

# HOUDINI: Introducing Object Tracking and Pen Recognition for LLP Tabletops

Adrian Hülsmann and Julian Maicher

University of Paderborn, Department of Computer Science, Germany  
adrian.marius.huelsmann@uni-paderborn.de,  
jmaicher@mail.uni-paderborn.de

**Abstract.** Tangible objects on a \tabletop offer a lot of different opportunities to interact with an application. Most of the current tabletops are built using optical tracking principles and especially LLP tabletops provide very good tracking results for touch input. In this paper we introduce HOUDINI as a method for LLP object tracking and pen recognition, which is based on three different sizes of touch points that help us to identify touch points belonging to fingers, objects and pens. As a result, the whole recognition process is performed at the level of touch information rather than frame by frame image analysis. This leads to a very efficient and reliable tracking, thus allowing the objects to be moved very fast without being lost.

**Keywords:** tabletop, interactive surface, object tracking, LLP, pen recognition.

## 1 Introduction

Tangible objects provide a natural way of interaction between tabletop and user. They offer opportunities which pure touch input does not offer, for example adding tactile feedback [20], allowing intuitive map navigation [15], a more precise adjustment of parameters [18] [6] [11], solving input conflicts [13] or improving awareness in co-located collaborative group settings [17].

Tabletops are commonly based upon optical tracking technologies. In the process, the tabletop surface is enriched with infrared (IF) light that gets reflected down into one or more IF-camera(s) as soon as fingers or objects hit the surface. Then it is further processed within the tracking software before the extracted touch information is sent to the application.

The predominant methods for optical tracking are *Diffused Illumination* (DI), where the IF-illumination takes places from below the tabletop surface, *Frustrated Total Internal Reflection* (FTIR)[5], in which IF-light is brought into the surface from the side where it is trapped (because of the equally named physical principle) until a touch of a finger allows it to reflect down into the camera(s), *Diffused Surface Illumination* (DSI)[2], which similar to FTIR uses IF-light coming from the side, but also requires a special Endlighten™ acrylic as surface material, and *Laser Light Plane* (LLP), where laser beams establish a plane of IF-light just above the tabletop surface that gets scattered at every touch point [16].

Each of those principles has its own advantages and disadvantages and raises special problems. Besides touch input, DI also allows object recognition through attached fiducial symbols that are easily made out of printed paper [9]. But for good tracking results, DI depends on an equally illuminated surface, which is not easy to achieve from below, so that this method sometimes suffers from false inputs. Whereas FTIR is not able to detect fiducial symbols, it is very good for the tracking of fast finger movements, because of the IF-lighting delivering a camera image that is rich in contrast. However, FTIR needs an acrylic surface to ensure the physical effect of total internal reflection. As a consequence, very big tabletops are hardly to achieve, since acrylic is not as stable in shape as glass and deflects with bigger dimensions. To counter this, the acrylic needs to be considerably thicker; this rapidly increases its price.

DSI also allows object recognition through fiducial symbols, but it needs a rather expensive Endlighten™ acrylic with the same downside concerning form stability mentioned above. Since in LLP tabletops the surface is not directly illuminated and the plane of IF-light only gets scattered during touch input, the camera image is very rich in contrast, thus enabling a fast and reliable tracking of fingers, even during fast movements.

In addition, LLP tabletops can be built with much bigger dimensions, because stable glass panes can be used, which are also significantly cheaper than acrylic. By the use of lasers, the illumination also becomes independent from the tabletop size, in contrast to DI, FTIR and DSI, where a bigger surface requires more or stronger IF-light emitters.

In summary, LLP tabletops enable a fast and reliable tracking of fingers and can be built with much bigger dimensions. But due to the missing possibility to track objects they are not very common. That is why in this paper we will introduce a first method for LLP object tracking and also pen recognition, which is based on touch recognition and therefore results in a reliable tracking, even when objects are manipulated very fast.

## 2 Related Work

reactIVision [9] is a computer vision framework using fiducial symbols for object tracking. The symbols consist of black and white patterns that are recognized by a camera and through image analysis these patterns are subsequently transferred into region adjacency graphs for object differentiation. Fiducials only work for tabletops based on DI and DSI since the marker patterns must be equally illuminated from the bottom.

ToyVision [12] also uses fiducials. In addition, the authors added features to augment the objects, giving the opportunity to manipulate objects while using them. To give an example, a button was added to the object which can be pressed to interact with the application.

