Rich multi-touch input with passive tokens

Rafael Morales González

NNT : 2017SACLS309

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PARIS-SACLAY
PRÉPARÉE A L'UNIVERSITÉ PARIS-SUD

Ecole doctorale n°580
Sciences et Technologies de l'Information et de la Communication

Spécialité de doctorat: Informatique

par

RAFAEL MORALES GONZÁLEZ

Rich Multi-Touch Input with Passive Tokens

Thèse présentée et soutenue à Gif-sur-Yvette, le 9 Octobre 2017.

Composition du Jury :

Chantal Reynaud	Professeur, Université Paris-Sud	Présidente
Stéphane Conversy	Professeur, l'ENAC et Université de Toulouse	Rapporteur
Laurent Grisoni	Professeur, Université de Lille I	Rapporteur
Yvonne Jansen	Chargée de recherche, Université Pierre et Marie Curie	Examinateur
Stéphane Huot	Directeur de Recherche, Inria Lille	Examinateur
Emmanuel Pietriga	Directeur de Recherche, Inria Saclay	Directeur de thèse
Caroline Appert	Chargée de Recherche, Université Paris-Sud	Co-encadrant de thèse
Gilles Bailly	Chargé de Recherche, Université Pierre et Marie Curie	Co-encadrant de thèse

SYNTHÈSES

L'entrée multi-points offre un canal d'interaction très expressif pour les dispositifs équipés d'une technologie tactile multi-points. Cependant, alors que la taille du canal de communication est, en théorie, tr és grande, la plupart des systèmes n'en font, en pratique, qu'un usage très limité. Cet état de fait est probablement dû à la difficulté de gérer un grand nombre de gestes multi-points, et ce pour deux raisons principales: (1) les limites cognitives et motrices des humains, et (2) les difficultés techniques pour l'élaboration de systèmes de reconnaissance robustes. Cette thèse étudie une nouvelle technique d'entrée, TouchTokens, pour enrichir le vocabulaire de gestes multi-points, en se basant sur la position relative des points de contact et des objets (tokens) passifs. Un TouchToken est un "token" passif avec des encoches qui indiquent à l'utilisateur comment l'attraper. Ainsi, lorsque l'utilisateur tient le token tout en étant en contact avec la surface, lorsque les utilisateurs tiennent un token tout en étant en contact avec la surface, le système reconnaît le schéma de points de contact correspondant avec une grande robustesse.

Nous commençons par présenter le principe avec des tokens rigides de forme basique. L'algorithme de reconnaissance et la conception des tokens sont issus des conclusions d'une étude formative dans laquelle nous avons collecté et analysé des schémas de points de contact lorsque les utilisateurs tiennent des tokens de taille et de forme variable. Cette première étude montre que les utilisateurs ont des stratégies individuelles cohérentes, mais que ces stratégies dépendent de l'utilisateur. Ces conclusions nous ont mené à l'élaboration de tokens avec des encoches afin que les utilisateurs attrapent un même token toujours de la même façon. L'expérience que nous avons menée sur ce nouvel ensemble de tokens démontre que nous pouvons les reconnaître avec un niveau de robustesse supérieur à 95%. Nous discutons les rôles que peuvent jouer les TouchTokens dans les systèmes interactifs, et nous présentons un échantillon d'applications de démonstration : jeux, contrôle d'accès, contrôles, etc.

La conception initiale des TouchTokens ne supporte qu'un ensemble d'interactions se limitant au modèle à deux états de l'interaction directe. En d'autres termes, le système peut simplement capturer la position d'un token lorsque celui-ci est effectivement tenu par l'utilisateur. Dans un second projet, nous décrivons une technique de fabrication avec une découpeuse laser qui permet de faire des tokens flexibles que les utilisateurs peuvent, par exemple, courber ou compresser en plus de les faire glisser sur la surface. Nous augmentons notre reconnaisseur pour analyser les micro-mouvements des doigts pendant la manipulation du token afin de reconnaître ces manipulations. Ce nouveau reconnaisseur est basé sur l'analyse des micro-mouvements des doigts nous permet également de discriminer, lorsque l'utilisateur enlève ses doigts de la surface, le cas où il enlève le token de la surface, du cas où le token est resté sur la surface. Nous rapportons sur les expériences que nous avons menées pour déterminer la valeur des paramètres de nos différents reconnaisseurs, et tester leur robustesse. Nous obtenons des taux de reconnaissance supérieurs à 90% sur les données collectées.

Nous finissons cette thèse par la présentation de deux outils qui permettent de construire et reconnaître des tokens de forme arbitraire, TouchTokenBuilder and TouchTokenTracker. TouchTokenBuilder est une application logicielle qui permet de placer des encoches sur des contours vectoriels de forme arbitraire, et qui alerte en cas de conflit de reconnaissance entre tokens. TouchTokenBuilder produit deux fichiers en sortie: une description vectorielle des tokens pour leur construction, et une description numérique servant à leur reconnaissance. TouchTokenTracker est une librairie logicielle qui prend cette description numérique en entrée, et qui permet aux développeurs de traquer la géométrie (position, orientation et forme) des tokens au cours de leur manipulation sur la surface. Pour valider cette suite d'outils, nous utilisons une approche basée sur un mini-benchmark de trois ensembles de tokens de forme arbitraire. Nous rapportons ensuite sur une expérience dans laquelle nous demandons aux participants de manipuler ces tokens, afin de mesurer la capacité de TouchTokenTracker pour reconnaître l'identité de ces tokens, et capturer leur géométrie.

ACKNOWLEDGEMENT

I want to thank the people who have guided me along the Ph.D. path. The following list is certainly not exhaustive and I apologize to anyone who I did not mentioned.

In particular, I want to thank my supervisor Caroline Appert. I can say without doubt that I have been very fortunate and honoured to work with her. She demonstrated every day how research should be done. But more importantly, she always had an open door to help me in my troubles and ideas. She tolerated my going off on wild tangents (including changing my topic) and mistakes (I had many).

Gilles Bailly has played an equally important role, providing me with insightful comments. He never stopped encouraging and challenging me to always give my best. Without his ideas, enthusiasm, optimistic and a lot of chocolates in the team meetings, it would not have been the same.

I would also like to thank my supervisor Emmanuel Pietriga for thoroughly going over the papers and for all his insightful advice on how to improve everything like the demos, videos and of course my horrible presentations.

Now, at the end of my PhD, I am proud to say that it has been a true privilege to work with these amazing researchers. You taught me what means 'to do research' and how to appreciate it.

I will change in this paragraph to thank my family in Spanish. Nada de esto sería posible sin el soporte de mi familia. Papá, Don Isidro Morales Ovejero, te dedico la tesis doctoral. Te respeto por millones de cosas, pero sobre todo por tu capacidad de esfuerzo que he visto desde que tengo uso de razón. Sin darte cuenta has conseguido ser mi principal referencia en mi vida, tú, una persona sin estudios, me has enseñado más que cualquier libro. Mamá, Doña Teresa González Cuevas, mi profesora, mi madre, mi todo. Gracias, gracias y gracias por tu bondad, compresión e infinita paciencia desde que nací.

Que estas lineas sirvan para recordaros, que todo lo que he consigo en la vida os lo debo a vosotros dos. Os he echado muchísimo de menos.

I did not forget you, my spanish friends. I am far away from you, but it was a pleasure to talk to you from time to time with big messages and football matches. Thousands of stories that I missed but this line is for you. Many thanks for all the hours I spent with you.

Thanks to all my colleagues that accompanied my PhD at ILDA team and HCI Sorbonne!

And above all, I want to thank my wonderful girlfriend, María Otero Lago, she has given me support when I needed it most, distraction when I needed to get my head off work and love. I cannot imagine accomplishing this work without you, literally. I hope that you will continue to be my friend, my 'cosi' and partner for the rest of my life.

TABLE OF CONTENTS

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

In this dissertation, we explore a design space of gesture and tangible interaction for proposing a large input vocabulary that involves passive tokens and gestures on multi-touch surfaces. We focus on passive tokens because they are fast to build, easy-to-make and do not require any additional electronic components. Similarly, multi-touch technology has become increasingly accessible with regards to cost and diversity of products available.

With devices such as smartphones and tablets, multi-touch surfaces have become ubiquitous in our personal life. Larger setups such as tabletops and large vertical displays that support direct touch also become more widespread not only for research purpose [2] but also for public spaces [3]. Direct touch offers a great medium for gesturing on screen, thus providing a "natural" way of communicating between humans and computers. Interacting with gestures has retained attention for about 40 years [4, 5]. It provides many advantages including more expressiveness [6], cognitive benefits [7] and flexible manipulations for *e.g.*, selecting multiple items at once [8].

Multi-touch offers a very expressive input channel in theory, but the vocabulary of interaction is much more limited in practice. This is because of both human capabilities and system limitations. On the one hand, cognitive and motor resources limit the number of associations humans can memorize and the complexity of gestures they can perform. Also, fingers are not independent of each other, with some fingers moving inadvertently

to some degree when others move [9, 10]. This means that the number of degrees of freedom is actually less than what the technology can track (usually 10 contact points × 2D = 20 DoF). On the other hand, recognizing humans' gestures that involve a large number of muscles and joints from a sample of contact points on a tactile surface is difficult, the contact points providing only a limited picture of the actual gesture. Also, multi-touch devices suffer from two limitations: low precision and screen occlusion. Precision problems occur when selecting small targets directly with the relatively fat finger tip (*e.g.*, selecting a hypertext link within a series of hypertext links that are close to one another). Occlusion problems occur because of the directness of input as the object of interest can be displayed below the hand that interacts with the device (*e.g.*, dragging an icon from the top left corner to the bottom right corner for a right-handed user is challenging as her hand occludes the area where to drop the icon off).

In order to facilitate learning and provide more intuitiveness, HCI designers often try to use metaphors with the real-world as a design guideline. A more radical approach is to design *tangible interfaces* by turning virtual objects or controllers into physical objects that users can manipulate. Such interfaces have been shown as having several advantages such as facilitating learning [11], increasing performance [12], combining control and representation into a single physical device [13, 14], and improving the quality of collaboration [15]. But, despite all these advantages, tangible interaction is currently more challenging to implement than direct touch interaction. First, tangibles offer an even larger design space for building interactive systems with no guidelines or conventions that designers can rely on. Furthermore, whatever the technology considered, building and tracking tangible remains an effortful process in terms of fabrication (consistent circuit, good grasp, etc.) and software development (stability, friction with the screen, sensitivity to lighting conditions, proper processing software to achieve low enough, etc).

In this dissertation, we propose to address some issues of these two types of interaction, multi-touch gestures and tangibles, by combining them together with the concept of TOUCHTOKENS. A TOUCHTOKEN is a passive token that suggests a specific grip so that, when held on a multi-touch surface, the system can recognize the pattern associated with the token and then know what token the user is manipulating. By offering some tangibility to multi-touch gestures, TOUCHTOKENS save users from discovering and learning “invisible” gestures. They also allow interface designers to move objects from the virtual world into the physical world, giving an opportunity to reduce the difficulties

related to the manipulation of virtual objects with fingers. Besides, in comparison with other technologies for building tangible interfaces (such as making objects conductive or using image-based analyses to recognize passive objects), TOUCHTOKENS are low-cost and very easy to build. During my PhD, we have explored the idea of TOUCHTOKENS in depth. Our work can be presented along three main projects.

In the first project, we conducted empirical studies in order to evaluate the feasibility of the approach. We started by considering simple tokens that were differing in their shape and size to see whether users had a specific way of holding them. While we observed some consistency in users' grasps, the variability remained too high to design a robust recognition system. We thus designed a set of passive tokens featuring notches that constrain users' grasp. That way, the finger pattern associated with a specific token was very consistent. This approach allowed us to develop a recognition algorithm that classifies the corresponding finger patterns with a high level of accuracy. The recognition engine does even not require any training. We demonstrated with a set of application examples how our approach enables application designers with a flexible and efficient approach to develop a large variety of tangible interfaces.

In the second project, we aimed at increasing the expressive power of TouchTokens by introducing laser-cut lattice hinges in their construction, making them flexible. This design offers some flexibility to the tokens so that users can also *squeeze* or *bend* them in addition to sliding them on the surface. We improved our previous recognizer by analyzing the micro-movements of the fingers that hold the tokens in order to detect three new interactions: *squeeze* and *bend* but also whether a token is left on the surface or not. Bend events are especially useful for command triggers, while the squeezed state can be used for quasi-modal interactions (similarly to mouse drags).

In the third project, we wanted to increase the number and variety of TOUCHTOKENS that interface designers can consider in the applications that they develop. While we considered only six pre-defined tokens with basic shapes in our first approach, we now offer the ability for designers to design tokens that suit their specific needs. We introduced two tools that aim at facilitating the development of such custom-made TOUCHTOKENS, *TouchTokenBuilder* and *TouchTokenTracker*. With these tools, interface designers can create conflict-free sets of arbitrarily-shaped tokens using a simple direct-manipulation interface. Moreover, they can track the tokens' full geometry: location, orientation and shape.

1.1 Thesis Overview

This manuscript is divided into six chapters.

- **Chapter 2** provides a summary of previous works about gesture-based interaction and tangible interaction on multi-touch surfaces. We first define and discuss the evolution of gesture-based interaction. We then define and discuss the technologies of tangible interaction. Finally, we discuss the potential of combining gestures and tangibles as rich input channel for interactive surfaces.

- **Chapter 3** introduces the concept of TOUCHTOKENS. We present the design, the fabrication of TOUCHTOKENS as well as our recognizer algorithm. We also report on a formative study to collect touch patterns and a summative study to evaluate the recognizer accuracy. Finally, we present application domains that would benefit from TOUCHTOKENS and the limitations of our approach.

- **Chapter 4** introduces FLEXIBLE TOUCHTOKENS, a novel design of TOUCHTOKENS making them more expressive. We describe the principle of the new set of flexible tokens based on laser-cut lattice hinges and three novel interactions. We then report on a formative study to collect data, design our recognizer and evaluate its performance. We then present two applications that demonstrate the practical feasibility and the potential of FLEXIBLE TOUCHTOKENS. Finally, we discuss the limitations of this approach.

- **Chapter 5** introduces two tools that allows designers to build and recognize TOUCHTOKENS that feature arbitrary shapes. We first describe TOUCHTOKENBUILDER, which helps designers to build TOUCHTOKENS tangibles and to place notches in passive materials and TOUCHTOKENTRACKER, which provides additional information for tracking the tangibles. We then present some proof-of-concept token sets designed with TOUCHTOKENTRACKERand report on experiments to evaluate TOUCHTOKENTRACKER's recognition. Finally, we discuss the limitations of these tools.

- **Chapter 6** concludes with future directions for research. We first summarize our contributions. We then discuss some limitations of our approach and possible directions for future work.

CHAPTER 2

REVIEW OF LITERATURE AND TECHNOLOGY

This chapter provides a summary of previous works involving gestures and tangibles on multi-touch surfaces. We define what gesture-based interaction is, and its evolution from single to multiple contact points. We then define tangible interaction, and discuss its properties and the different technologies that support their implementation. Finally, we illustrate how combining gestures and tangibles can offer a rich input channel for interactive surfaces, listing different application domains where this combination has already been considered.

2.1 Gesture-based Interaction

Gesture-based interaction refers to interactive environments in which users communicate with the system using gestures. Stößel and Blessing [16] define a gesture, in the context of HCI, as *"a coordinated and intended movement of body parts to achieve communication. The information which it contains is specified by the configuration of body parts, the speed and direction of the movement and must be meaningful to its receptor."*

There is a large variety of gestures that can be used for HCI (e.g. stroke gestures [17], mid-air gestures [18], etc.). The choice of the type of gestures depends on both the design of the system and the tracking technology which it relies on. In our thesis, we focus on gestures performed on a tactile surface.

2.1.1 Single touch input

With single touch input technologies, the system is able to track only one contact point, typically corresponding to the tip of the stylus or the tip of the user's finger.

