eliminated due to having not enough matching id markers.

To identify the correct pairs of circle blobs, we take every pair of large blobs as candidate for a circle and verify the decision by counting the number of ID-marker blobs that were assigned to the circle. If this number is larger than a threshold, then the pair is verified as a circle. This yields satisfying results for the following reason: During the assignment, as mentioned above, the ID blobs are added to the 'best matching' circle. In the case of only one or two it is still possible that two ID blobs have a position that matches to a 'wrong' circle. Figure 5 shows this situation. The red markers can be assigned to a wrong ring candidate. It can be easily seen that it is very unlikely and in this situation even impossible to assign more than three ID markers to wrong circles.

Mathematically we need more than $(2 * n) - 2$ ID markers where n is the number of ring tangibles, since every ring candidate can cut every ring two times minus the two cuts through the ring markers. In practice, a minimum required amount of 5 to 6 ID markers worked well, even when using more than four *TangibleRings*.

This method is robust enough to scale with an arbitrary number of concentric rings, though we think that more than two or three concentric rings would not be very useful or user friendly.

Implementation details

In this section we highlight some details of the implementation that are not directly connected to the ring detection but are of importance during work with the system or to improve the stability of the system. One problem when working with objects on the tabletop is that sometimes contact of the object with the surface is misinterpreted as finger touch event by the system. To overcome this problem we implemented a distance criteria δ (approximately half the width of the used ring) and ignore touch events that lie in the area of $radius + / - \delta$ around the circle center in the area where the ring touches the surface. Note that this way we keep the touching capabilities inside the ring and directly outside the ring, e.g. for implementation of a menu around or inside the ring etc.

To further stabilize the detection, we implemented a history that takes care of adding and removing rings from the system. If during one frame a ring is no longer detected e.g. the user moves the ring, but also lifts it a little so that the camera of the tabletop loses focus of the markers, the ring is not removed immediately. Only if it is removed for several frames (e.g. 5-7) it is removed completely from the system. This prevents the system from 'flickering' in those situations. Since the size of a tangible does not change over time, we can use the knowledge about its radius and store it in a list together with its ID to improve stability. By a comparison of the

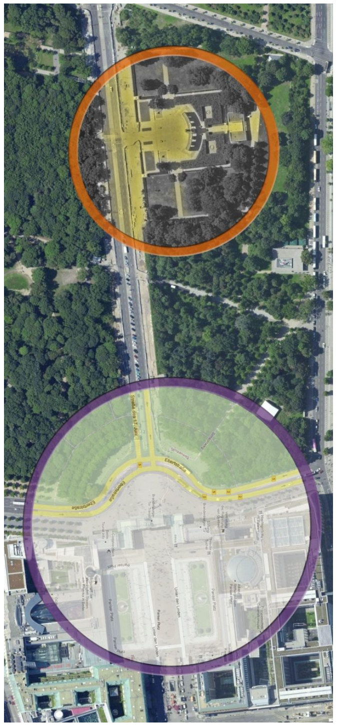

Figure 6: Two rings controlling different local parameters in the map application.

detected radii of the rings and the radius associated with the ID we can filter out false positive ring detection. Also changes on gesture recognition are necessary inside the ring. Here we wanted to take care that a gestures inside the ring can be interpreted relative to the orientation of the ring and 'up' is always 'up' in sense of the orientation of the ring and not the screen. In the end we offer two direction-values the application can use, in cases of gestures inside the ring. The global direction relative to the screen and the relative direction that depends on the orientation of the ring. We let the application decide which value to use.

Concentric rings

Another key aspect of our approach is the possibility to combine information related to the ring tangible by combining the tangibles themselves. The implementation using markers for circle detection works different and is more stable. Here concentric rings are recognized independently, yielding two separate circles instead of a single large one. So we only need to compare the distances of the found circle centers. If the distance between two circle centers is lower than the largest of the two associated radii, the circles are connected and each ring stores its partner(s) in a list sorted by radius. This way it is easy for each ring to know if it is an inner or outer ring and it can act according to the application e.g. only the outer ring should offer an ring menu on the outside like in Figure 7. Also false positive touch events inside the outer ring coming from the markers of the inner ring(s) should be ignored, so we should only react on touch events within the radius of the smallest ring in the list.

