# Combining multi-touch surfaces and tangible interaction towards a continuous interaction space

**Rafael Nunes**
LASIGE, Universidade de Lisboa
rlnunes@di.fc.ul.pt

**Carlos Duarte**
LASIGE, Universidade de Lisboa
cad@di.fc.ul.pt

## ABSTRACT
Multi-touch interaction scenarios are usually limited to one surface even when combined with tangibles. Traditional scenarios where people interact with physical objects on and above tables or other surfaces have failed to be fully translated into existing technologies, such as multi-touch set-ups, which don't support natural interactions by combining the surface and the area above it into one continuous interaction space.

We aim to build and explore a set-up that allows users to benefit from a continuous interaction space on and above the table with multi-touch and tangible support. We expect to find and solve problems that can arise in various scenarios, both individual and collaborative.

A set of different existing technologies will be integrated to monitor user interactions and an accompanying API will be developed and presented to serve as a tool for future development of applications that draw from this kind of set-up.

### Author Keywords
Tangible, Multi-touch, Continuous interaction space

### ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

### General Terms
Human Factors; Design; Measurement; Experimentation.

## INTRODUCTION
Multi-touch surfaces are emerging in an ever growing list of everyday scenarios. Traditionally, this type of set-up allows interaction with elements on the surface projection through touch input. This is a paradigm that has been studied in many works ([12], [16]) and it became clear that it could benefit from some augmentations that would add to the interaction experience, such as what is shown in [17], where a small room installation is proposed to explore a variety of interactions related to interactive displays and the space that they inhabit, and in [10] which proposes an interaction technique and hardware sensor for sensing handoffs that use physical touch above the table.


IUI 2014 Workshop: Interacting with Smart Objects, February 24, 2014, Haifa, Israel


But even these new scenarios aren't fully capable of supporting natural user behaviour with physical objects around a surface, such as object manipulation, exchange, moving and other actions, which can be present both on and above the surface. We wish to address the existing problem of combining digitally controlled interaction capabilities with real actions that are natural to the user in non-technological set-ups.

The area above the table surface can pave the way for new interactions when combined with the surface itself, functioning as a continuous interaction space. Gesture recognition adds natural user interaction above the surface, while touch recognition allows it on the surface. Physical objects also come natural to any user in everyday scenarios with traditional tables, and we wish to contribute towards a multi-touch augmented set-up that supports natural user interaction through touch, gestures and tangibles on and above the surface, by combining different technologies. To achieve this we have built a multi-touch table that allows interaction both on and above the surface with touch, gestures and tangibles.

An API is being developed to aid in the development of client applications for this type of set-up by abstracting different technologies into one form of input while providing a set of automatic tools for expected interactions and system behaviour.

In this paper we present our proposed set-up and the technology that was put to use towards building it. The proposed API is explained in detail with a list of events and properties that will help future development of client applications for this type of set-up. Finally we describe our expectations for future work on our set-up and API along with conclusions we have taken from this work.

## RELATED WORK
Extensive research has been done on similar set-ups, but it has mainly focused in either *on the surface* ([7], [9]) or *Above the surface* ([14], [13]) interactions. *On the surface* interactions are traditionally actions like selecting, grabbing, throwing, rotating and moving while *Above the surface* interactions are typically point and select and accessing areas not reachable by the user's direct touch [11].

In [11], the continuous interaction space is defined as being composed of the direct touch surface and the space above,

Figure 1. This allows new ways of interaction such as extended continuous gestures, which are described as gestures that start through direct touch on the surface and continue in the space above it. Mirrored gestures allow users to perform the same action via a gesture either directly on the surface or the space above it, which supports the user's natural behaviour by allowing one to choose either a gesture on or above the surface to interact with the digital content. It is also possible to pick and drop digital objects through physics-based interaction by grabbing a digital object on the surface and pulling it upwards. These gestures take advantage of the continuous interaction space by removing traditional restrictions and allowing both modalities to work together. However, this work does not support manipulation of actual physical objects in the continuous interaction space which is a necessary step towards natural user behaviour.