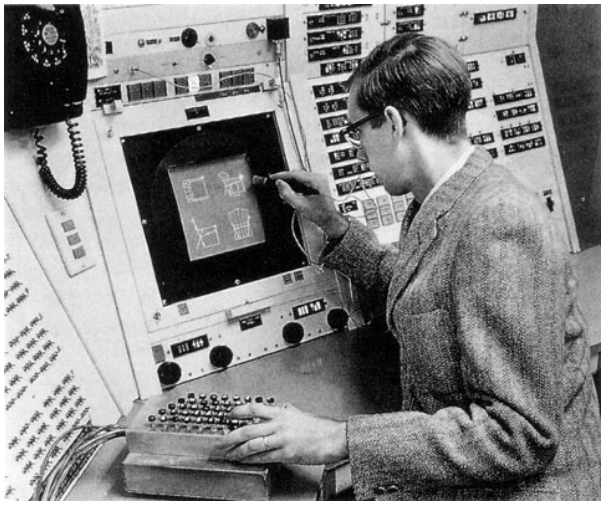


Figure 2.1: Sketchpad

Source: *http://history-computer.com/ModernComputer/Software/Sketchpad.html*

In 1963, Ivan Sutherland designed and developed one of the most important projects in HCI, Sketchpad [19], which can be seen as the pioneer of direct touch. Sketchpad is the first system that uses an optical stylus to interact directly with the virtual interface displayed on the screen. As Fig. 2.1 illustrates, the user could directly manipulate the graphical elements in a drawing editor. Ivan Sutherland's Sketchpad is usually considered as the most influential interactive system to the modern Graphical User interfaces (GUIs).

Since then, many efforts have been reported with respect to supporting stylus-based interaction. As an example, Moran et al. [20] presented Tivoli, which provides basic stylus-based scribbling and erasing interaction, allowing users to edit and organize materials on the Xerox LiveBoard. Wolf et al. presented We-Met (Window Environment Meeting Enhancement Tools) [21], which is a meeting room system that allows multiple users to use their stylus to annotate a shared document that is simultaneously displayed on each user's notepad. Stylus-based interaction offers a great precision as opposed to other input methods such as finger-based interaction [22]. This precision offers important

advantages for selecting targets [23], inking [24], annotating and taking notes [22], crossing goals [25], radial steering [26], and sketching [27].

Stylus-based interaction is also appropriate to the use of stroke commands. Stroke commands can be designed so that they resemble an activity or metaphor from the real world (*e.g.* performing a question mark gesture to invoke help), potentially facilitating discovery and learning. Appert and Zhai [28] have demonstrated the cognitive advantages of using stroke shortcuts in comparison with using keyboard shortcuts. In particular, they have showed that users were better at learning and recalling associations with gestures rather than with keystroke sequences. However, some studies have shown that there are some issues that must be taken in account when designing gesture-based interaction [17]. For example, Long et al. [29] found that while performing gestures with a stylus is faster and more iconic than textual commands, they can also cause usability issues because they are difficult to design and learn (stroke gestures are “invisible” to users and are thus difficult to discover and execute well at first). Bau and Mackay [30] tackled issues related to learning and memorization by proposing an appropriate visual feedforward technique when performing stroke gestures. Their technique, OctoPocus (Figure 2.2), helps novice users to perform a specific gesture, by revealing at first all available gestures (using colored trails that represent the next portion of available gestures), and progressively eliminating gesture candidates while users draw to keep only gestures that match user's partial input.

Gesture-based interfaces should not only support users during the discovery and learning phases, but also rely on a robust gesture recognition engine. Various approaches to stroke gesture recognition have been studied including heuristic recognizers [31], Hidden Markov Models [32], and statistical classification [33]. These approaches have significantly improved overall gesture recognition performance.

Most current tactile surfaces support input using the user's finger tip instead of or in complement to a stylus. Interfaces for touch heavily rely on point-and-tap interaction paradigm augmented with a few simple techniques such as double tap for *e.g.*, opening a document, or press-and-hold for *e.g.*, opening a menu. Despite a handful of innovative techniques such as Ta-tap and Ta-ta-tap [34], that use the time intervals and distances between consecutive taps to expand the touch input vocabulary, point-and-tap interaction is basic. It provides a limited expressivity, and it causes usability issues such as the fat finger problem [35], which causes ambiguity for selection [36], or the occlusion problem

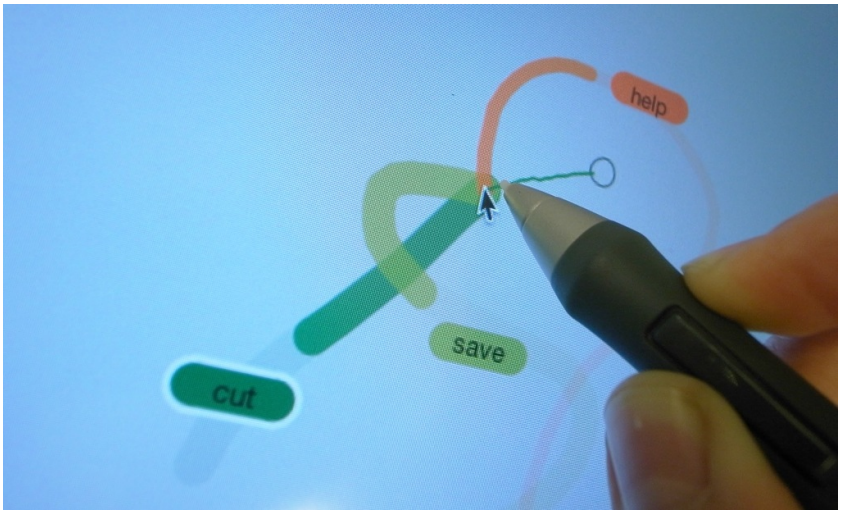


Figure 2.2: OctoPocus helps user to learn, execute and remember gesture commands

Source: *http://www.olivierbau.com/octopocus.php*

due to the hand that is used for touching the screen [37].

Some projects overcome these limitations. For example, one solution for *the fat finger problem* is to make all the targets large enough, as suggested by Wigdor and Wixon [38]. However, this solution limits the number of objects that the screen can display at once. Selection point ambiguity have been studied by Vogel and Baudisch [23]. They showed that the input point location is offset from the target that users intended to hit. They suggested that this offset comes from the fact that users perceive an input point that is different than the one that is actually captured by the surface. Based on this observation, they proposed to adapt and correct the captured input point by this offset. Moscovich [39] proposed another solution by approximating a finger's contact area with a small selection region, instead of using a single coordinate point. This resolves the ambiguity regarding which object has been selected by the finger (Figure 2.3). Solutions for *occlusion issues* have been proposed in LucidTouch [40] or Rubbing-tapping techniques [41]. LucidTouch reduces occlusion by allowing the user to perform multi-touch gestures on the back of the device. However, this solution requires that users hold the device with both hands, which also make it not tractable to other tactile devices such as tabletops or wall displays. Rubbing-tapping techniques [41] are more scalable to any kind of tactile screen as they rely on a software solution for recognizing a bi-manual interaction: diagonal rubbing gesture to point and zoom with one hand, and tapping with the other hand.

Figure 2.3: Moscovich's strategy for resolving the ambiguity regarding which screen object the finger is touching by increasing a target's effective width

Source: *[39]*

Other approaches for augmenting the expressivity of touch interaction rely on additional sensors, or the use of cameras and signal processing techniques in order to capture more properties such as which finger has touched the screen [42], how strong is the pressure applied on screen [43], or which part of the finger hit the screen [44].

Finger identification relies on pattern recognition techniques coupled with advanced sensors. For example, Sugiura and Koseki [45] developed a Fingerprint User Interface (FUI) system based on a fingerprint scanner sensor. FUI not only enables to associate different fingers with different commands but it can also be used to recognize different users. Holz and Baudisch [46] proposed an alternative technology that relies on fiber optic plates material and a high-resolution camera to capture fingerprints. The extremities of these fibers are reflective, allowing part of the infrared light to be reflected towards the camera. Then, when the light arrives at the surface of a finger it generates less reflection, creating black areas. This generates sufficient contrast to identify a person's fingerprint.

Finger pressure refers to the force that is applied on screen. Projects such as SimPress [47] or FatThumb [48] approximate pressure to the size of the finger's contact area, and propose to assign different meanings to a soft tap and a hard tap. Other projects use extra pressure sensors on the surface, such as the Force Gestures prototype [49] that is equipped with force transducers on the side of the screen, making it able to measure the normal force applied on the surface. Benko et al. [50] take another approach and describe how it is possible to identify and classify the amount of pressure of the finger in contact on the interactive surface using muscle sensing.

Acoustic sensors can also be used to recognize a larger number of gestures. TapSense [44], illustrated in Figure 2.4, is a project that uses an acoustic sensor to discriminate which part of the finger (nail, knuckle, pad or tip) have been used to hit the interactive

surface. This technique can be used with different devices and contexts such as a tabletop or a portable smartphone.

Figure 2.4: TapSense: acoustic signal is used to discriminate which portion of the finger (nail, tip, knuckle or pad) hit the screen

Source: *[44]*

Adding expressivity to the point-and-tap paradigm can also be done by the use of sliding gestures, which carry information in their trajectory and dynamics. These gestures have been implemented in popular touchscreen devices for *e.g.*, making a pattern to unlock a mobile phone, or sliding along a list item to delete it. These gestures can offer en efficient alternative to point-and-tap interaction. For example, Moyle and Cockburn [51] proposed an optimized recognizer for navigational flick gestures, and showed that such gestures can be more efficient than back and forward button for web navigation. Jain and Balakrishnan [52] developed a drag gesture starting from the bezel of a touch screen to distinguish the initial position of the bezel among similar drag gestures. Li [53] proposed the use of gestures that have more complex shape. His system, Gesture Search, allows a user to quickly access various data items on a mobile phone by performing a sequence of letter-shaped gestures on the touchscreen.

Finger micro-movements, as opposed to large-amplitude movements, have also gained researchers' attention as they allow users to execute the gesture fast and in a small touch area, making them appropriate to small devices. Among systems that have investigated this interaction, Roudaut's MicroRolls [54] and Bonnet 's ThumbRock [55] deserve particular attention. The MicroRolls interaction technique exploits thumb micro-gestures as a mechanism to enrich input vocabulary. They found that the movement of rolling a

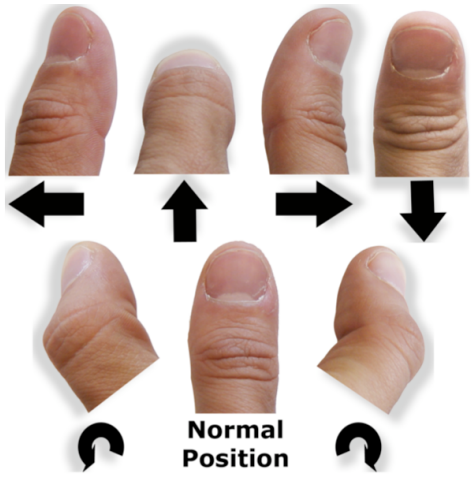


Figure 2.5: MicroRolls are micro gestures that consist in rolling the thumb tip in different directions

Source: *[54]*

thumb can be differentiated from drag, swipe and rubbing gestures. As shown in Figure 2.5, micro-gestures are accomplished by leaning the finger in six different orientation without the need to translate the finger tip on the screen. In the same spirit, Bonnet et al. [55] proposed the ThumbRock gesture, which consists in quickly rolling down and up the thumb tip on the screen. They proposed a recognition engine that is very accurate (96%) without the need for training or calibration, making the ThumbRock a good candidate to propose a selection gesture for direct input (like a mouse click does for indirect input).

All the projects mentioned above are about increasing the expressivity of touch input with a single contact point. Today, most tactile surfaces are actually capable of sensing multiple contact points, giving the opportunity to interact with the system using multiple fingers and even using both hands [38].

2.1.2 Multi-Touch input

Multi-touch input is supported by two main types of technology, each offering different opportunities for designing gesture sets: diffuse illumination and capacitive screens.

Diffuse illumination technology, mostly used for large displays like tabletop, has infrared sources mounted in the interior of the setup which emit infrared light towards the

surface. Objects or fingers on the surface reflect the infrared back into the table. Images of the surface can then be processed using image-based techniques for analyzing contact areas. Matsushita et al. [56] were one of the first researchers to use this technology in their HoloWall project. This system allowed a user to interact with a glass wall (the infrared camera was located behind) using physical objects, fingers, hands, and even their body.

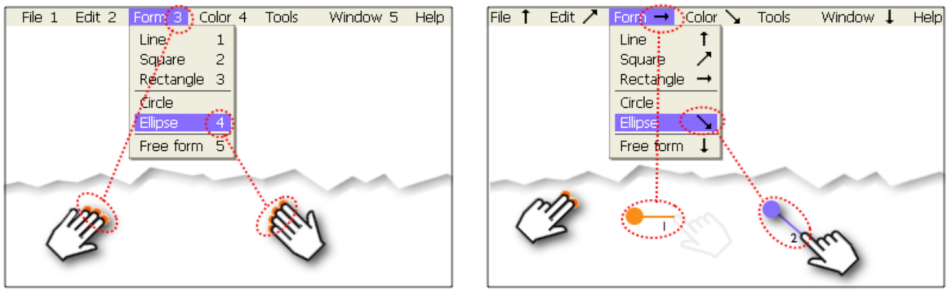


Figure 2.6: Two new interaction technique for Menus: Finger-Count (left) and Radial-Stroke (right).

Source: *[57]*

Using image-based analysis techniques, HCI researchers have proposed a large variety of innovative gesture-based interactions for this technology. For example, Benko et al. [47] presented up to five dual finger selection techniques, which allow the user to perform both cursor steering and selection with two fingers. Bailly et al. [57] focused on the number of contact points and the trajectory of the fingers on the surface to propose two ways of augmenting a menubar. The Finger-Count technique counts the number of fingers of each hand in contact with the surface, while the Radial-Stroke technique requires users to perform two short linear strokes with a certain orientation for selecting an item in the menu (Figure 2.6). Freeman and Balakrishnan [58] presented a system that allows the user to create, manipulate and reuse gestures to trigger commands on large multitouch interfaces. IdLense [59] also makes use of dynamic interface personalization. The system supports bimanual interaction where one hand is used to delimit a personal area and the other hand is used to manipulate this area's content.

Image-based analysis has also been used to identify users in projects such as Hands-Down [60] and ShapeTouch [61]. These projects aim at recognizing hand properties that can discriminate different users. While HandsDown analyzes the contact shape

of the hands on the surface, ShapeTouch captures the exact shapes of the finger tips by thresholding the input grayscale image obtained on the diffuse illumination surface. The same approach has been followed by Dang et al. [62] who developed an algorithm, based on finger orientation and relative distance between fingers, to map a set of finger contacts to a set of hands (Figure 2.7).

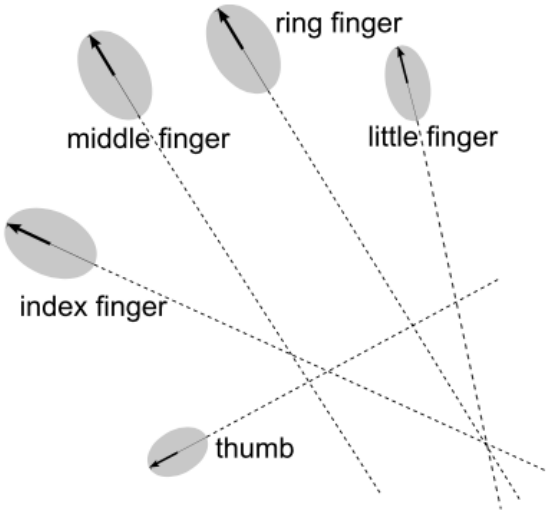


Figure 2.7: A finger arrangement is specific to a user

Source: *[62]*

Some researchers have focused on limiting physical occlusion on tabletops. For example, Khalilbeigi et al. [63] developed ObjectTop which displays an interactive halo in order to provide awarenesses of occluded items. Wigdor et al. [38] also addressed the occlusion problem by proposing interactions that combine whole shape gestures with finger input.

While diffuse illumination offers many advantages, especially because it captures the whole contact area, this technology has some drawbacks. In particular, it relies on bulky setups that are sensitive to lighting condition. By contrast, capacitive technology usually delivers only contact points but with more robustness. A capacitive surface detects a drop in capacitance when one or more fingers touch the screen. This decrease in capacitance is detected by sensors located around the edges of the screen, allowing the controller to determine the exact position of the multi-touch points. Capacitive touch screens can only be activated by the touch of human skin or of a capacitive stylus. Capacitive sensing

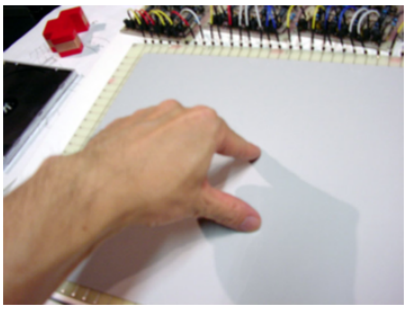


Figure 2.8: Tablet-size prototype.

Source: *[65]*

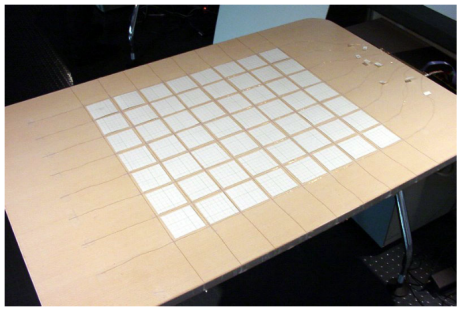


Figure 2.9: Tabletop-size prototype.