There are other tracking approaches which use RFID tags to identify objects. In [14] this idea is combined with the camera image of the surface. Thus, giving the

opportunity to identify which shadow represents which object and being able to track the objects by simply tracking the shadows.

Similar to our approach, in [7] markers are placed onto acrylic disks to allow object recognition. Nevertheless, this approach uses conventional FTIR and DI tracking instead of LLP. Objects are recognized on the basis of image analysis, the markers are elastic and used for pressure sensing.

### 3 Basic Idea for LLP Object Tracking and Pen Recognition

LLP tabletops are characterized by the fact that every finger, but also every general object breaking the laser light plane, causes a reflection of IR-light down into the camera and thus creates a touch point (blob) in the tracking software.

With this paper, we introduce a solution to distinguish between blobs caused by *fingers, objects or pens*. Our solution is based on *three different sizes of blobs* in combination with *specific design constraints* that help us to identify blobs belonging to objects and pens.

The presented HOUDINI system is designed as TUIO [10] proxy between the widely spread tracking software *Community Core Vision (CCV)* [3] and an arbitrary tabletop application. Basically, from the set of blobs delivered by CCV, we at first exclude blobs belonging to objects and pens. While bypassing all other blobs to the application, we at second add messages for the identified object and pens, hence splitting the initial blob set into messages for fingers, objects and pens.

Before we describe the design decisions and the implementation in more detail, we at first introduce some fundamental requirements that have been the center of the development.

### 4 System Requirements and Goals

Our goal was to implement a system that meets several requirements, which we consider to be fundamental for tangible interaction on tabletop displays.

- The system should support tangibles of different sizes.
- Multiple objects on the tabletop surface should be reliably tracked at the same time.
- Tracking data should include translation, rotation and basic state information (pressing a button) of tangible objects.
- Object tracking should not limit the speed of processing touch input data of fast moving fingers
- The system should be able to support translucent tangibles that offer additional possibilities for application design and reduce the amount of surface occlusion.
- The tangibles themselves should be passive, i.e. without additional electronics or batteries. This also holds for the pens used as input devices on the tabletop surface.

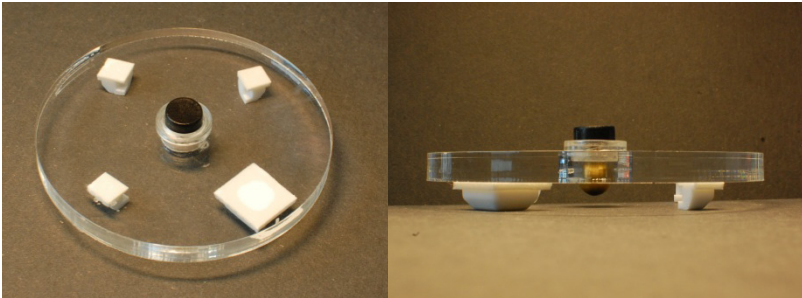
These are core requirements in the sense of general-purpose tangibles to be manipulated on the tabletop surface mostly by translation and rotation. The option to extend these basic tangibles with a physical button triggering a state change could be used in many application scenarios, for example to confirm a prior selection in the graphical user interface.

## 5 Design of Tangible Objects and Pens

Similar to other object recognition techniques, the fundamental idea of our approach is to attach markers at the bottom of objects forming individual *patterns* that can be identified by the system. But due to the characteristics of LLP, we do not rely on visual symbols and instead specify patterns as a set of touch points arranged in a certain way. This allows us to perform the object recognition at the level of touch information rather than analyzing the video stream from the camera frame by frame, as it is done by reacTIVision using fiducial markers [9].

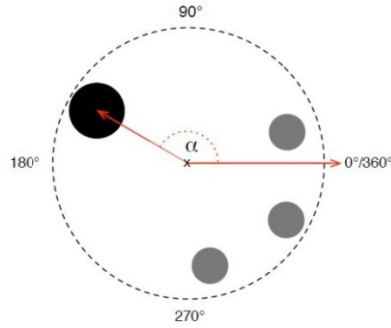
### 5.1 Object Pattern Definition and Derivation of Properties

Basically, our tangible objects consist of acrylic discs and attached beveled markers which reflect the infrared light plane down into the camera and hence create touch points (see Figure 1). As mentioned before, we use different blob sizes to distinguish between fingers, objects and pens. In this context, we specify an *object pattern* as a set of touch points consisting of several *small blobs* and exactly one *big blob*.