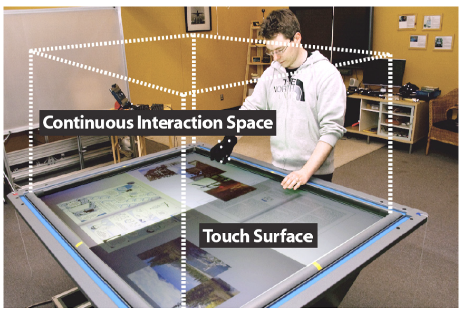


**Figure 1. Continuous interaction space, as depicted in** [11]

Although there is an extensive amount of research in multi-touch set-ups, few work has focused on user identification for this continuous interaction space. User identification enables new interaction possibilities by making the interface aware of which user is interacting with it. In [6] a proximity-aware multi-touch table is presented. Using 138 proximity sensors it detects a user's presence and location, determine body and arm locations, as well as distinguishing between right and left arms and map touch points to specific users and hands. In [15], a technique is proposed for user identification on interactive surfaces that enables users to access personal data on a shared surface, associate objects with their identity and customize appearance, content or functionality of the user interface. This type of research is relevant when considering collaborative scenarios, where multiple users interact with the same system at once. In [8] research was made on collaborative workspaces in which multiple users work on the same data set. Users would sit on special chairs that attribute different signals to each user, allowing the surface to know where it was being touched and by who. This was further explored in [10], where *electroTouch* was used to detect interactions between users above the surface by detecting current interruptions.

## SET-UP COMPONENTS

Different technologies allow our set-up to detect user interactions throughout the continuous interaction space. The following sections describe how these technologies were deployed and what they offer towards our continuous interaction space.

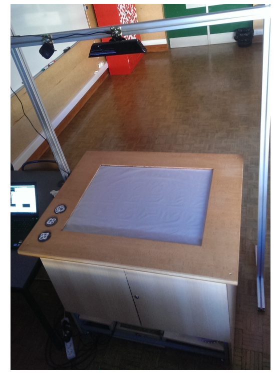

**Figure 2. Our multi-touch set-up. Two cameras are placed inside and above the surface**

### On the surface

We use the Frustrated Total Internal Reflection (FTIR) [1] method for touch interactions on the surface, which allows us to detect touch input through an array of infra-red light. A camera inside the set-up captures this light, which is then interpreted and translated into TUIO [5] protocol messages for client applications.

We decided to use *reacTIVision* [4] to achieve object tracking on the surface by tracking fiducial markers that are placed on physical objects. Since the previous camera inside our set-up had an infra-red filter, it could not detect visible light, which was a necessary feature for the fiducial tracking. As such, a second camera was introduced to capture the fiducial markers and translate them into TUIO [5] protocol messages for client applications.

### Above the surface

We used ThreeGear [3] to detect precise finger and hand tracking above the surface using one Kinect camera, as seen in Figure 2. This system allows detection of small gestures, like pinching and wrist movements instead of traditional arm detection. It is coupled with a corresponding API that allows writing of software applications based on its technology.

Tangible interactions above the table are inferred through gesture recognition. After an object is placed on the surface it is registered in the system as having hand A holding it, and can be tracked even after being lifted by tracking the hand holding it. Given that we cannot be always sure that the object is still being held by the same hand, we decided to complement this process by a camera above the surface that allows fiducial tracking. This allows confirmation of which hand is holding the object, instead of requiring this information to be inferred. Although the camera above removes the need to register tangibles on the surface before using them above, one cannot rely solely on fiducial tracking from above, since the marker can be easily lost due to other objects or hands moving over it. This is specially problematic in collaborative scenarios. Constant management of inferred and confirmed data from these sources is needed to achieve the best results, while removing limitations from the user, thus allowing natural interaction.

### Bridging the components

Having so many different technologies is quite cumbersome when considering communication between client applications and the various sources of input. We deployed a messaging framework, RabbitMQ [2], on our system architecture, Figure 3, to allow seamless communication between every technology. The API manages all the information from RabbitMQ and TUIO to deliver events to client applications. Any set-up component can use RabbitMQ to publish and subscribe to events, allowing seamless communication between different technologies and languages, making our system highly modular, since it is easy to add new components without making changes to previous configurations.