Source: *[65]*

is the most popular technology for multi-touch surfaces, and is widely used in portable devices like MP3 players, smartphones and tablets.

Capacitive sensing is not limited to small devices. For example, Dietz and Leigh proposed a procedure for creating a tabletop relying on capacitive sensing [64]. They developed a proof-of-concept prototype called DiamondTouch. The system has the ability to identify multiple users (up to four users) as well as it can detect multiple and simultaneous touches on the screen for supporting small group collaborations. It works by transmitting signals through antennas connected in the table. These signals are capacitively coupled through users and chairs to receivers, and can then be used to identify the user and the location related to a touch event. Rekimoto [65] also developed two prototypes, a table-size (Figure 2.8) and a tablet-size systems (Figure 2.9), that are based on capacitive sensing and mesh-shape antenna. The architecture is able to sense multiple human hand positions, finger gestures and even calculate the distance between the hand and the surface (hovering). As opposite to diffuse illumination technology, this technology does not suffer from issues related to lighting conditions.

With the exception of the prototypes mentioned above, capacitive touchscreen provides very limited information about the user's hand posture as it is limited to the number and position of touch points. Some research projects aim at improving the expressivity of interaction by relying only on this information. In particular, some projects make use of chord gestures for augmenting the vocabulary. The approach relies on contact points' relative position. For example, Lepinski et al. [66] investigated human capabilities for performing chording gestures for using multi-touch menus. The Arpège

project [67] have studied chord gestures, which are comfortable, and have proposed a contextual dynamic guide that help users to properly position their fingers for each chord command. Figure 2.10 illustrates the system with twelve chord commands. However, the technique requires per-user calibration to record the fingers' natural position when the hand rests in a comfortable posture. Wagner et al. [68] focused on three-finger chord gestures, and proposed a recognizer that analyzes the relative distances between touch points to discriminate chords that are performed with different sets of fingers.

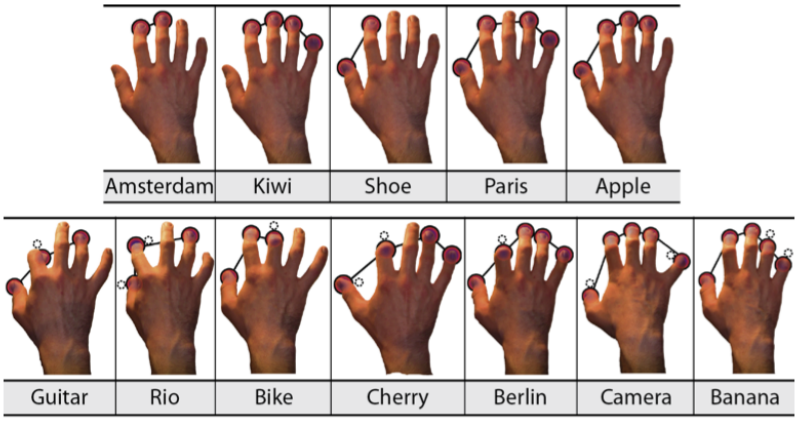


Figure 2.10: The Arpège system associates a chord with a command

Source: *[67]*

An interesting approach that is also based the contact points' relative position was presented by Harrison et al. [69] in TouchTools. They use machine learning to recognize up to seven touch patterns associated with seven hand postures on a capacitive screen (Figure 2.11). TouchTools' approach relies on the spatial finger configuration for each object that user would adopt if they were holding the corresponding physical object on the surface. Luo and Vogel [70] that gives some dynamics to a 2-finger chord with their pin-and-cross technique that allows users to select an object with one finger and cross a command line target with a second finger.

Multi-touch input also offers the possibility of interacting with two-hands. Some studies have shown that bimanual interaction techniques offer several benefits like reducing occlusion issues [71, 72], being less physically demanding and easy to learn for to text entry on tablet soft keyboards than unimanual text typing [73], being fast for completing task like moving, resizing and drawing an object [74], and offering a

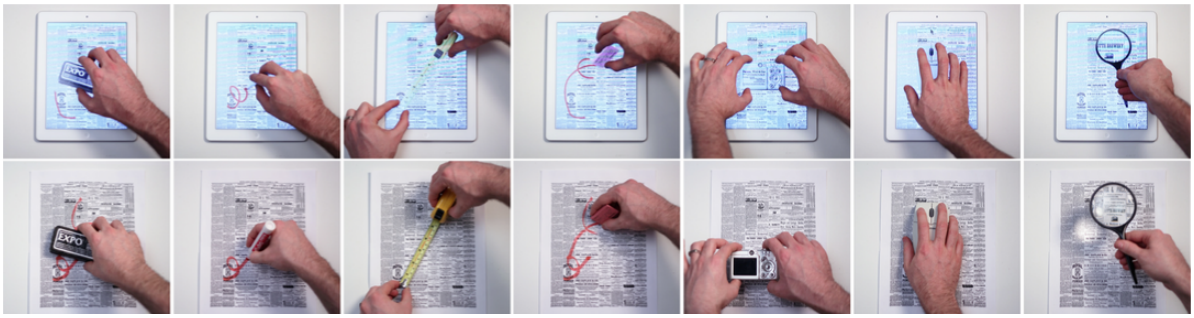


Figure 2.11: TouchTool: the 7 hand poses recognize in the system.

Source: *[69]*

good selection accuracy [47]. For example, Negulescu et al. [75] focused on improving precision selection, and designed two bimanual techniques based on pan-and-zoom. In their techniques, one hand is used to specify the center of zoom and the other hand is used to adjust the zoom factor. In a similar spirit, Kaser et al. presented FingerGlass [76], a bimanual technique for improving precision of graphical tasks on capacitive surfaces. With FingerGlass, users specify a circular region of interest using two finger of their non-dominant hand. This makes a magnifying glass pops up, allowing them to point with their dominant hand in this zoomed-in representation with a higher precision.

Bimanual interaction can also be used to define two simultaneous focal points for facilitating comparison. Butscher et al. [77] presented two navigation techniques for comparison tasks on tactile surfaces. With *PhysicLenses*, users can define two magnifying glasses at the same time. With *SpaceFold*, users can *fold* the virtual space along both the x- and y-axis to bring two areas close to each other.

Bimanual interaction on small tactile screens can be tricky, especially when one hand is used to hold the device. However, Wagner et al. [78] showed that it is still possible to design bimanual interaction techniques. They studied how users naturally hold tablets while interacting with them in different situations like walking or sitting, and designed bimanual techniques where the hand that holds the device is used to invoke commands by interacting in the small area that can still be reached by users' fingers. Following this approach, Foucault et al. [74] developed *SPad*, a bimanual interaction technique for activating quasimodes with the thumb of the hand that holds the device, and interact with the content using the other hand.

Gesture-based interaction consists in using finger tips or styli to perform gestures for triggering command, or directly selecting elements displayed on screen. It focuses exclusively on virtual representations displayed on screen. In the next section, we review another interaction style that makes physical objects play a role for interacting with a virtual environment. *Tangible interaction* offers many advantages. In particular, they give an opportunity to design more intuitive interactions that make a more optimal use of humans' manual dexterity and motor skills [79, 80].

2.2 Tangible Interaction

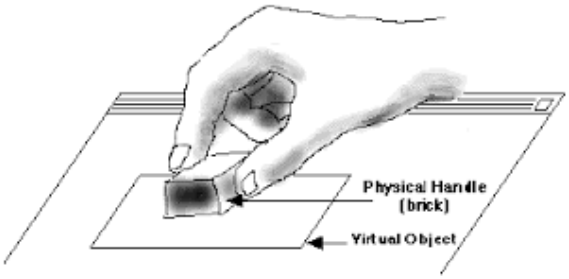


Figure 2.12: Representation of Bricks: a physical manipulation of digital elements.

Source: *[81]*

Tangible interaction relies on systems that use physical artefacts to manipulate, control, organize, and even represent the digital information. It can be seen as an extension and amplification of the concept of direct manipulation. It has been inspired by many different disciplines, including psychology, sociology, robotics, or engineering.

In 1995, Fitzmaurice et al [81] introduced the term tangible/graspable for the first time in the HCI field. The *Bricks* project illustrated how sensing of the identity, location and rotation of two physically tracked 'bricks' on a digital display could be used for controlling graphics. In addition to introducing the notion of Graspable User Interfaces, the system could support bimanual, parallel, and collaborative interactions using these bricks as control virtual elements (Figure 2.12). The number of tangibles was significant. For example, a single brick would result in translation and rotation, while zooming was achieved by manipulating two bricks. Many of these tangible interaction techniques and two-handed manipulation are the ones used nowadays in tangible systems where the objects are connected between them such as FlowBlocks [82] and SystemBlocks [83].

After this first conceptual implementation of tangible interaction, many prototypes have been developed for actually supporting tangible interaction using various approaches including image-based analyses (*e.g.*, DigitalDesk [84] and Urp [85]), electromagnetic sensing (*e.g.*, SenseTable [86]), radio sensing (*e.g.*, Caretta [87]), or an array of multiple sensors (*e.g.*, Metadesk [88]).

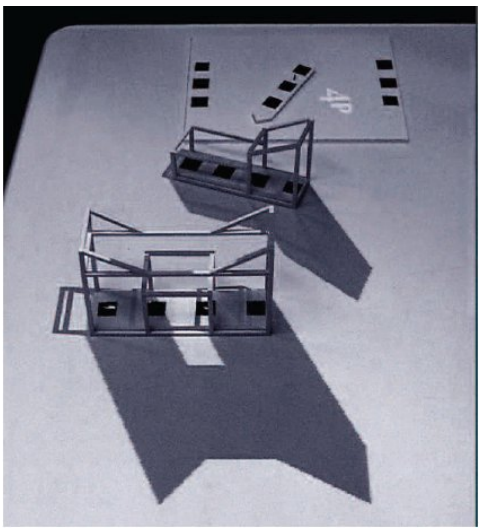


Figure 2.13: Urp: miniature building models that cast digital shadows.

Source: *[85]*

Wellner developed the DigitalDesk prototype [84] to support rapid and direct computer-based interaction on a physical desk. The DigitalDesk is a real desk with a computer projected on it, and a video camera pointed down at it for recording the objects and the user's movements on the surface. The system uses image processing based on thresholding techniques to discriminate when a finger is in contact with the desk screen for *e.g.*, selecting an area or an object. Moreover, the system demonstrated rich interaction techniques that involved both physical objects like paper and virtual elements projected on the desk. Later the metaDESK was built by Ullmp and Ishii focusing on the use of tangibles (physical entities) as controllers for a virtual scene. Underkoffler and Ishii also proposed Urp [85], a tangible interface for urban planning. As illustrated in Figure 2.13, users can place small architectural models on a horizontal surface, and the system simulates their shadows or sun reflections. Patten et al. [86] developed the SenseTable tabletop prototype. The system relies on electromagnetic sensing (wireless tracking) to determine the position of up to ten objects on a horizontal surface. Moreover, the system

offers several improvements over previous tracking approaches like computer vision. For example, the system can track objects without sensitivity to occlusion or changes in lighting condition, and is very responsive to objects state changes.

All the prototypes mentioned above have contributed to make tangible interaction more mature, in the sense that working technologies and interaction paradigms are now in place. However, as sensing fingers and multiple objects at the same time on an interactive surface is non-trivial, most of these projects do not support sensing fingers and the interaction with tangibles remains rather limited.

Since the development of these large setups, many projects have also worked on tangible interaction with small-sized objects or *tokens*. Tokens are small tangible units that physically represent objects or controllers of a virtual application. With exception of active tokens that are equipped with a display, tokens are usually manipulated relative to a display surface. Manipulation of displayed digital information is usually achieved by arranging the different tokens in the spirit of Token+Constraint's approach to interaction [89]. In the rest of this section, we review the literature about tangible tokens.

2.2.1 Radio sensing

Tokens that rely on radio sensing for communication are usually autonomous units that are equipped with a processor and a wireless radio communication unit (bluetooth, wifi, etc). Some of them are additionally equipped with a screen that works independently from any interactive surface.

Such tokens are used in the mediaBlocks system [90], which is a Tangible User Interface that consists of a set of blocks that serve as information containers in order to transfer and store data for digital media. Klum et al. [91] developed Stackables that rely on a vertical spatial arrangement of embedded token-system to express facet queries. Stackables utilize a single token or multiple tokens that could be reprogrammed and support interaction for search. For instance, multiple tokens can be combined with a logical AND or negation through vertical stacking for searching a movie.

In the Siftables project, Merrill et al. [92] presented tokens that are equipped with a color LCD screen, an accelerometer and a radio frequency communication unit, making them able to wirelessly communicate and display visual feedback. Similarly, Sifteo cubes feature a 1.5-inch with full-color clickable screen, and several sensors to detect various

Figure 2.14: Example game with Sifteo cubes.

Source: *[92]*

interactions like shaking, tilting, rotating. They can even detect cubes that are placed adjacent to one another, as illustrated in Figure 2.14. While Siftables and Sifteo cubes can be used to program a large variety of applications, SmartTokens [93] were designed for event notifications and personal task management. These simpler tangibles do not feature any display.

Other projects rely on the Radio Frequency IDentification (RFID) technology to make surfaces able to track tangibles. A RFID reader can actually track a tagged object in a short distance [94]. For example, Kubicki et al. [95] developed an interactive tabletop that uses RFID. The table can detect and identify several tangibles, and even allow users to superimpose tangibles on one another. Antle et al. [96] have also used this technology for developing tangible interfaces for learning. In their system, tokens were identified with a unique RFID tag, and the tabletop prototype included a RFID reader. Children could use tokens to access different pieces of multimedia information. The Activity Pad [97] is another tangible interface that focuses on learning activities. It is a surface covered by a grid of 24 Near Field Communication readers that act as placeholders for tangibles which are augmented with NFC tags. The Activity Pad allows teachers to design educational activities by drawing their interface on an A4 paper sheet and attaching NFC tags to each tangible. They can then put the sheet on the pad, and record the correct location for each tangible. However, the system only supports 24 different locations, which limits the range of possible interactions and applications.

2.2.2 Magnetic Sensing

Magnetic sensing for tangible interaction relies on augmenting objects with magnets so that they will emit a magnetic field that can then be recognized. Magnetic fields can be captured with magnetometers that most mobile devices feature as in *e.g.*, the MagnID project [98] illustrated in Figure 2.15 or the MagGetz project [99]. Magnetic sensing offers the great advantage of tracking objects within a relatively high-amplitude range so that tokens can be manipulated around the sensors without the need of touching them or being very close to them. Hwang et al. have even relied on this approach for building the MagPen [100], which is a magnetic stylus that can be used on and around a regular smartphone. They developed a recognizer that is able to track the orientation of the pen and the pressure applied on screen, as well as identifying different pens. Geckos [101] are also magnetic tokens that offer a larger vocabulary of interaction than only movements. This latter system includes a force-resistive screen to create a constant pressure map (footprint), offering tokens that can detect pressure and gestures performed on top of them, as illustrated in the Figure 2.16. Additionally, as electromagnets can be used to hold objects on a vertical surface, Geckos can also work on non-horizontal surfaces.

Other projects have rather explored the use of Hall sensors to sense tokens' magnetic fields. Liang et al. [102] have proposed to insert a grid of such sensor behind the surface to get a 2D image of the magnetic field. This image can then be analyzed to track the location and orientation of the tangibles above the surface. The problem of magnetic sensing is that it is sensible to interferences between the different fields, making difficult the use of several tangibles at once. A solution to this problem consists of shielding each object with a case made of galvanized steel to avoid attraction and repellence effects between several tokens as Liang et al. [103] proposed in the GaussBricks project.

While magnetic sensing opens a large design space for rich interactions with tangibles, it requires to potentially augment the surface with magnetic sensors, and implement non-trivial solutions to avoid interferences that can impact the different magnetic fields, which can not only occur because of the manipulation of multiple tokens but also because of the presence of ferrous objects in the environment. Furthermore, as illustrated in Figure 2.15, augmenting tokens with magnets might not be trivial.