**Fig. 1.** Tangible object with attached markers for creating blobs of different sizes

During the manipulation of objects on a tabletop, the most significant properties to track are the object's *position* and *rotation angle*, which in our approach both must be derived from the object's pattern, e.g. a set of several small blobs and exactly one big blob. The position of the object is treated as the center of the *minimum covering circle* of all blobs which can be calculated in linear time [21]. The rotation angle  $\alpha$  of an object can be computed as shown in Figure 2. It is clearly determined by the vector from the object's position towards the position of the big blob and the x-axis of the underlying coordinate system.

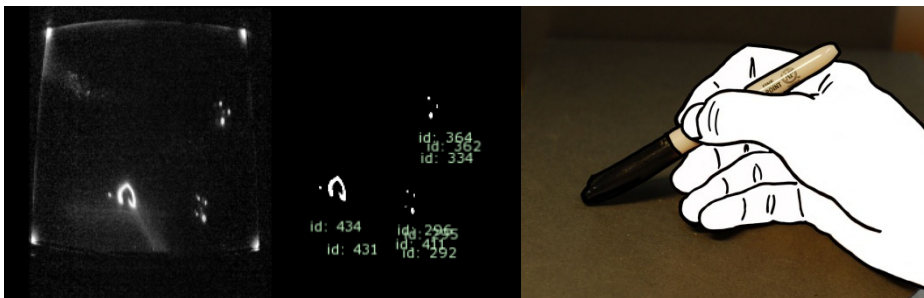


**Fig. 2.** Computation of the rotation angle. The rotation angle  $\alpha$  of an object is determined by the vector from the center of the minimum covering circle to the position of the big blob.

As shown in Figure 1, objects can also have a physical button in order to trigger events. This is very useful for situations where an object is for example used to invoke a menu in which rotation of the object is used for selecting one of the menu items and the physical button is used for confirmation [12]. When pressing the button an *activation blob* appears on the surface underneath the object. The activation blob represents the button and is stored in addition to the other blobs in the pattern. When it is recognized, the object is considered to be active. Then the marker ID of the object is negated to notify applications that the button has been pressed.

### 5.2 Pen Pattern Definition

The same principle of different blob sizes applies to the identification of pens that are used for tabletop interaction. Normally, when writing with a pen, the palm of the hand naturally rests on the writing surface. We turn this to our profit, by interpreting a *very big blob* as a resting palm. An occurring small blob located in a circular area with a certain distance to this very big blob is then interpreted as the tip of a pen (see Figure 3). As a result, the small blob’s marker ID is modified to notify applications that it belongs to a pen tip touching the surface.



**Fig. 3.** Palm triggers pen recognition (right). Blob patterns of two objects and one pen (left).

The advantage of this method is in the lightweight design without any additional electronic parts and the possibility to use standard office pens. Obviously, these should be capped or used with the end of the pen. Another benefit is that there is no need for a special writing surface overlay, which considerably reduces contrast of the visual display, for example used with the widely spread Anoto Digital Pen [1].

## 6 Implementation

HOUDINI is designed as TUIO [19] proxy between the widely spread tracking software *Community Core Vision* (CCV) [3] and an arbitrary tabletop application. As mentioned before, the system differentiates between *small*, *big* and *very big* blobs. We therefore use a modified version of CCV, which adds the size of the blobs to the data of touch points sent via the TUIO protocol using the */tuo/2Dblb* message profile.

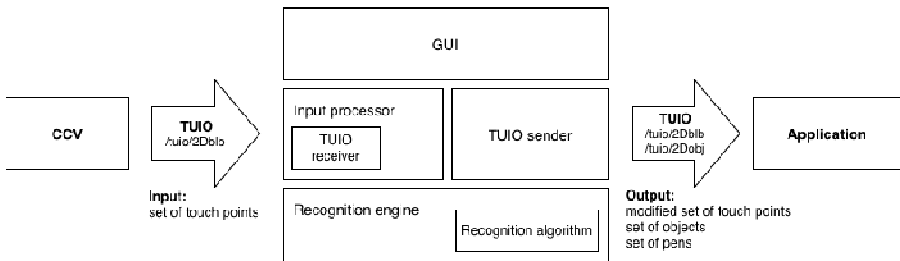


Fig. 4. Overview about the architecture of HOUDINI

The system listens for touch input on the port used by CCV and, if necessary, augments the input with object or pen-related information before sending it on a different port to an application using the */tuo/2Dblb* message profile for touch points belonging to no pattern (fingers) and the */tuo/2Dobj* message profile for identified objects or pens.