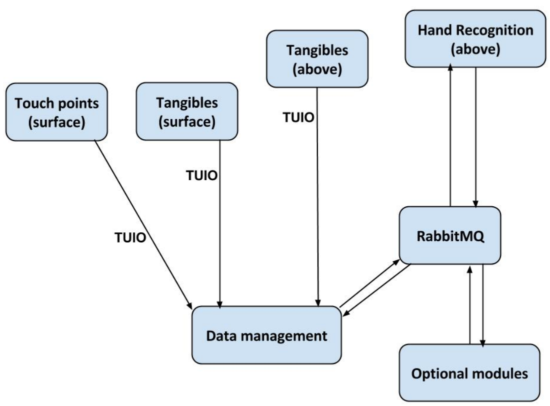


**Figure 3. System architecture**

### Solving Occlusion

Since fiducial tracking is supported by a camera that captures visible light, surface projections can get in the way and cause missing fiducial markers in a tracking scenario. To solve this problem, we searched for a background color that would allow easy and full fiducial tracking on the surface and applied it to an *object follower*. An *object follower* is a circle that surrounds the fiducial when it is tracked for the first time and constantly moves below it while maintaining itself above any projection. Optionally, tangibles can be rotated to increase or decrease the radius of *object followers*. This way we can be sure that the color below the fiducial will always be the desired color for tracking and significantly reduce the probability of miss-tracking a fiducial marker.

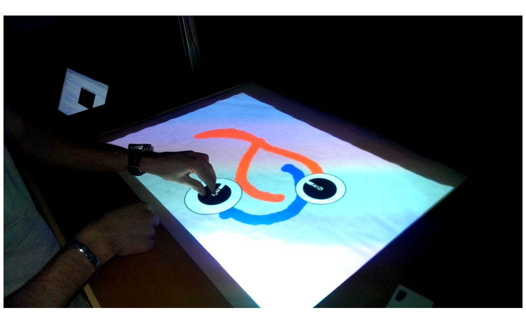

**Figure 4. API controlling *object followers* for tangibles**

In Figure 4 tangibles are being used as painting brushes with different colors. This scenario can rapidly cause color confusion in the projection, damaging the system's tracking abilities. Thanks to the *object follower* method, objects are easily tracked even when moving above painted zones.

## BUILDING AN API

### Event handling

Our system uses a considerable amount of technologies, so when considering the future of this work, we felt that an API would serve us greatly towards the development of applications. The API is fully coded in JavaScript to support development of HTML client applications. We have, so far, introduced some basic events and features to this API. These will be explained below:

| API Events | |
|---|---|
| Events triggered whenever an object **enters**, **moves** or **leaves** the surface | **object_added** **object_updated** **object_removed** |
| Events triggered whenever an object **enters**, **moves** or **leaves** an element that is expecting this event | **object.added** **object.updated** **object.removed** |
| Event triggered whenever a **touch** is tracked **inside** an element that is expecting this event | **touch.press** |
| Event triggered whenever a **touch** already being tracked **moves** inside an element that is expecting this event | **touch.update** |
| Event triggered whenever a **touch** is **no longer** tracked inside an element that is expecting this event | **touch.release** |

### Element Properties

Some properties can be easily attached to elements by adding the respective class to them. This way the API saves the user the trouble of making extra calculations. Next we detail a set of classes that can be added to elements and the properties they receive:

- **movable**

  A movable element is automatically moved by the API whenever a touch is registered inside it and movement follows. When the touch is released the element stays in the new position.

- **touchable**

  A touchable element receives events related to touch inside the area that corresponds to it, and can then respond to those events in whatever way the user wishes.

  Events received: **touch.press**; **touch.update**; **touch.release**

- **object-aware**

  An object-aware element receives events related to object tracking inside the area that corresponds to it and can then respond to those events in whatever way the user wishes.

  Events received: **object.added**; **object.updated**; **object.removed**

## FUTURE WORK

We want to find new ways to achieve user identification on our set-up, while making it easier to use it in collaborative scenarios, reducing confusion between multiple user interactions. Hand recognition above the table can be expanded to provide multiple hand pair recognition, thus allowing identification of user interactions by a corresponding set of hands.