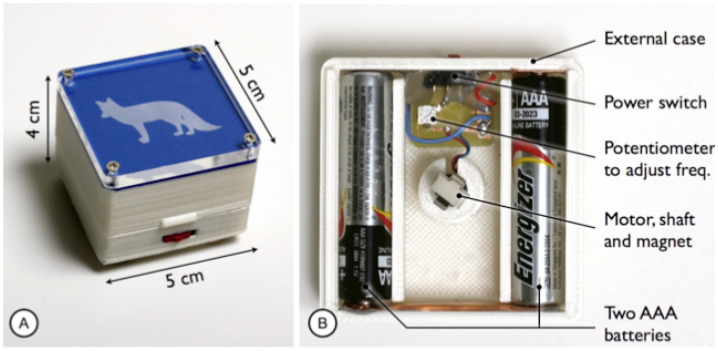


Figure 2.15: MagnID's design

Source: *[98]*

Figure 2.16: A Gecko token can sense pressure applied on top of it

Source: *[101]*

2.2.3 Image-based Sensing

Other projects have explored ways of building tangible interfaces that rely on passive tokens. We can define a passive token as an object that does not embed any electronics itself. Passive tokens are usually more lightweight, and easier to build than active tokens.

Diffuse illumination technology, which we already mentioned for multi-touch input, is the most common approach for enabling tangible interaction with passive tokens on an interactive surface. The system is able to recognize both objects and hands in contact with the surface by using computer-vision algorithms to analyze the frames captured by IR cameras. Such techniques have been used, *e.g.*, to track mice and keyboards [105] or to design physical tools. Vogel and Casiez presented the Conté [104] tool, which is an artistic crayon that consists of an acrylic block that emits and reflects IR light. When

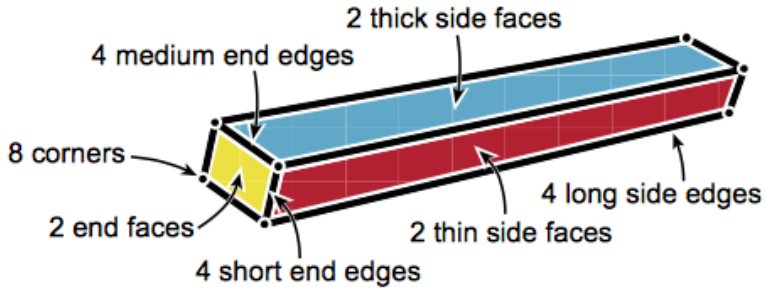


Figure 2.17: The Conté object and the seven types of contacts that the system can detect

Source: *[104]*

tethered, its location and orientation can be tracked on diffuse illumination surfaces. Additionally, they investigated mode switching techniques that detect different sides of the crayon in contact with the surface (Figure 2.17). Weiss et al. [106] developed SLAP Widgets that are physical widgets to interact with a digital representation on a diffuse illumination tabletop.

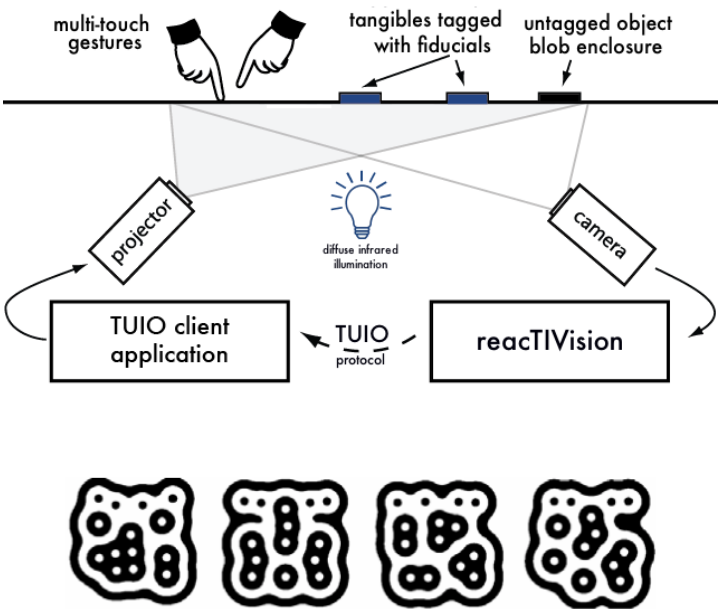


Figure 2.18: The ReacTable's framework diagram

Source: *http://reactivision.sourceforge.net*

Recognizing objects with computer-vision algorithms is much easier if those objects are augmented with fiducial markers. As illustrated in Figure 2.18, Jordà used this approach in the ReacTable project [107]. The interactive tabletop relies on fiducial markers attached beneath the objects that act as controllers for music composition. By rotating or moving the tangibles on the surface, a person (or several people) can produce, transform or control a variety of sounds, beats and notes.

Hence, researchers have investigated tangibles that support more interactions. For example Lumino [108] reflects incoming light to the surface in a specific way in order to support stacks of tangibles, as illustrated in Figure 2.19. Theses complex blocks are built with glass fiber bundle so that their three-dimensional arrangement can be tracked with a diffuse illumination tabletop. A similar approach has been followed by Bartindale and Harrison [109], who developed tokens that feature fiducial markers and transparent areas so that they can be stacked on one another. Williams et al. [110] presented the TZee, which is a transparent tangible that has the shape of a truncated pyramid and that supports gesturing on its sides.

In most projects that make use of diffuse illumination technology, the camera is beneath the surface. However, the cameras can also be located elsewhere. For example, Portico [111] uses two small cameras mounted on foldable arms that are attached to the sides of a tablet in order to track objects on and around the screen. Portico's vision-based algorithm is advanced enough to recognize untagged objects.

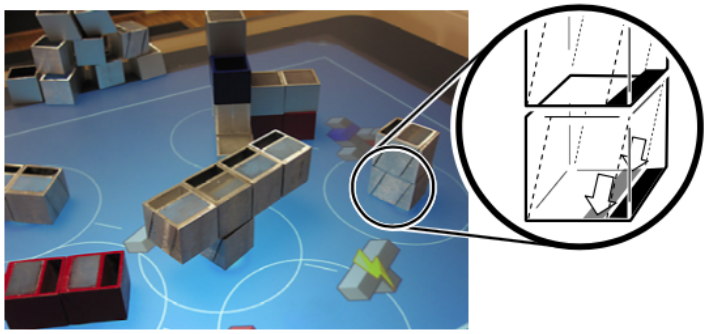


Figure 2.19: Luminos are tangible building blocks that allow the underlying diffuse illumination table to track their 3D arrangement

Source: *[108]*

While diffuse illumination technology offers rich possibilities for object and hand

detection through computer-vision methods, it also has some disadvantages. In particular, it requires specific environmental conditions to avoid issues related to lighting condition. It is also sensible to occlusion and usually relies on hardware setups that are bulky.

2.2.4 Capacitive Sensing

Several research projects have also investigated how tangible interfaces can be built on capacitive screens, which is the most widespread technology for tactile surfaces. Capacitive surfaces detect a drop in capacitance when one or more fingers touch them. This technology has multiple advantages. First, it is available on most popular touch screens in the market. Second, the screen resolution can be high, without suffering from lighting conditions. Finally, the accuracy for detecting and tracking fingers positions is very high.

The most common approach for making such surfaces able to track objects consists of making a conductive circuit between users' fingers and the capacitive surface through the tokens' feet that are in contact with the surface. As soon as the user touches the token, the feet become grounded and generate a drop in capacitance. The design of these passive tokens requires implementing unique token feet configurations, as fiducial markers must be unique when using diffuse illumination technology.

Following this principle, researchers have built physical widgets [112], physical button pads that can be clipped to the edges of a device [113], or even more advanced objects that feature moving parts [13] or that can be stacked [114]. For example, Capstones and ZebraWidgets, which are illustrated in Figure 2.20, are capacitive units that can be assembled to configure different conductive circuits, enabling more manipulations with the tangibles that can, for instance, be stacked or feature moving parts.

However, designing conductive tokens is not straightforward, as capacitive screens have been designed for human fingers. Tokens feet must be carefully positioned (minimal size and the minimal distance between two feet), and the circuit must be stable so that the generated touch pattern can be recognized [115]. Other projects have explored more cost-effective ways of building conductive objects. For example, Wiethoff et al. [116] use cardboard and conductive ink to build conductive tangibles (Figure 2.21). While this approach enables low-fidelity quick prototyping and encourages iterative design, it does not scale to real usages. Blagojevic and Plimmer [117] presented a design experience where they built a small set of geometric tools such as a ruler, a protractor and a set

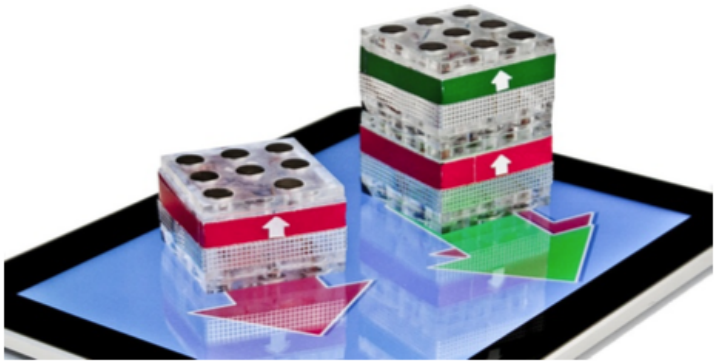


Figure 2.20: CapStone: example of stackable blocks

Source: *[114]*

square for a tabletop drawing application. They tested several iterations in their design process for constructing the tangible hardware by combining different low-cost conductive materials (*e.g.*, conductive ink, conductive foam, aluminium tape, copper wires). Their experiment showed that making a physical tool conductive is quite difficult, as many factors have to be considered (consistent circuit, stability, friction with the screen, good grasp, etc.).

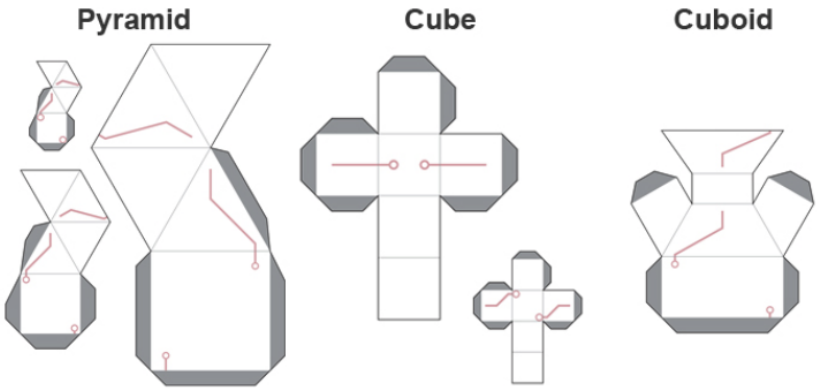


Figure 2.21: Example template of paper objects.

Source: *[116]*

An important limitation of conductive tokens is that they can be detected by the capacitive surface only when users touch them. PUCs [118] are an exception. They rely on the principle of mutual capacitance so as to be detected even when users do not

touch them. However, this approach requires to augment the surface with an additional calibration clip to cheat the implemented adaptive filtering that tends to interfere with the PUCs' detection.

This overview of tangible interaction shows that building objects that can be used to interact with a virtual world is not straightforward, and either require some knowledge in electronics or use image-based analysis techniques on setups that are sensible to environmental conditions.

2.3 Application domains

In this section, we list the different application domains that can benefit from gesture-based and/or tangible input. Researchers have considered using these input channels in various application domains such as music composition, collaboration, games, or education. This list is not exhaustive. Tokens and gestures have been proposed for many more concrete applications such as programming [119], data base querying [120], or big data manipulation [121]. However, this section illustrates the variety of what these input channels can control, and how they can enhance user experience.

MUSIC COMPOSITION

Music composition on multi-touch surfaces has become very popular. It is a field that often requires many dimensions to control for creating sound, modifying frequencies, synchronizing elements, etc. The combination of gesture-based interaction and tangible interaction allows artists to control many instrument dimensions, and can also bring the necessary dynamics to artistic composition. For example, Audiopad [122] is a composition and performance instrument for electronic music. As illustrated in Figure 2.22, the system is able to track the object positions on a tabletop surface, and converts their motion into music. Similarly, the Scrapple [123] is a musical instrument that synthesizes music based on how several shaped tangibles are laid out on a rectangular table. The ReacTable project [124] is one of the most popular and commercial multi-touch tabletop and tangible instruments for electronic music performance. The system is a circular tabletop surface that offers two ways of interactions. It allows users to not only put tokens on it for making music and second, but also to adjust parameters by interacting with the

touch-based widgets that are displayed down or around the token. AudioCubes [125] is another commercial system. Its approach consists of active cubes that can communicate with each other. The cubes can be positioned on a table, and the users can modify sounds by changing their relative positions.

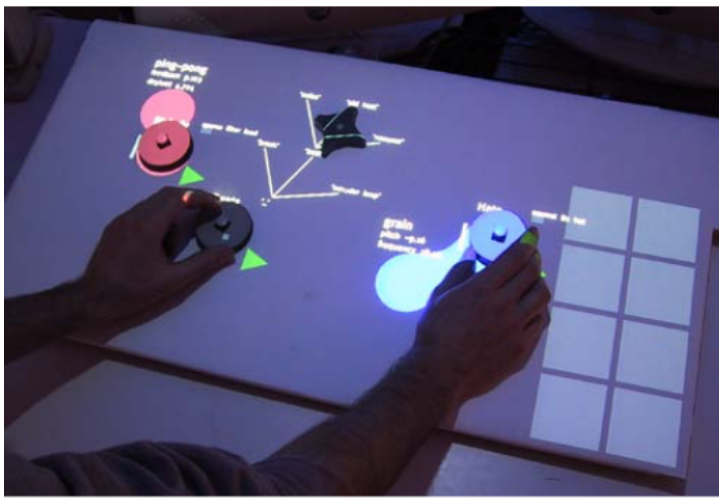


Figure 2.22: The Audiopad prototype being manipulated.

Source: *[122]*

COLLABORATION

Instead of restricting input to an ordered sequence of events (click, double click, etc.), multi-touch surfaces accept several touch events at potentially distant locations, enabling several users to work collaboratively. Furthermore, tangible interfaces have the potential of facilitating several kinds of collaborative activities [121] that are not possible or poorly supported by single user technologies like mouse or mobile phones. This makes these input channels especially suited to work with large screens such as tabletops and wall-sized displays that allow co-located users to work together on and around the same content [13] (Figure 2.23). For example, Kobayashi et al. [126] presented a TUI prototype, based on physical pucks and an interactive tabletop, for collaborative IP network design. Using this system, a group of experts can work together to manipulate directly network topologies, control multiple parameters of stations and links. Experts can also run simulations, and get the result of their simulations in real time.

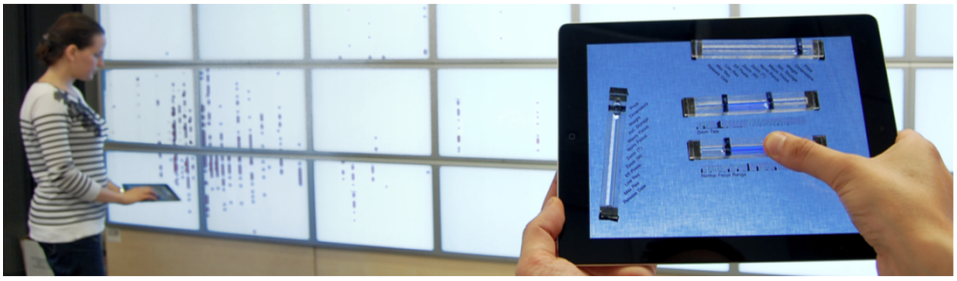


Figure 2.23: Two users manipulating the Wall display using their tangible remote controllers.

Source: *[13]*

GAMES

Tangibles, such as pawns or pucks, can serve as game controllers. For example, Figure 2.24 shows PERCs tangibles [127] illustrated in the context of a game where the tangibles are spaceships that can be moved and rotated on a 2D surface that depicts space. Miners [128] is another example of a game on a large interactive surface where multiple users can cooperate for rescuing workers in a mine using gestures and tangibles. Many projects have also proposed tangible games for children (*e.g.*, [129–131]). For example, Xie et al. [132] proposed a tangible interface for resolving a jigsaw puzzle. They performed a comparative study between this tangible interface, the real puzzle and a graphical interface. They observed that the tangible interface was offering some benefits for collaborative problem solving.

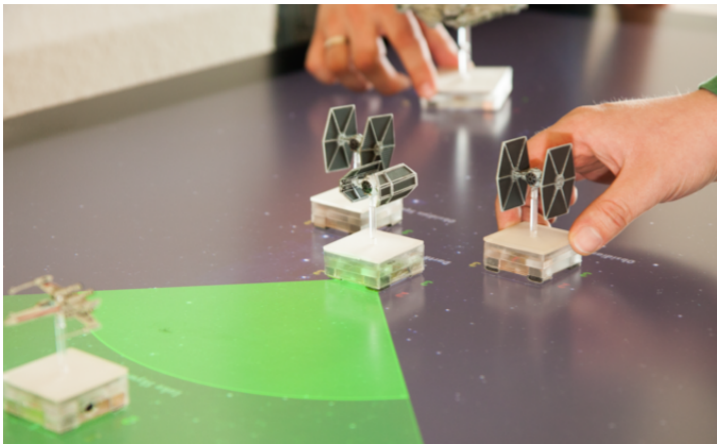


Figure 2.24: PERCs: using tangibles as ships in an interactive board game on a multi-touch screens.