### 6.1 Architecture

HOUDINI is composed of four main building blocks which are shown in Figure 4. The *input processor* is responsible for receiving TUIO bundles, analyzing and, if necessary, filtering therein contained SET and ALIVE messages and afterwards forwarding them to the *TUIO sender* for dispatching. The *recognition engine* controls the recognition algorithm, supplies its input and processes the results of a recognition cycle. The *GUI* provides configuration options, a live view and pattern management functionalities for the user.

### 6.2 Recognition Algorithm

The recognition algorithm identifies objects by comparing the blobs on the surface to the patterns in the database and identifies pen input by looking for the separately specified pen pattern. The algorithm basically can be divided into two parts.

First, the *recognition part*, wherein the algorithm is working on all current blob messages delivered by CCV and tries to find blobs belonging to objects and pens. For already known objects or pens, the algorithm creates a new message with updated parameters which is sent to the application. For new objects and pens, the algorithm stores the IDs belonging to the blobs of the object or pen for later identification and creates a new object message containing position, rotation angle (for objects only) and ID of the object or pen. Afterwards this message is sent to the application.

Since objects have a unique pattern, whereas pens all share the same pattern, the system is not able to differentiate between pens after they lost contact to the surface. The following pseudocode illustrates the recognition process of objects in more detail.

```

for all bigBlob  $\in$  BigBlobs do
  for all pattern  $\in$  Patterns do
    collect blobs matching distance in pattern
    if more blobs found than needed then
      create subsets from the set of found blobs
    else if # of blobs equals # of blobs in pattern then
      put all found points into subset.
    end if
    for all s  $\in$  Subsets do
      compute minimum covering circle for s
      compute vector from center of circle towards
      big-Blob
      for all blob  $\in$  s do
        check whether angle of blobs matches angles
        in pattern
      end for
      if all blobs match position of pattern then
        pattern found! add to results.
      end if
    end for
  end for
end for

```

After receiving the blob messages from CCV, the algorithm collects all big blobs on the surface that have not been assigned to an object. For each of these big blobs the algorithm tries to find small blobs around it, which match the distance specified in one of the patterns. Since in this step only the distance towards the big blob is taken into account, there may be multiple points matching the specified distances. If this is the case, subsets are created, such that for each pattern point exactly one blob, which matches the distance, is taken into account. Hence, each subset consists of the big blob and one blob for each pattern point. As a next step, subsequently for each subset, if at least one is found, the minimum covering circle is computed. After that, the algorithm checks for each small blob if it is at the correct position. In order to do so, the vector from the center of the minimum covering circle towards the big blob is computed. This vector is then used to compute all angles for all other blobs and to compare them to the corresponding angles in the pattern. If both, the distance and the

angle matches, the blob is at the correct position. The vector from the center to the big blob also determines the rotation angle of the object. The IDs of the blobs which have been assigned to an object are stored for future reference. These blobs can be ignored by the algorithm when searching for new objects.

The second part of the algorithm is the *reassignment of missing touch points*. As written above, if an object is recognized, all touch points and their IDs are assigned to this object. That way, if a blob disappears for a short time and then reappears with a new ID, the algorithm is able to replace the missing ID with the new one. The benefit of trying to reassign blobs compared to directly removing the object and recognizing it later is that this method reduces object-added and object-removed events, thus provides the application with more stable objects.

### 6.3 Graphical User Interface

HOUDINI's graphical user interface allows an easy setup and on-the-fly configuration of the tracking environment. The four main components are shown in Figure 5.

1. The registered patterns are shown at the left side including a thumbnail containing the blobs which form the pattern. Patterns are color-coded in order to give a visual connection between patterns and actual recognized objects in the live preview. Patterns can be added and removed from the library easily e.g., when a pattern should not be tracked anymore.
2. The live preview is a miniature view of the whole tabletop. It visualizes all blobs and derived tangible objects and pens.
3. Configuration parameters are shown at the right side and allow the modification of thresholds between the different blob sizes.
4. The main functions are arranged in a menu bar at the top. The live view, object and pen recognition can be switched on and off and new patterns can be registered.