Our API will be improved to add more events that we feel are necessary for application developers. The algorithm for constant swapping of information between inferred and confirmed data above the table is going to be improved to achieve better results and reduce errors as much as possible. There will be testing to gather data on development of applications for this type of set-up with our API.

A set of applications will also be developed to test our set-up and validate our findings towards natural user interactions within the continuous interaction space.

## CONCLUSIONS

We proposed an augmented multi-touch set-up that supports multi-touch and tangible interactions on and above the surface merging both into one continuous interaction space. By combining a set of technologies this set-up is capable of supporting natural user interactions with various modalities without interruptions when interacting with the surface and the area above.

An accompanying API was developed to support future development for this type of set-up by integrating all of these technologies into one single form of input. By providing a set of events and tools for client applications, the API allows easy and seamless development of applications.

## ACKNOWLEDGMENTS

This work was supported by FCT through funding of SiTECOSE project, ref. PTDC/EIA-EIA/117058/2010, and LaSIGE Strategic Project, ref. PEst-OE/EEI/UI0408/2014

## REFERENCES

1. Frustrated total internal reflection. `http://wiki.nuigroup.com/FTIR#Links:`, Dec 2013.
2. Rabbitmq. `http://www.rabbitmq.com`, Dec 2013.
3. Threegear systems. `http://www.threegear.com/technology.html`, Dec 2013.
4. A toolkit for tangible multi-touch surfaces. `http://reactivision.sourceforge.net/`, Dec 2013.
5. Tuio 1.1 protocol specification. `http://www.tuio.org/?specification`, Dec 2013.
6. Annett, M., Grossman, T., Wigdor, D., and Fitzmaurice, G. Medusa: a proximity-aware multi-touch tabletop. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST ’11, ACM (New York, NY, USA, 2011), 337–346.
7. Cao, X., Wilson, A., Balakrishnan, R., Hinckley, K., and Hudson, S. Shapetouch: Leveraging contact shape on interactive surfaces. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on* (2008), 129–136.
8. Dietz, P., and Leigh, D. Diamondtouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST ’01, ACM (New York, NY, USA, 2001), 219–226.
9. Epps, J., Lichman, S., and Wu, M. A study of hand shape use in tabletop gesture interaction. In *CHI ’06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’06, ACM (New York, NY, USA, 2006), 748–753.
10. Jun, L., Pinelle, D., Gutwin, C., and Subramanian, S. Improving digital handoff in shared tabletop workspaces. In *3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems (TABLETOP 2008)* (2008), 9–16.
11. Marquardt, N., Jota, R., Greenberg, S., and Jorge, J. A. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III*, INTERACT’11, Springer-Verlag (Berlin, Heidelberg, 2011), 461–476.
12. Park, S. Implementation of new gestures on the multi-touch table. In *ICT Convergence (ICTC), 2012 International Conference on* (2012), 168–169.
13. Parker, J., Mandryk, R., and Inkpen, K. Integrating point and touch for interaction with digital tabletop displays. *Computer Graphics and Applications, IEEE 26*, 5 (2006), 28–35.
14. Parker, J. K., Mandryk, R. L., and Inkpen, K. M. Tractorbeam: Seamless integration of local and remote pointing for tabletop displays. In *Proceedings of Graphics Interface 2005*, GI ’05, Canadian Human-Computer Communications Society (School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005), 33–40.
15. Schmidt, D., Chong, M. K., and Gellersen, H. Handsdown: hand-contour-based user identification for interactive surfaces. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI ’10, ACM (New York, NY, USA, 2010), 432–441.
16. Schöning, J., Hook, J., Bartindale, T., Schmidt, D., Oliver, P., Echtler, F., Motamedi, N., Brandl, P., and Zadow, U. Building interactive multi-touch surfaces. *Tabletops-Horizontal Interactive Displays* (2010), 27–49.
17. Wilson, A. D., and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST ’10, ACM (New York, NY, USA, 2010), 273–282.