Source: *[127]*

EDUCATION

Many studies have emphasized the educational advantages that tangibles can offer. Some of them have focused on assisting children in simple tasks. For example, the Read-It application proposed by Sluis et al. [133] showed that tangibles help children between 7-to 9-year old in their reading activities. Marco et al. [134] proposed a simple math application where children place little pieces on top of the surface to complete

simple tasks like additions and subtractions. Other projects have studied physical input and tangible representation for learning complex concepts. Do-Leng et al. [135] proposed a system that involves a multi-touch tabletop and physical paper. The project enables users to engage in natural and one-to-one mapping between physical objects and virtual concepts to support learning tasks. Combinatorix [136] is a project that combines tokens with an interactive tabletop to support students in learning, resolving and understanding mathematical complex problems (Figure 2.25). Many interactive museums also combine gesture and tangible interaction with interactive displays for explaining concepts to the visitors. For example, BacPack [137] is an interactive museum exhibit that engage visitors in the process of designing bacterias to resolve a concrete problem. The system allows visitors to explorer personal compositions to create bacterias, collaborate with others in the same space and observe the result in real time.

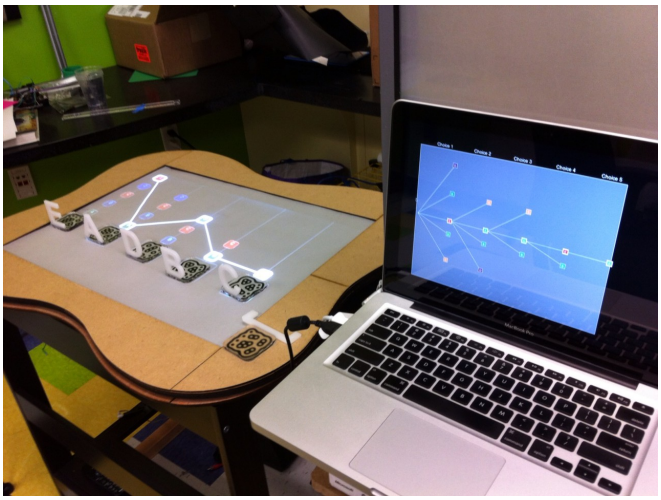


Figure 2.25: Combinatorix: tokens control a tabletop display (left) and a probability tree (right screen).

Source: *[136]*

2.4 Summary

The number of projects investigating gestures and/or tangibles on multi-touch surfaces is very large, reflecting the importance of this approach. The diverse application prototypes that have been developed with these types of input also suggest that they can be beneficial to many application domains.

Researchers have been basically following two approaches for discriminating gestures performed on a multi-touch surface. The first approach is based on the number of contact points involved in the gesture, and the second one is the contact points' trajectory. Only a very few research projects have taken into account the relative spatial configuration of contact points, with the aim of associating a contact point with a specific finger. In the next chapter, we introduce the core contribution of this thesis, TOUCHTOKENS, which are passive tokens that allow users to perform specific touch patterns, a property for discriminating multi-touch gestures that has not yet been considered.

Because of their physicality, TOUCHTOKENS also belong to the family of tangible interfaces. However, as opposed to all the projects that involve tangibles which we reviewed in this chapter, TOUCHTOKENS do not require a certain level of knowledge in electronics or in image processing. They are fully passive tokens that simply feature notches indicating how users should grasp them. When held on the capacitive surface, the system can recognize the specific touch pattern that is associated with the notches configuration. They thus enable the implementation of tangible interfaces with easy-to-fabricate and low-cost tokens and a regular tactile surface such as a smartphone or a tablet.

CHAPTER 3

TouchTokens - Combining Tangible and Gestural Input

In the previous chapter, we observed that only a few researchers have considered the relative spatial configuration of contact points for designing input vocabularies for tactile surfaces. We believe that it may due to two main raisons. On the one hand, users may find this type of gestures difficult to learn and perform [67] as it requires to retrieve and adopt an exact finger configuration. On the other hand, recognizing the corresponding touch patterns may require to resort to complex recognition algorithm such as machine learning approaches [69] for discriminating several patterns.

Our initial idea was that a physical object would afford a natural grasp, and would thus guide a specific finger configuration when holding it. Different object shapes and sizes will lead to different grasps that would be different enough to be discriminated. This approach would have the interesting counterpart of making objects recognizable based on a finger configuration. We decided to test this hypothesis in the context of passive objects held on a tactile surface. Our hope was that, when users hold different objects on a multi-touch surface, we could discriminate the resulting touch patterns because they would be guided by the object affordable grasp. This chapter presents our initial investigation of this idea, and how we found out that, although users were adopting quite coherent grasps for a given object and that different objects lead to different grasps, there is still some variability in grasps and strategies for grasping a given object. This

variability prevented us from designing a recognition engine that could classify these different grasps with a high enough accuracy. This leads us to the design of specific objects, TOUCHTOKENS, that feature notches to clearly indicate how users should grasp them.

This chapter starts with a presentation of TOUCHTOKENS' principle, which is a system that consists of a set of easy-to-make passive tokens and a fast recognizer that is able to discriminate the unique touch pattern associated with each token in the set. Then, we report on a formative user study to collect touch patterns, in which participants had to grasp and manipulate a set of twelve tokens of varying shape and size on a tabletop surface. Finally, we develop a recognition algorithm that can classify the resulting patterns with a high level of accuracy (>95%) without any training, enabling application designers to associate rich touch input vocabularies with command triggers and parameter controls.

3.1 Functionality Overview

The primary objective of TOUCHTOKENS, Figure 3.1, is to guide the registration pose [138] of multi-touch gestures on an interactive surface. TOUCHTOKENS take advantage of users' ability to grab physical objects in the real word. When users grab the token and place it on the surface, touching the token and the surface at the same time, the system recognizes a specific touch pattern associated with the token. Moreover, these touch patterns are recognized at registration and remain active until all contact points have left the surface. In particular, users can relax their grasp in the execution phase of their gesture, thus reducing finger occlusion and enabling a larger range of motion.

3.2 Hypothesis

Our hypothesis is that the geometry (shape and size) of an object impacts how users grab it. Different objects will thus have different touch patterns on the tactile surface, which can be discriminated. In the subsection Formative Experiment we report on a study in which participants had to grasp a set of twelve tokens that vary in shape and size to demonstrate our hypothesis. We first present our recognition algorithm.

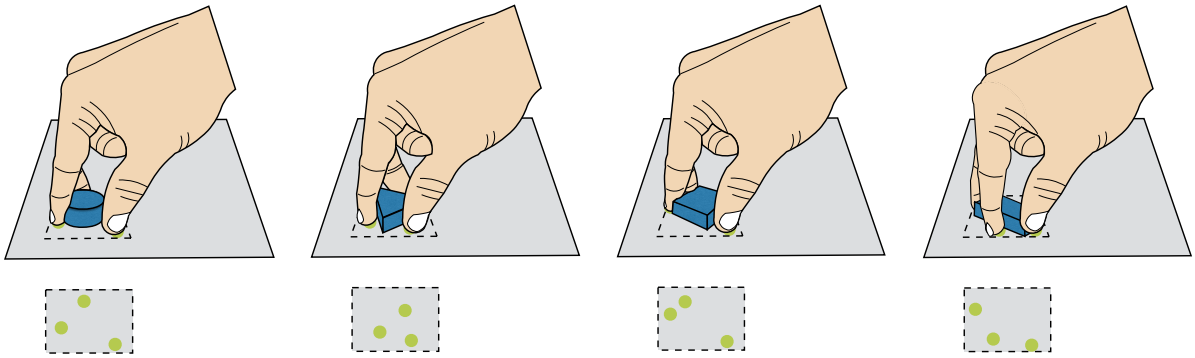


Figure 3.1: Passive tokens that guide fingers to specific spatial configurations, resulting in distinguishable touch patterns.

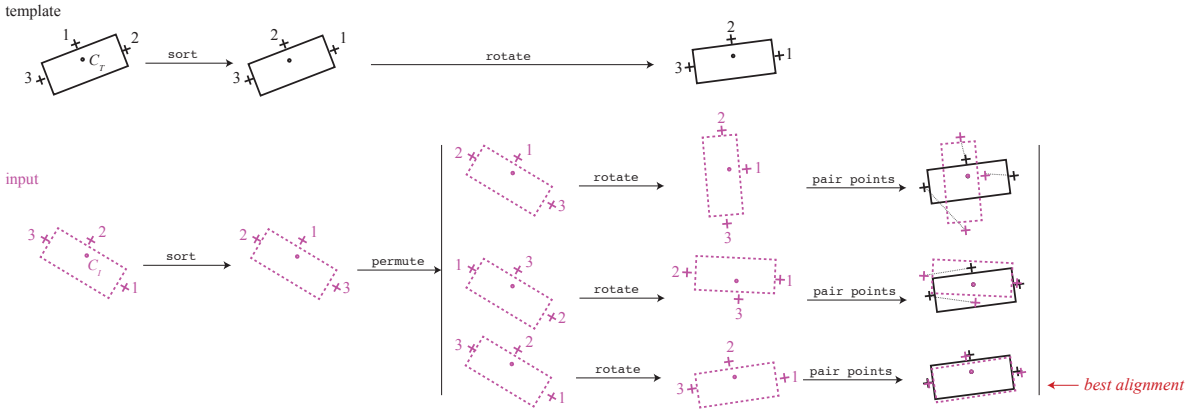


Figure 3.2: Template and input touch pattern alignment process.

3.3 Recognition Algorithm

When grabbing a token with more than two fingers in contact with the surface, TOUCHTOKENS can infer its identity, and thus the corresponding registration pose, from the relative spatial configuration of the touch points. The recognition engine is initialized with one or more typical touch patterns per token and, when a touch pattern of at least three points occurs, the algorithm computes the distance between this input pattern and the set of template patterns. The recognized token is the one associated with the template that minimizes this distance metric.

Computing the distance between two touch patterns (input I:$\{I_1,...,I_n\}$ and template T:$\{T_1,...,T_n\}$) is not straightforward, however. First, most tactile surfaces do not provide finger identification. Second, tokens have an arbitrary orientation on the surface. Fig-

ure 3.2 illustrates how our algorithm processes touch patterns in order to identify the *best alignment* between the reference template and actual input patterns, from which the distance is computed.

The key steps for identifying the best alignment are as follows: (1) compute the centroid C_I of the three (or more) touch points; (2) generate all sequences of touchpoint labels (permutations) so that their IDs always appear in counterclockwise order; (3) rotate all these touch patterns so as to align vector $\overrightarrow{C_I I_1}$ with the x-axis. (4) The algorithm then translates touch patterns to align the input (C_I) and template (C_T) centroids. (5) It finally pairs the points in the permutation with the template's points in order to compute the distance, simply by summing all distances between paired points. The distance between reference template and actual input is given by the *best input alignment*, which is the permutation that minimizes this distance metric.

A typical implementation of the recognition engine amounts to about a hundred lines of code, and will work on any capacitive surface. The engine relies on simple geometrical features, which makes it easier to understand recognition errors compared to less transparent techniques such as those based on machine learning, that work as black boxes. The algorithm is very fast: recognition time scales linearly with the number of candidate templates. A Java implementation is available publicly, featuring both TUIO and Android APIs at [1].

3.4 Formative Experiment

TOUCHTOKENS project relies on the hypothesis that the geometry of tokens impacts how users grasp them, resulting in distinguishable touch patterns. In order to test this hypothesis and identify a set of tokens that can actually be discriminated, we ran a study in which participants had to grasp a set of twelve tokens that vary in shape and size.

3.4.1 Token Set

We selected a set of $4 \times 3 = 12$ tokens (Figure 3.3) that vary in their shape (square, circle, rectangle, and triangle) and size (3cm, 4cm and 5cm). The choice of size was informed by informal tests, taking into account both human and technological constraints. The tokens should remain comfortable to grasp with at least three fingers, which entails biomechanical constraints on the minimum and maximum token size. Capacitive surfaces

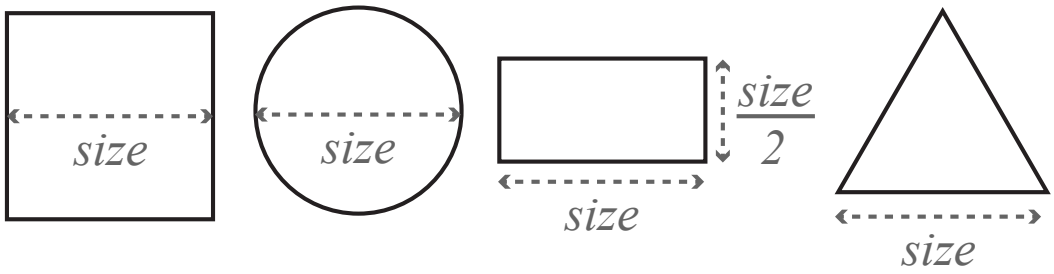


Figure 3.3: Set of tokens used in the first study ($size \in 3cm, 4cm, 5cm$).

also impose a minimal distance between finger tips, which will be seen as a single point if too close to one another.

Our tokens are made of wood and are 6mm thick. We had initially considered tokens 3mm thick, but those were too difficult to grab. The tokens' corners are also slightly rounded so as to avoid sharp wedges that could have hurt participants.

3.4.2 Types of Interaction

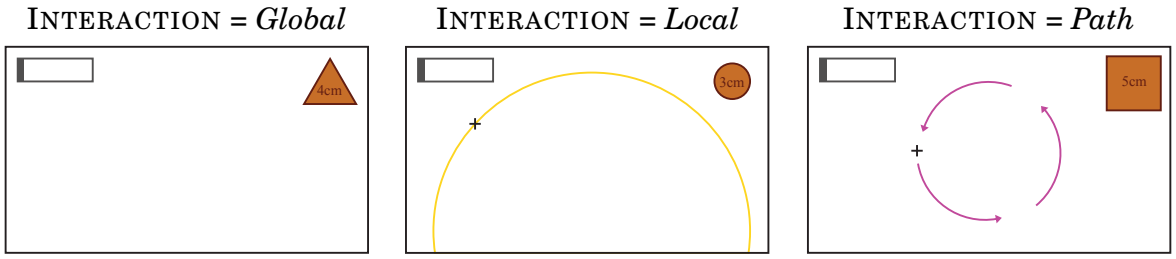


Figure 3.4: Types of interactions.

Participants are seated in front of the tabletop (at the center of the long edge) and perform a series of trials with the different tokens (Figure 3.5). As illustrated in Figure 3.4, the graphical display always features a progress bar in the top-left corner and a picture of the token to use in the current trial in the top-right corner. The action to be done with the token depends on the type of interaction (INTERACTION):

- **The *Global* condition:** operationalizes the case where users invoke a global command with the token (e.g., launching an app).

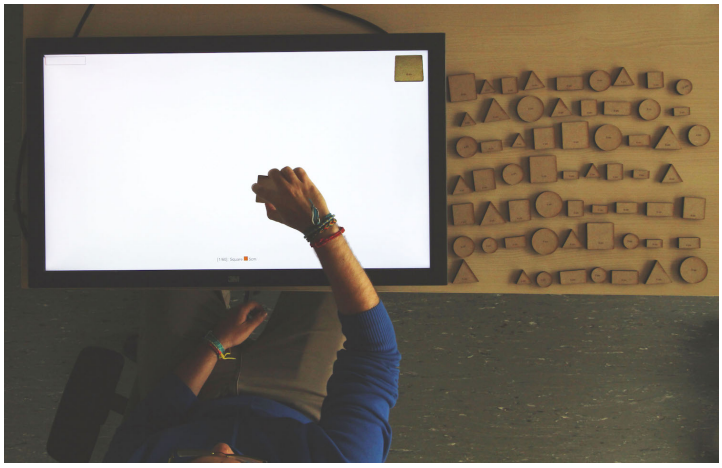


Figure 3.5: Experimental setup.

- **The *Local* condition:** corresponds to the case where users apply a command at a specific location on screen (e.g., copying a graphical object).
- **The *Path* condition:** captures the case where users invoke a command and set its parameter value with a gesture (e.g., adjusting the opacity of a layer in a visualization).