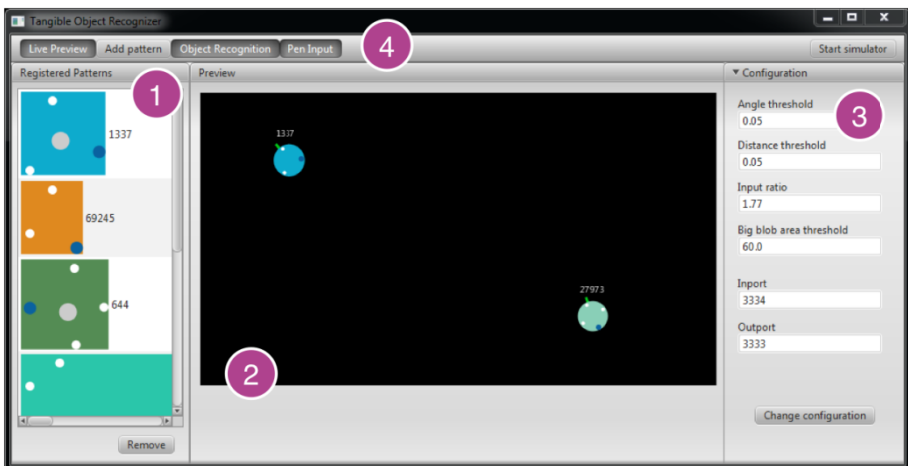


Fig. 5. HOUDINI's graphical user interface showing the main components



## 7 Conditions for Proper Tracking

At first, marker points attached to objects must be at a certain distance to each other in order to avoid melting of two blobs into one. Therefore, tangible objects must have a certain size, whereby bigger objects provide more space for the marker points. As a result, there is a clear link between the object size and the number of distinguishable patterns. An exact number of distinguishable patterns is difficult to estimate, but we successfully constructed ten different patterns using acrylic discs with a diameter of 80mm (with a tabletop dimension of 100cm x 160cm) that can be used at the same time without any problems and we see potential for a variety of more. At second, on LLP tabletops many marker points may lead to shadowing of blobs. We therefore recommend using at least one IF-laser per corner for proper results.

Concerning the user experience, we argue that a reliable tracking of less objects is more important than being able to differentiate between many objects at the expense of tracking quality. The real benefit of HOUDINI is that it is based on pure blob data, which are very stable and reliable in LLP tabletop environments, because of high contrast between touched and untouched surface areas. HOUDINI builds upon that feature by implementing object and pen recognition on TUIO messages only and with no need for complex image analysis or processing raw image data.

Therefore, HOUDINI is most suitable for applications scenarios with fewer objects, ranging from using tangibles to control parameters of music mixing interfaces [4], to physically navigate across a map [22], activate widgets and menus [8] or playing games which make use of fast moving objects like air hockey.

## 8 Conclusion and Future Work

We have presented HOUDINI as a TUIO proxy between the tracking software CCV and an arbitrary tabletop application and therefore implemented a solution for an object and pen recognition technique that is designed for LLP tabletops. It also completely fulfills the initially mentioned requirements.

As a result, it supports translucent tangibles of different sizes, whereby multiple objects and pens are reliably tracked at the same time without limiting the speed of processing touch input data. Object messages include translation, rotation and basic state information (pressing a button), while pens only contain information about the position and whether they touch the surface or not. Further, multiple pens can be tracked at the same time, but they cannot be distinguished from each other, because they all share the same blob pattern. This also results from the last requirement which asks for a solution without additional electronics or batteries. Instead, standard office pens can be used, which contribute to a better user experience and seamless integration of scenarios using tabletop interaction in combination with writing on real paper.

By focusing on patterns based on touch points and only using the TUIO messages for calculating the object properties, we were able to realize reliable tracking results, even during fast movements of the tangible objects.

Necessarily, our approach leads to a dependency between the object size and the amount of distinguishable patterns. We therefore do not propose our system for applications that intend to use many different tangibles.

Instead, we think HOUDINI is appropriate for the remaining majority of application scenarios, where only a few tangible objects are used, but which then can be moved very fast without being lost during the tracking.