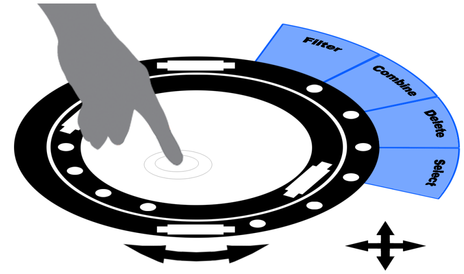

Figure 7: Ring features: translation, rotation(independent for multiple concentric rings), interior touch surface, ring menu outside for application specific function.

Application: Interactive Map

For demonstration purposes we have built multiple applications that use our *TangibleRings* concept. For this paper we have chosen a map-based scenario in which users can easily manage different information layers and filters.

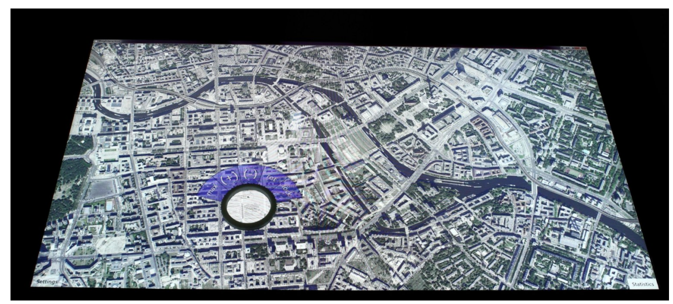

Figure 8: Overview over the map application.

This map application is a simple but powerful demo that presents some of the possibilities of the new ring concept (see Figure 8). It basically shows a satellite image of an area. Each ring used has a defined function and shows different information in its interior, e.g. classical map view, street names, heat map, traffic or population

Figure 9: Opacity, zoom factor, and traffic density plus heat map controlled by TangibleRings.

density, air pollution etc. Figure 6 shows the scheme for this application. Multiple rings show and control different information and parameters inside their local viewport. In the sense of collaborative work, multiple users can use multiple rings on the tabletop to control, combine and share these information. If rings are combined and nested into each other their information is also combined and for example street names and traffic density are shown together. Figure 9 gives some examples. It shows three rings representing information about street names, traffic density and a heat map. If the user in this example is interested in the combination of street names and traffic density or the heat map, he can visualize those attributes by combining the associated rings. The rings are combined and so is the information within, giving the users the chance to get new insights through exploration of the combined data. Although all users are working on the same screen, the screen space inside the rings offers an individual view for each user in which he can zoom in and out (see Figure 9) using a ring menu or a gesture inside the ring and explore the data set without interfering with other users if he needs a wider overview. By default, the rotation of the rings controls the opacity of the connected information layer (see Figure 9). If rings are combined the opacity of the combined rings can be controlled individually. Additionally, we can use the rotation of the rings in a different way. Using the ring menu, it is possible to change into a rotation mode that allows to rotate the screen inside the ring. This is useful when a user wants to read textual information which is upside down from his perspective and makes collaborative work easier if the users are positioned around the tabletop. Every user can explore the date in his own perspective.

A 'lock' function can fix the window shown inside a ring. This way a user who found some important information can select and save this part of the information and can 'give' it to other users in a intuitive way. Since the information is stored in the system and associated with the ID of the ring, the user can even remove the ring from the tabletop. If it returns to the tabletop, the information will be restored for further analysis.