In each INTERACTION there is a progress bar that indicates for how long participants have *dwelled*. It starts filling-in as soon as a stable touch pattern is detected on the surface. The dwell's duration depends on the type of interaction. If the number of fingers in contact changes, or if the touch pattern's centroid drifts away from its initial position by more than 30 pixels, the progress bar is reset and participants have to perform the trial again.

The experiment was divided into three phases, one per INTERACTION condition, always presented in the same order:

1. In the first phase (INTERACTION = *Global*), participants have to select the right token, put it anywhere on the tabletop, hold it with at least three fingers, and hold still for at least 1 second.
2. In the second phase (INTERACTION = *Local*), participants have to select the right token, put it on the cross (Figure 3.6), holding it still with at least three fingers for at least 1 second. The cross can be in five different LOCATION. These locations

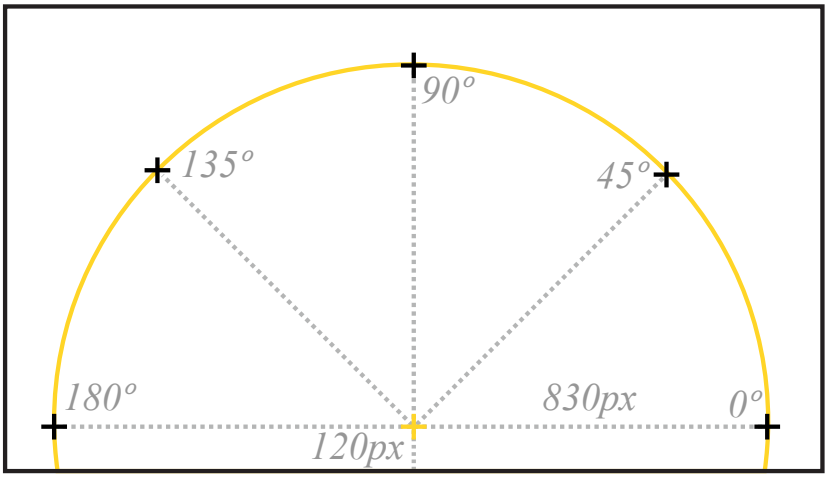


Figure 3.6: In the INTERACTION = *Local* condition, participants have to put the token at a specific location (LOCATION ∈ 0°, 45°, 90°, 135°, 180°).

are chosen on a semi-circle roughly centered on the participant as in [139] (see Figure 3.6), as the token's location on the surface (relative to the participant) may influence the neutral hand posture and thus how the token is grasped. The distance between the touch pattern's centroid and the center of the cross must be at most 50px. If this distance is greater, the progress bar turns red and participants must perform the trial again.

3. In the third phase (INTERACTION = *Path*), participants must hold the token still with at least three fingers for a short period of 100ms. The background turns from gray to white. Participants then have to slide the token along the path indicated by purple arrows. In this condition, participants can plan a manipulation with the token, which may influence their initial grasp [139]. When sliding the token, they can lift some fingers but must keep at least one finger in contact with the surface. If they lift all fingers before having performed the whole gesture, the background turns back to gray and they have to start again. Figure 3.7 shows the six types of paths that participants had to follow with each token. We chose these tasks based on the taxonomy of multi-touch gestures from [140]. For external circular gestures (*Ext-CCW* and *Ext-CW*), participants have to slide the token along a clockwise or counterclockwise circular path. As soon as the touch pattern's centroid has completed one full circle, the background turns green and participants can proceed to the next trial. For internal circular gestures (*Int-CCW* and *Int-CW*), participants have to rotate the token around its center, as they would do with a

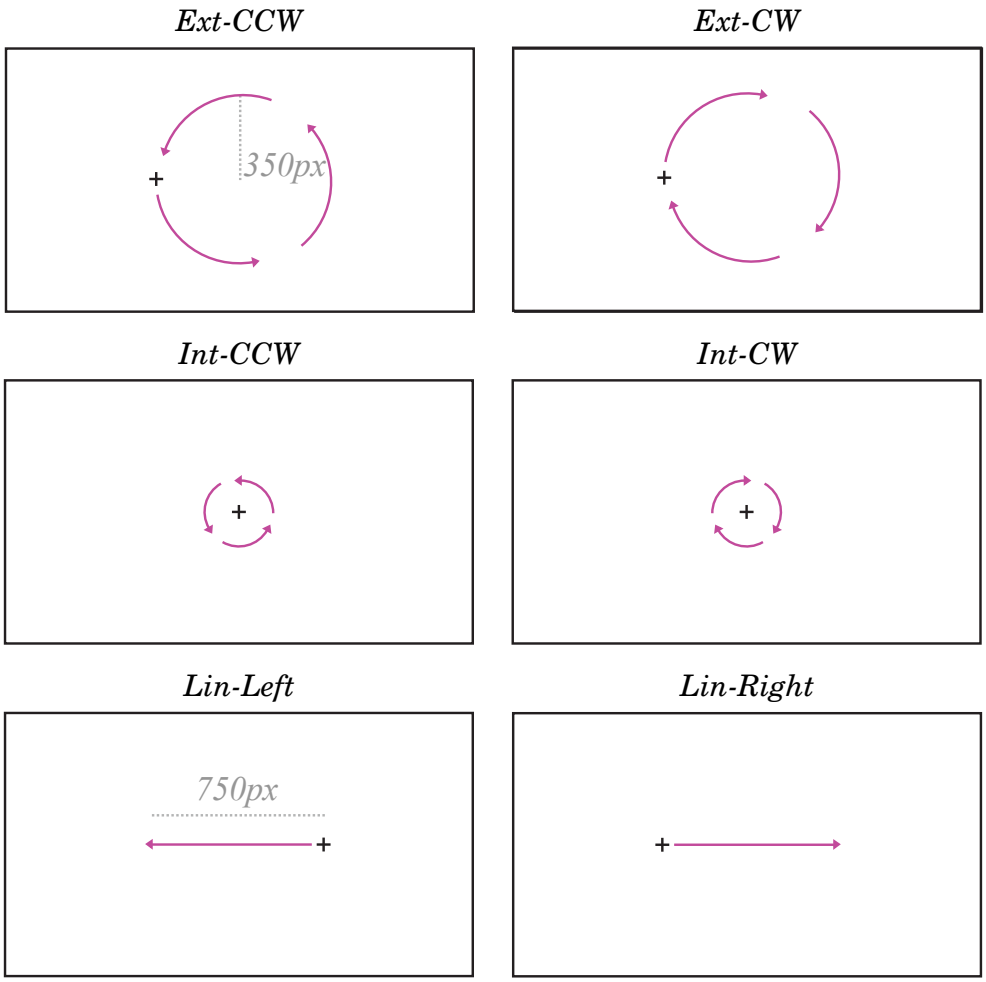


Figure 3.7: In the INTERACTION = *Path* condition, the participant has to put the token on the surface and slide it along a specific path (PATH ∈ {*Ext-CCW*, *Ext-CW*, *Int-CCW*, *Int-CW*, *Lin-Left*, *Lin-Right*}).

physical circular knob. As soon as the touch pattern has been rotated by at least 45° around its centroid, the background turns green to indicate that the trial has been successfully completed. Finally, for linear gestures (*Lin-Left* and *Lin-Right*), participants simply have to slide the token to match the amplitude and direction indicated by the arrow.

3.4.3 Participant and Apparatus

Twelve users participated in this experiment, and manipulated tokens on a 3M C3266P6 capacitive screen with dimensions 698.4 x 392.85 mm (1920 x 1080 pixels) that was

placed horizontally on a desk.

The experimental software was developed in Java 2D (JDK 7) and ran on a Mac Pro 2.8 GHz Intel Quad Core with 16GB memory, running Mac OS X 10.7.5. The experiment consists of three phases that will describe in the next section.

3.4.4 Procedure

Participants are seated at the center of the long side of the tabletop. They receive instructions detailing the goal of the experiment and the different experimental tasks they will have to perform. In particular, the operator initially informs participants that the goal is to design a system that is able to recognize tokens based on users' grasp. He encourages them to be consistent in their grasp across trials with tokens that have the same shape. In order to identify which grasp is comfortable, the operator gives participants four tokens, one per shape with $size = 4cm$ ($Square_4$, $Circle_4$, $Rectangle_4$ and $Triangle_4$), and asks them to manipulate each token a bit on the surface in order to choose a comfortable grasp. The operator then notes this grasp in his logs and the experiment starts.

As mentioned above, the experiment consists of three phases that are always presented in the same order:

- Phase 1 (INTERACTION = *Global*): 12 TOKEN × 5 repetitions = 60 trials. In this phase, the presentation order for the trials is randomized in order to observe if people are actually able to grasp the same token consistently across different trials that are not consecutive. To minimize the visual search time associated with identifying the right token to take, the operator printed 5 copies of each individual token and initially sorted the 60 tokens on the table, on the right side of the screen (Figure 3.5).

- Phase 2 (INTERACTION = *Local*): 12 TOKEN × 5 LOCATION × 2 repetitions = 120 trials. The order of TOKEN × LOCATION is randomized across participants. The 2 repetitions per TOKEN × LOCATION condition are presented one after another to limit the length of the experiment.

- Phase 3 (INTERACTION = *Path*): 12 TOKEN × 6 GESTURE × 2 repetitions = 144 trials. As in phase 2, the order of TOKEN × GESTURE conditions is randomized

across participants, with the 2 repetitions presented one after another.

After completion of these three phases, participants receive a questionnaire where they have to give a comfort score for each of the twelve tokens. The questionnaire features 12 Likert-scale type questions where participants have to give a rating between 0 (not comfortable to grasp at all) to 5 (very comfortable). The overall procedure lasted about an hour.

3.4.5 Results

We first tested if participants' grasps of the different tokens can be distinguished using the recognition strategy described in the previous section. To that end, we train our recognition algorithm using the first three trials of Phase 1 as templates for each token. This training strategy corresponds to what a system relying on a light training phase would require. We then evaluate our algorithm on the remaining trials, i.e., $(2 \times 12 + 2 \times 12 \times 5 + 2 \times 12 \times 6) \times 12$ participant = 3456 trials. We also tested our algorithm with different training strategies to accommodate more variability (e.g., considering templates picked from the three experiment phases) but there was no clear gain compared against the training cost it would entail for end-users.

A χ^2 analysis reveals that both INTERACTION ($\chi^2(2, N = 3456) = 12, p = 0.002, \phi = 0.06$) and TOKEN ($\chi^2(11, N = 3456) = 109, p < 0.001, \phi = 0.18$) have a significant effect on RECOGNITION RATE. Figure 3.8 illustrates the observed differences between conditions. A finer analysis of TOKEN's effect on RECOGNITION RATE per INTERACTION shows that TOKEN has a significant effect on RECOGNITION RATE in all INTERACTION conditions (*Global*: $\chi^2(2, N = 288) = 25, p = 0.009, \phi = 0.3$, *Local*: $\chi^2(2, N = 1440) = 58, p < 0.001, \phi = 0.2$ and *Path*: $\chi^2(2, N = 1728) = 85, p < 0.001, \phi = 0.2$). The effect of secondary factors (LOCATION for *Local* and GESTURE for *Path*) on RECOGNITION RATE is not significant ($p = 0.4$ and $p = 0.2$).

We then wanted to investigate the impact of the token subset's size on recognition rate. In order to identify the largest number of grasps that can be accurately discriminated for each participant, we computed all possible subsets of tokens among the initial set of 12. The total number of subsets comprising at least two tokens (TOKENCOUNT >= 2) is:

$$\sum_{TokenCount=2}^{12} \binom{12}{TokenCount} = 2^{12} - 12 - 1 = 4083$$

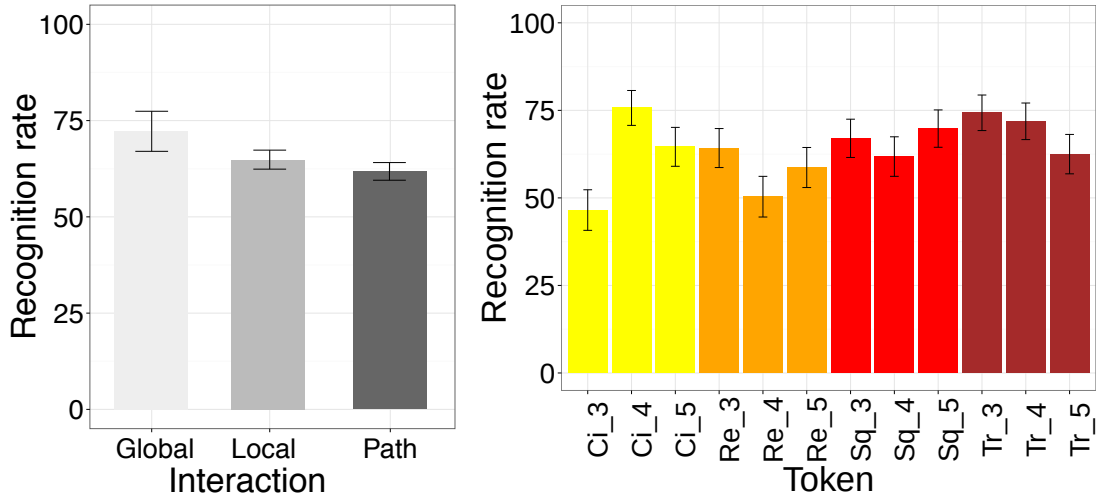


Figure 3.8: Recognition rate per INTERACTION (left) and per TOKEN (right). Error bars represent the 95% confidence interval.

For each subset, we ran our recognition algorithm with the same training strategy (only the first three trials from experiment phase INTERACTION = *Global*) in order to compute, for this subset, the recognition rate per participant. We observe that the per-subset recognition rate across participants exhibits a very high variability. For example, if we consider subsets that have 7 tokens (TOKENCOUNT = 7), the "worst" subset has a recognition rate of 63% on average across participants (worst-performing participant: 31%, best-performing participant: 98%), while the "best" subset has a recognition rate of 81% on average across participants (worst-performing participant: 57%, best-performing participant: 100%).

RECOGNITION RATE PER PARTICIPANT

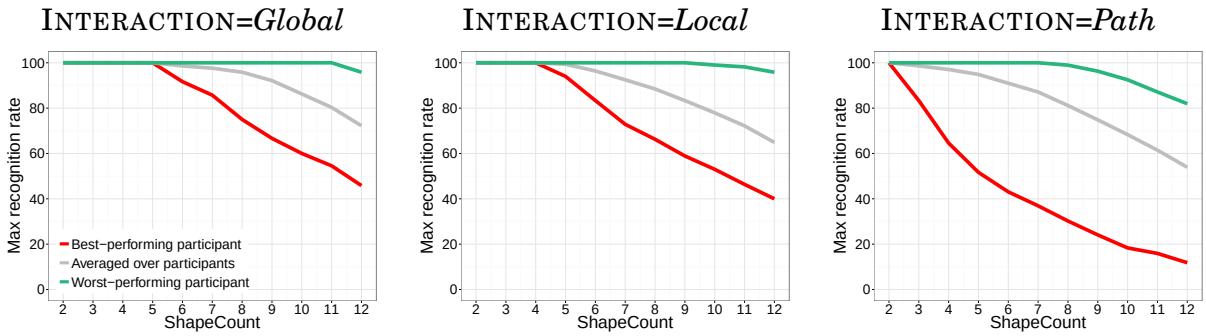


Figure 3.9: Recognition rate per INTERACTION × TOKEN

In order to test how many distinguishable grasps can be recognized per participant, we report the maximal recognition rate for each value of TOKENCOUNT $\in \{2, ..., 12\}$. If a participant P gets a maximal recognition rate R for TOKENCOUNT=N, this means that there exists at least one set of N tokens that are recognized with $R\%$ accuracy on average for participant P. Figure 3.9 reports these recognition rates for the best-performing and worst-performing participants, as well as the average over all participants. The charts illustrate that our algorithm can accurately discriminate a high number of grasps for some participants (the best-performing participant has a recognition accuracy higher than 90% for up to 10 tokens in all INTERACTION conditions), while it performs quite poorly for others (the worst-performing participant has a recognition accuracy lower than 90% even for sets of only three tokens in condition INTERACTION = *Path*). This variability comes from two sources: *intra-grasp variability* and *inter-grasp similarity*.

Figure 3.10 displays the 27 touch patterns we have collected for $Triangle_4$ for two participants. It illustrates two extreme levels of *intra-grasp variability*. Participant 1 (left) grasps token $Triangle_4$ in a very consistent manner, while Participant 9 (right) demonstrates much more variation in how he grasps it, challenging our recognition strategy. The second source of confusion comes from *inter-grasp similarity*: if a user chooses one grasp strategy for a given token that is very similar to the one he uses for another token in terms of similarity of the touch patterns, the two tokens will get confounded. Together, these two phenomena explain why we observe such a large variability across participants regarding the composition of the token sets that are recognized accurately.