## References

1. Anoto Digital Pen, <http://www.anoto.com>
2. Akechi, N., Mizumata, T., Sakamoto, R.: Hovering fingertips detection on diffused surface illumination. In: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ITS 2011, p. 1. ACM, New York (2011)
3. Community Core Vision, <http://ccv.nuigroup.com/>
4. Gelineck, S., Büchert, M., Andersen, J.: Towards a more flexible and creative music mixing interface. In: CHI 2013 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2013, pp. 733–738. ACM, New York (2013)
5. Han, J.Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, UIST 2005, pp. 115–118. ACM, New York (2005)
6. Hancock, M., Hilliges, O., Collins, C., Baur, D., Carpendale, S.: Exploring tangible and direct touch interfaces for manipulating 2d and 3d information on a digital table. In: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ITS 2009, pp. 77–84. ACM, New York (2009)
7. Hennecke, F., Berwein, F., Butz, A.: Optical pressure sensing for tangible user interfaces. In: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ITS 2011, pp. 45–48. ACM, New York (2011)
8. Jetter, H.-C., Gerken, J., Zöllner, M., Reiterer, H., Milic-Frayling, N.: Materializing the query with facet-streams: A hybrid surface for collaborative search on tabletops. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2011, pp. 3013–3022. ACM, New York (2011)
9. Kaltenbrunner, M., Bencina, R.: Reactivision: A computer-vision framework for table-based tangible interaction. In: Proceedings of the 1st International Conference on Tangible and Embedded Interaction, TEI 2007, pp. 69–74. ACM, New York (2007)
10. Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E.: Tuio: A protocol for tabletop tangible user interfaces. In: Proceedings of the 2nd Interactive Sonification Workshop (2005)
11. Lucchi, A., Jermann, P., Zufferey, G., Dillenbourg, P.: An empirical evaluation of touch and tangible interfaces for tabletop displays. In: Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction, TEI 2010, pp. 177–184. ACM, New York (2010)
12. Marco, J., Cerezo, E., Baldassarri, S.: Toyvision: A toolkit for prototyping tabletop tangible games. In: Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2012, pp. 71–80. ACM, New York (2012)
13. Olson, I.C., Atrash Leong, Z., Wilensky, U., Horn, M.S.: It's just a toolbar!: Using tangibles to help children manage conflict around a multi-touch tabletop. In: Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction, TEI 2011, pp. 29–36. ACM, New York (2011)

14. Olwal, A., Wilson, A.D.: Surfacefusion: Unobtrusive tracking of everyday objects in tangible user interfaces. In: Proceedings of Graphics Interface 2008, GI 2008, pp. 235–242. Canadian Information Processing Society, Toronto (2008)
15. Piovesana, M., Chen, Y.-J., Yu, N.-H., Wu, H.-T., Chan, L.-W., Hung, Y.-P.: Multi-display map touring with tangible widget. In: Proceedings of the International Conference on Multimedia, MM 2010, pp. 679–682. ACM, New York (2010)
16. Schning, J., Hook, J., Bartindale, T., Schmidt, D., Olivier, P., Echtler, F., Motamedi, N., Brandl, P., von Zadow, U.: Building interactive multi-touch surfaces. In: Müller-Tomfelde, C. (ed.) Tabletops. Human-Computer Interaction Series, pp. 27–49. Springer (2010)
17. Terrenghi, L., Kirk, D., Richter, H., Krämer, S., Hilliges, O., Butz, A.: Physical handles at the interactive surface: Exploring tangibility and its benefits. In: Proceedings of the Working Conference on Advanced Visual Interfaces, AVI 2008, pp. 138–145. ACM, New York (2008)
18. Tuddenham, P., Kirk, D., Izadi, S.: Graspables revisited: Multi-touch vs. tangible input for tabletop displays in acquisition and manipulation tasks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2010, pp. 2223–2232. ACM, New York (2010)
19. TUIO, <http://www.tuio.org>
20. Weiss, M., Wagner, J., Jansen, Y., Jennings, R., Khoshabeh, R., Hollan, J.D., Borchers, J.: Slap widgets: Bridging the gap between virtual and physical controls on tabletops. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2009, pp. 481–490. ACM, New York (2009)
21. Welzl, E.: Smallest enclosing disks (balls and ellipsoids). In: Maurer, H. (ed.) New Results and New Trends in Computer Science. LNCS, vol. 555, pp. 359–370. Springer, Heidelberg (1991)
22. Wu, A., Reilly, D., Tang, A., M.: Tangible navigation and object manipulation in virtual environments. In: Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction, TEI 2011, pp. 37–44. ACM, New York (2011)