Sometimes the local view inside a ring seems to be to small for an overview especially if the whole group of users need to discuss the information inside. For this case to share information with all users around the tabletop, we added a function that maximizes the information layer and applies the connected layer (or layers if multiple rings are used) to the full screen. This way it is possible to have a global overview over the data. In this mode a ring no longer acts as a lens but still as a controller and the opacity of each layer can still be regulated by rotation of the associated ring. In addition, rings can still be added to or removed from the maximized rings.

Conclusions

In this paper, we presented *TangibleRings*, a new sort of tangibles developed to overcome some of the major disadvantages in using tangible user interfaces on tabletops. By choosing ring-shaped objects, we are able to reduce screen occlusion and allow the usage of virtual lenses and multitouch interaction within their interior. In addition, this concept of combinable rings offers an intuitive and easy way to merge and manipulate the associated data, e.g. by adjusting settings or selecting items by turning the tangibles. One of the major benefits of our approach is the ability to reproduce this method in an easy and affordable manner, since even common items like the prototypes in Figure 2 would deliver adequate results. Furthermore, our tangibles are unpowered and do not need special or expensive technological elements.

With the presented map application we were able to show the concept of the *TangibleRings*. In future work this concept can easily be adapted to a variety of possible application areas such as architecture, urban planning, information visualization, or even medicine.

References

[1] Baudisch, P., Becker, T., and Rudeck, F. Lumino: tangible blocks for tabletop computers based on glass fiber bundles. In *Proc. CHI'10*, ACM Press (2010), 1165–1174.

[2] Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. Toolglass and magic lenses: the see-through interface. In *Proc. SIGGRAPH '93*, ACM Press (1993), 73–80.

[3] Chan, L., Müller, S., Roudaut, A., and Baudisch, P. Capstones and zebrawidgets: sensing stacks of building blocks, dials and sliders on capacitive touch screens. In *Proc. CHI '12*, ACM Press (2012), 2189–2192.

[4] Ishii, H. The tangible user interface and its evolution. *Commun. ACM 51*, 6 (June 2008), 32–36.

[5] Patten, J., Ishii, H., Hines, J., and Pangaro, G. Sensetable: a wireless object tracking platform for tangible user interfaces. In *Proc. CHI '01*, ACM Press (2001), 253–260.

[6] Rekimoto, J., Ullmer, B., and Oba, H. Datatiles: a modular platform for mixed physical and graphical interactions. In *Proc. CHI '01*, ACM Press (2001), 269–276.

[7] Schmalstieg, D., Encarnação, L. M., and Szalavári, Z. Using transparent props for interaction with the virtual table. In *Proc. I3D '99*, ACM Press (1999), 147–153.

[8] Shaer, O., and Hornecker, E. Tangible User Interfaces: Past, Present and Future Directions (preprint). *Found. Trends Hum.-Comput. Interact. 3*, 1-2 (2009), 1–137.

[9] Spindler, M., Tominski, C., Schumann, H., and Dachselt, R. Tangible views for information visualization. In *Proc. ITS '10*, ACM Press (2010), 157–166.

[10] Underkoffler, J., and Ishii, H. Illuminating light: an optical design tool with a luminous-tangible interface. In *Proc. CHI '98*, ACM Press (1998), 542–549.

[11] von Zadow, U., Daiber, F., Schning, J., and Krger, A. Globaldata: Multi-user interaction with geographic information systems on interactive surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces. ACM International Conference on Interactive Tabletops and Surfaces (ITS-2010)*, M. Haller and D. Wigdor, Eds., ACM (2010).

[12] Waldner, M., Hauber, J., Zauner, J., Haller, M., and Billinghurst, M. Tangible tiles: design and evaluation of a tangible user interface in a collaborative tabletop setup. In *Proc. OZCHI '06*, ACM Press (2006), 151–158.

[13] Weiss, M., Wagner, J., Jennings, R., Jansen, Y., Khoshabeh, R., Hollan, J. D., and Borchers, J. Slapbook: tangible widgets on multi-touch tables in groupware environments. In *Proc. TEI '09*, ACM Press (2009), 297–300.