RECOGNITION RATE BETWEEN PARTICIPANTS

Figure 3.9 reports the best set of tokens for each participant, and thus does not reflect the fact that the same subset of tokens can be very accurately recognized for one participant while it will be poorly recognized for another participant. We report the biggest sets of tokens that reach consensus among all our participants below (i.e., the sets of tokens that have a recognition accuracy of at least 90% for all participants):

- for INTERACTION=*Global*, we find 6 sets of 5 tokens;
- for INTERACTION=*Local*, we find 13 sets of 3 tokens;
- for INTERACTION=*Path*, we find 6 pairs of tokens;

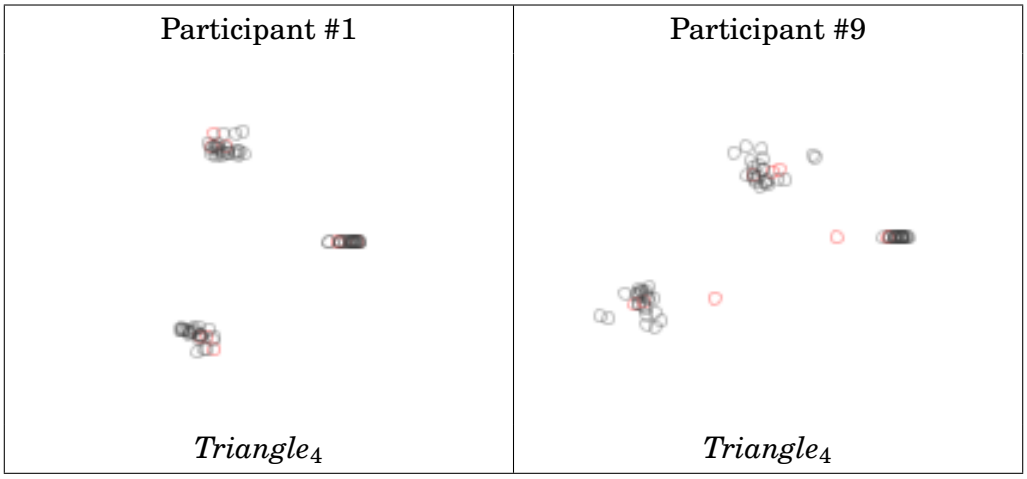


Figure 3.10: Touch patterns (aligned by our algorithm) for a participant who adopts very consistent grasps for token $Triangle_4$ (left) and for a participant who adopts varying grasps (right). Red dots belong to touch patterns that are used as templates.

- for all INTERACTION conditions undifferentiated, we find 8 sets of 3 tokens with an average of at least 90% accuracy for all participants.

3.4.6 Grasp strategies

Figure 3.11 summarizes the different grasp strategies that participants adopted for the different token shapes (extracted from an analysis of the operator's logs and video sequences recorded during the experiment). We observed that all participants use the same strategy for circles (C). Squares and rectangles receive less consensus, with three different strategies observed for each of them. The main strategy for squares uses three edges (S_1; 6/12). The two other strategies use only two edges, and differ in the distance between the two fingers on the same edge: small (S_2; 4/12) or large (S_3; 2/12). For rectangles, one strategy uses the two long edges only (R_1; 5/12). The two other strategies use three edges: two contact points on the short edges (R_2; 4/12) or on the long edges (R_3; 3/12). One of the grasp strategy for triangles makes use of a corner (T_2; 2/12), which was quite surprising. Two participants adopted it, but actually rated it as very uncomfortable.

To understand what kind of confusions occur in the recognition process, we implemented a visualization that displays all collected touch patterns using the best alignment computed by our recognition algorithm (Figure 3.10 was built with this tool). We com-

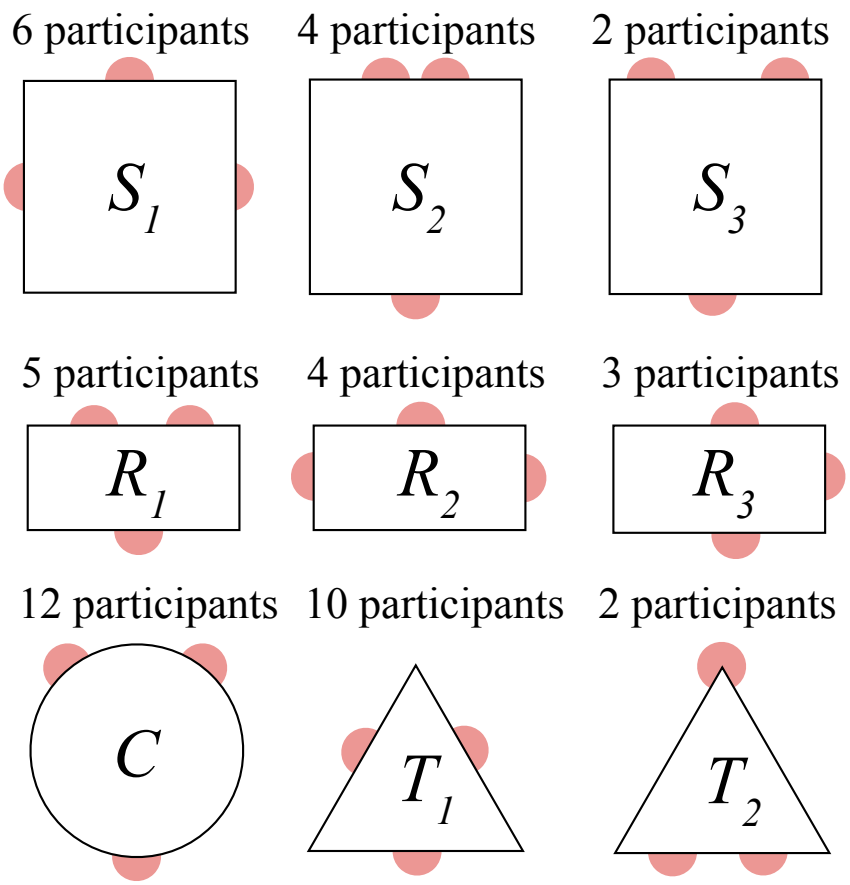


Figure 3.11: The nine grasp strategies observed in Experiment 1

puted the confusion matrix by considering the 27 types of touch patterns (3 $size$ × 9 grasp strategies). The visualization tool was a good complement to the confusion matrix as (1) some confusions do not appear in the matrix if a template for one token is too close to a template for another token; and (2) the different grasp strategies were not adopted the same number of times, leading to numbers in the confusion matrix that could not be compared in an absolute manner. From this analysis, we draw a few take-away messages. The flat isosceles triangle of grasp strategy R_2 is very representative and well-recognized. T_2 is also representative, but is too uncomfortable to be further considered. In contrast, some postures are difficult to distinguish. For instance, touch patterns R_1 and R_3 often form an equilateral triangle similar to the one of T_1. Finally, S_1 and C can also cause confusions.

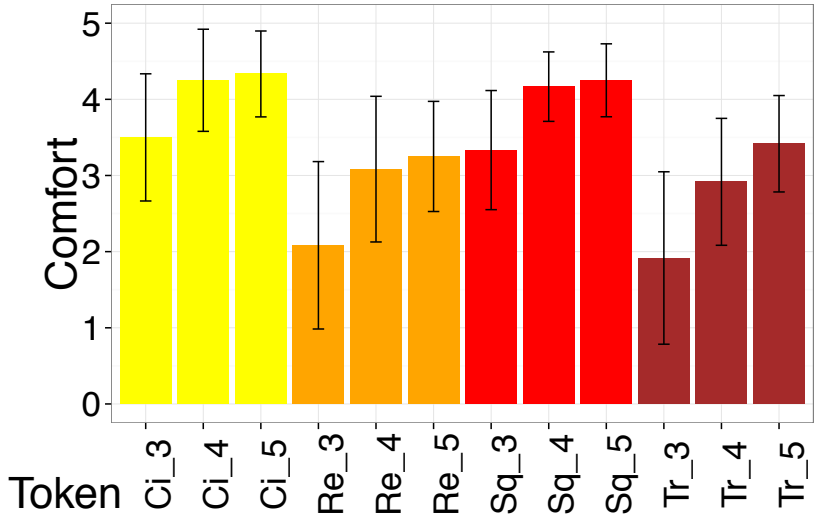


Figure 3.12: Comfort score per token. Error bars represent the 95% confidence interval.

3.5 New Set of Token

Our foundational hypothesis was that physical tokens constrain users' grasp in a consistent manner, which leads to consistent touch patterns that can be recognized with a high level of accuracy. The results of our formative experiment revealed that our hypothesis was only verified for some participants. We also observed significant variations in grasp strategies among users, which means that a set of tokens that works well for one user might not work so well for another user. As we aim at devising a solution that works effectively for all users in a consistent manner, we investigated a solution to decrease the different sources of variability.

We designed a new set of tokens, illustrated in Figure 3.13, similar to those considered in the formative study, but that feature notches. The purpose of these notches is to afford a particular grasp strategy, i.e., to suggest a specific way of positioning the fingers to grab a given token. The design of these tokens was guided by the following requirements. We wanted the token set to feature a wide range of shapes, as tokens should be easy to identify by visual and tactual perception [141]. Sets that feature different shapes also provide better mnemonic cues, making it easier for users to remember token-command associations. Finally, the tokens should remain comfortable to grasp. Based on these requirements, we picked the most comfortable size for each shape (5cm), and added the circular and square tokens of 4cm, which were also rated as very comfortable (Figure 3.12). We limited our summative study to this set of six tokens which, together

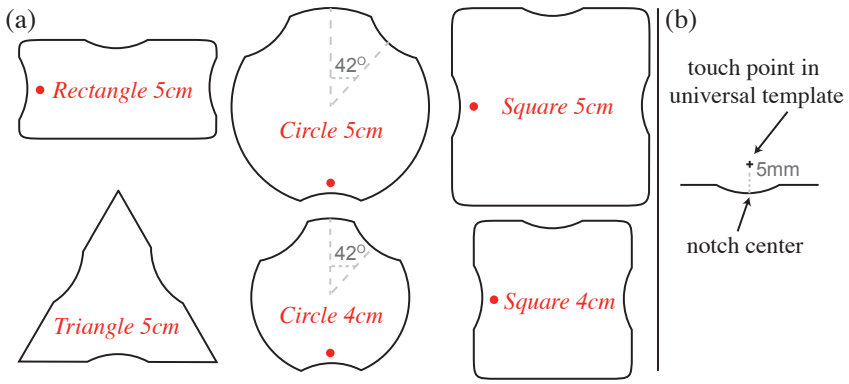


Figure 3.13: (a) The 6 tokens with notches. (b) The touch point's location in the template is offset by 5mm along the normal to the token's edge.

with all token manipulation gestures, already provides a rich input vocabulary.

The grasp strategies observed during our formative experiment (Figure 3.11) informed the positioning of notches on token shapes. The notches' dimensions were refined through trial and error: narrow and deep notches introduce corners under finger tips, which make them uncomfortable; large and shallow notches are more comfortable, but introduce tangential variability in finger position. Our final design tries to strike a balance, and consists of notches 15mm wide and 1.5mm deep. Tokens whose shape afforded variable grasp strategies in the previous experiment feature a dot that indicates where to put the thumb, as illustrated in Figure 3.13-a.

These new tokens are designed to strongly constrain how users grasp them. We hypothesize that this will result in significantly reduced level of variability, which should enable our approach to work without any training. For each token, we compute a representative touch pattern that will act as a *universal template* for all users. The touch pattern is derived from the notches' position, slightly offset from the token's edge along the normal to that edge, so as to better capture users' grasp (Figure 3.13-b). The exact value of this offset (5mm) is calculated from the average offset measured in trials performed with circular tokens in the previous experiment, comparing the radius of the circle that passes through the three touch points with the radius of the actual physical token. The precise vector-based description of these tokens, ready for laser-cutting or 3D printing, is available at [1].

3.6 Summative Experiment

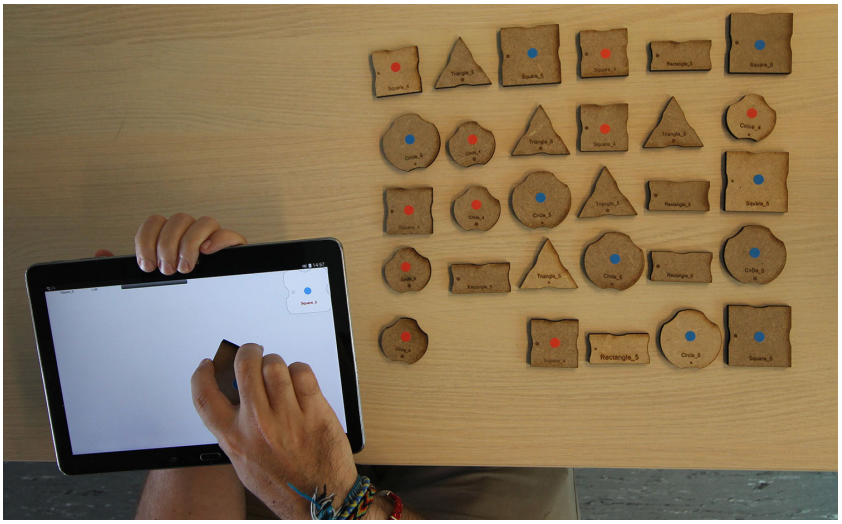


Figure 3.14: Experimental setup in the *Tablet* condition.

We ran a controlled experiment to test users' ability to manipulate tokens with notches, and to evaluate our algorithm's accuracy when provided with the above-mentioned universal templates in combination with this particular kind of tokens. The experimental design is similar to that of the previous study, but uses the set of 6 tokens of Figure 3.13. We also include an additional DEVICE condition: participants perform the tasks on both the tabletop and a tablet. Because of the smaller size of the tablet, we exclude the *Local* condition when DEVICE=*Tablet*, as the different locations (Figure 3.6) are clearly too close to one another to impact users' grasp. Contrary to our formative experiment, participants did not receive any other instructions than to grasp the tokens using the notches. In particular, the operator never asked them to adopt a consistent grasp across trials for a given token.

3.6.1 Experiment Design

Half of the participants started with the *Tabletop*, while the other half started with the *Tablet*. The strategy for counterbalancing the presentation order of trials is exactly the same as in the first experiment. The only difference lies in the *Tablet* condition, in which participants only performed *Global* and *Path* tasks (in this order), but not the *Local* task.

In the *Tabletop* condition, we collected 12 participants × 6 TOKEN × (5 [*Global*] + 2 × 5 LOCATION [*Local*] + 2 × 6 GESTURE [*Path*]) = 1944 trials. In the *Tabletop* condition, we

collected 12 participants * 6 TOKEN × (5 [*Global*] + 2 × 6 GESTURE [*Path*]) = 1224 trials.

3.6.2 Participants & Apparatus

12 volunteers (3 female), aged 23 to 39 years-old (average 26.4, median 24.5), one left-handed, participated in this experiment. Five of them had participated in the previous study. The experimental setup for the *Tabletop* condition was exactly the same as in the previous experiment. In the *Tablet* condition, participants were seated at the same table, but had to hold the tablet during the whole experiment, as illustrated in Figure 3.14. The tablet (Samsung GT-P5110 Galaxy Tab 2) had a 256.7 x 175.3 mm display area with a resolution of 1280 x 800 pixels.

3.6.3 Results

As illustrated in Figure 3.15, the recognition rate in both DEVICE conditions is very high: 98.7% on the *Tabletop* and 99.3% on the *Tablet*. A χ^2 analysis reveals that the effect of INTERACTION on RECOGNITION RATE is significant neither in the *Tabletop* condition ($p = 0.8$) nor in the *Tablet* condition ($p = 0.3$). However, TOKEN has a significant effect in both DEVICE conditions (*Tabletop*: $\chi^2(5, N = 1944) = 30, p < 0.001, \phi = 0.12$ and *Tablet*: $\chi^2(5, N = 1224) = 30, p < 0.001, \phi = 0.16$)). Actually, in the *Tabletop* condition, the RECOGNITION RATE is a bit lower for $Circle_4$ (95.6%) than it is for all other tokens (> 98.7%). The same is true for token $Square_4$ (96.5%) in comparison with all other tokens (> 99%) in the *Tablet* condition.

Interestingly, even if we realized a posteriori that the thumb marker (dot) is meant for right-handed users, our left-handed participant did not have any trouble manipulating the tokens. He simply put his thumb in the notch opposite to the dot, ignoring the latter. Of course, he was able to do so because our tokens feature an axis of symmetry. However, we expect that TOUCHTOKENS's approach can be used for arbitrary-shape tokens, including some that would not feature a symmetric touch pattern. In that case, users can still flip them to accommodate their handedness, provided that the tokens are flat. If a token cannot be flipped easily, a solution would consist in designing two variants: same shape but pattern of notches mirrored. When the pattern cannot be mirrored because of the shape's geometry, it is still possible to design two patterns, one for each handedness.

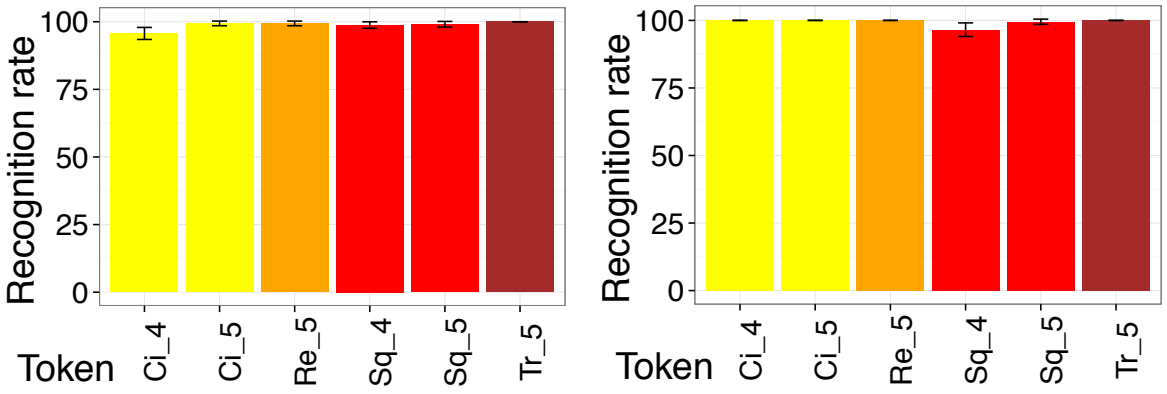


Figure 3.15: Recognition rate per TOKEN in the *Tabletop* (left) and *Tablet* (right) conditions. Error bars represent the 95% confidence interval.

Our algorithm is fast, robust, and easy to implement. It also features the best recognition rate among all alternatives that we implemented and tested on the data collected during our formative study.

Alternative approaches we considered led to significantly poorer performance. In particular, we tested k-Nearest-Neighbour (k=1 and k=3) and SVM algorithms, using both raw data and describing features. The raw data was pre-processed to make it independent from rotation angle and finger identification. The describing features we considered included the touch envelope's area, as well as various descriptive statistics (min, max, mean, median and standard deviation) for measures such as point-centroid distance, distance between successive points, distance between any pair of points, etc. These machine learning approaches yielded recognition rates ranging from 50% to 85% per participant. Compared to this, the analytical approach detailed in this paper, which consists in aligning touch patterns using their centroid as a reference point, works much better.

We also considered using as a reference point the center of the best-fit circle (i.e. the circle that passes through three touch points while minimizing the distance to all remaining points) rather than the centroid, but results were slightly worse. The recognition rate was lowered by about 3% on average.

3.7 Fabrication

TOUCHTOKENS require neither embedding electronics in the tokens nor augmenting the tactile surface with additional hardware (such as, e.g., a computer vision system), which makes setup easy. Tokens can be built from any non-conductive material such as wood, plastic, metal or glass, since the system only relies on the fingers' relative position, which is already provided by the tactile surface.

This flexibility allows designers to easily prototype and test different TOUCHTOKENS variants with a 3D printer or a laser cutter. In particular, designers have a lot of control on the tokens' appearance. For tokens that have permanent roles associated with them, interface designers can engrave an icon or a label on them, or use a specific color. For temporary associations, end-users could adopt more volatile solutions, such as adding stickers or writing with an erasable pen if the chosen material affords it (e.g., pencil on a wooden token). Tokens can also be made of transparent material such as glass or acrylic, to avoid occluding the content displayed on the tactile surface.

A PDF file description that contains the vector shapes for the six tokens ready for laser-cutting (see Figure 3.16) is available at [1].

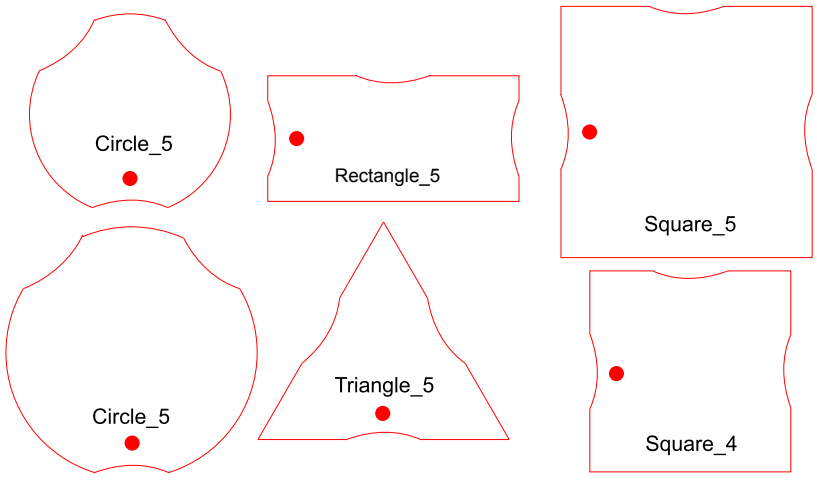


Figure 3.16: Document for printing or laser cutting.

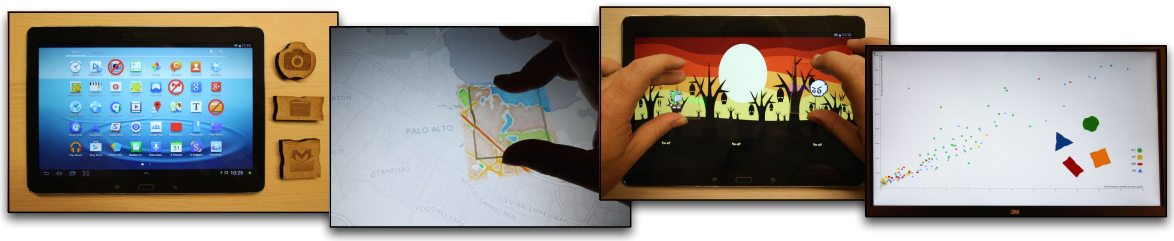


Figure 3.17: Proof-of-concept applications: access control, tangible magic lenses, character controllers in a game, data visualization.

3.8 Application Domain

The above results show that, using tokens with notches, it is possible to build robust applications that will take advantage of both gesture-based and tangible interaction. Application domains that would benefit from such type of input are quite varied and have already been discussed in the literature, including: geographical information systems [142], database querying [120, 143], information management [90, 144] and music composition [107]. We developed a set of proof-of-concept applications to illustrate the different roles that TOUCHTOKENS can play in an interactive system (see Figure 3.17).

TOUCHTOKENS can act as *controllers* or *filters*, and can be used to manipulate both the content of an application or the presentation of this content. For instance, they can be used to adjust the parameters of a visualization, enabling users to focus more on the result of their actions as the manipulation of physical tokens decreases the demand on visual attention [143]. We have developed a simple scatterplot visualization in D3 [145] to illustrate this idea. The different categories in the data (e.g., countries grouped by continent) are associated with different symbols (which have distinct shapes and colors), as is typically the case when visualizing multi-variate datasets. One TOUCHTOKEN, with matching shape and color, is associated with each category and can be used to adjust the visual representation of the corresponding data points in the scatterplot: changing their size by rotating the token, and their opacity by sliding it.

TOUCHTOKENS can be transparent, in which case they will typically be used as physical see-through tools [146, 147], altering the content that falls below the token (e.g., filtering) or changing its visual attributes (e.g., rendering). For instance, we have developed a simple mapping application in which tokens are associated with different layers. The tokens act as magic lenses [146] that reveal the corresponding layer while

leaving the context untouched. See-through tools can also be used to move content in the workspace, as demonstrated in our simple game, where transparent tokens control the location and orientation of individual characters.

TOUCHTOKENS can also act as a receptacle for, or tangible representative of, digital content. Tokens then give access to the associated content [148]. One of our demo application illustrates how TOUCHTOKENS can be used for access control. Tokens can be used, e.g., to launch applications whose icons are otherwise invisible or disabled on the tablet's homescreen, enabling the device to be shared with family (parental control) and friends with some restrictions. Access to private content can be made even more secure by requiring that the token be put in a specific location, or that a particular gesture be performed with it.

Our last application demonstrates the use of TOUCHTOKENS as digital containers. Users can reify photo albums into tokens, and add a picture to an album by holding the corresponding token above it. They can then display an album's content as a grid of thumbnails by rotating the token on the surface, or launch a slideshow by sliding it.

3.9 Summary

In this chapter, we ran a formative experiment to investigate the possibility of recognizing individual tokens by categorizing their associated touch patterns. We were hypothesizing that differences in token shape and size might be sufficient to accurately discriminate those patterns. Our results revealed significant inter-user variability in terms of accuracy: our algorithm can recognize up to ten touch patterns with more than 90% accuracy for some users, while for other users, its accuracy falls down as soon as three or more tokens are in the set. This variability comes from two sources:

1. some users employ different grasp strategies for the same token;
2. some users employ grasp strategies for different tokens that yield very similar touch patterns.

Based on these observations, we then designed a set of six tokens featuring notches aimed at reducing this variability while remaining comfortable to grasp. A summative

experiment showed that with this set of tokens, our recognizer has a minimum accuracy over all participants higher than 95% (avg. 98%), and this without any training. Augmented with notches, TOUCHTOKENS offer a low-cost, yet reliable, solution for enabling tangible interaction on multi-touch surfaces. As mentioned earlier, we make this recognizer freely available, along with vector-based templates for the tokens.

TOUCHTOKENS offer a number of advantages but also have a few limitations. First, they support only basic manipulations: once a token has been recognized, the user can only slide it over the surface. This makes TOUCHTOKENS less expressive than other projects that have proposed tangibles that can be *e.g*, stacked [108] or tilted [92]. Second, we tested a limited set of tokens that feature basic geometrical shapes only, which puts a significant practical limit to how tailored token sets can be. The next chapters address these limitations.

CHAPTER 4

FLEXIBLE TOUCHTOKENS - INCREASING GESTURE VOCABULARY

4.1 Introduction

TOUCHTOKENs provide a simple means to develop tangible interfaces. Manipulating the tokens while maintaining the fingers in contact with the touch-sensitive surface leads to specific multi-touch spatial patterns that can be uniquely identified using a relatively simple software recognizer.

However, the system only tracks a token's position, meaning that users are limited in how they can manipulate tokens. They can only slide tokens on the surface, and they must keep maintaining a token to make the system able to detect its presence. For example, if users have been manipulating a token and they drop it off the surface, the system cannot detect whether the token has been removed or if it is still on the surface. In this chapter, we propose a new token design that allows users to perform more advanced manipulations with the tokens, and that allows the system to detect whether a token has been lifted off a surface or left on it. We achieve this while preserving the original spirit of TOUCHTOKENS, that is without instrumenting the tokens. We hypothesize that when users are manipulating tokens on the surface, the touch traces of fingers' micro-movements carry enough information to make the system able to recognize some token manipulations.

This chapter starts with the description of our approach for increasing the gesture vocabulary with passive tokens on multi-touch surfaces. We detail a new design of our previous set of tokens, based on lattice hinges which can easily be obtained using fabrication processes such as laser cutting. This design makes the tokens flexible, allowing users to squeeze or bend them. We then report on a formative study in which we collected a sample of finger micro-movements that are representative of the different token manipulations. We use this collection of touch traces to design our recognizer, and evaluate its performance. Finally, we present a few demonstrations to illustrate how these events can be used for developing powerful interfaces.

4.2 Functionality Overview

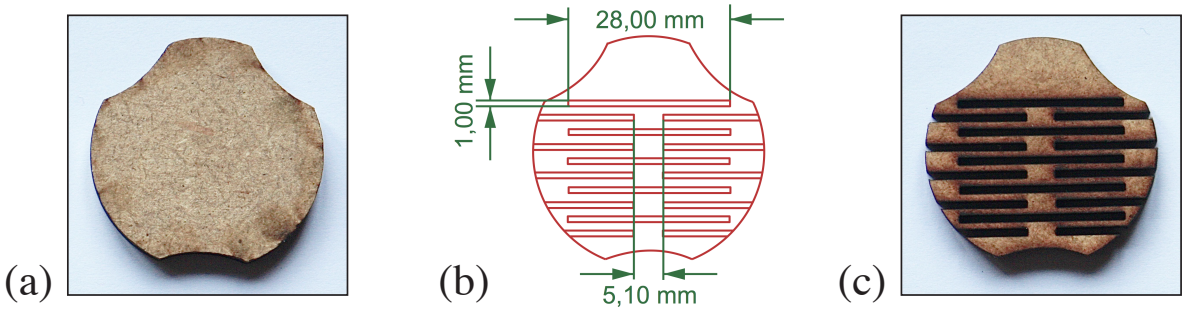


Figure 4.1: Making a TOUCHTOKENS flexible: (a) original, rigid TOUCHTOKENS (circle, 4cm in diameter), (b) schematics of lattice-hinges, (c) flexible TOUCHTOKENS.

In this project, our goal was to make TOUCHTOKENS more expressive than the basic two-state model of touch interaction. We achieve this with two components. First, we propose a new token design by introducing laser-cut lattice hinges that make tokens flexible (Figure 4.1). Second, we design a new recognizer, that analyzes the micro-movements of the fingers that hold the tokens and enables the system to detect three types of manipulation:

1. when a token has been left on or lifted off the surface;

2. when a token is squeezed;

3. when a token has been bent.

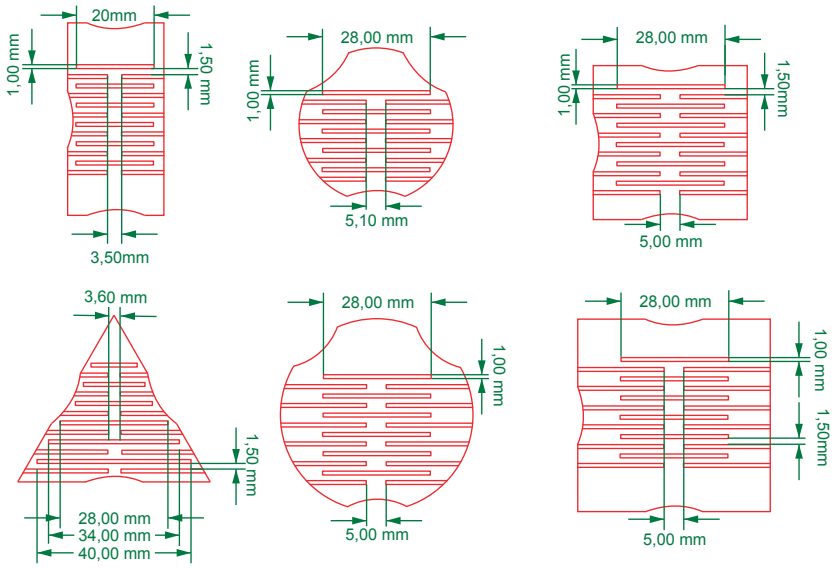


Figure 4.2: Detailed design schematics of our flexible TOUCHTOKENS (tokens are oriented so that the thumb notch is always on the bottom side). We make these vector descriptions available at [1].

4.3 Flexible TouchTokens - New Token Set

Figure 4.2 shows our novel set of tokens, which can be squeezed or bent by pinching them. Laser-cutting lattice hinges is a common method in the maker community to make a piece of wood flexible using laser cutting. In our case, we performed several design iterations so as to make the tokens comfortable to manipulate while ensuring enough robustness. The kerfs' orientation was chosen so as to match that of the comfortable pinch formed by the thumb on one side and the {index, middle} couple of fingers on the other side. The kerfs' width, length and interspacing provide enough elasticity to make the tokens easy to deform without requiring too high a force, while ensuring that they revert to their original shape when not pinched. We also considered resistance to avoid accidental pinches during regular manipulations, and robustness to avoid the risk of breaking.

4.4 Flexible TouchTokens Interaction

We contribute three novel primitives to the interaction vocabulary of TOUCHTOKENS: a state (*on/off*), a quasi-mode (*squeezed*) and a discrete event (*bent*). We achieve this with a novel design that makes the tokens flexible, and with an analysis of the micro-movements

users make when performing these interactions, following an approach similar to the recognizers designed to detect thumb-tip micro-gestures [54, 55]. This section describes our new tokens and introduces our hypotheses regarding the micro-movements we expect to observe.

4.4.1 Detecting Tokens' *on*/*off* State

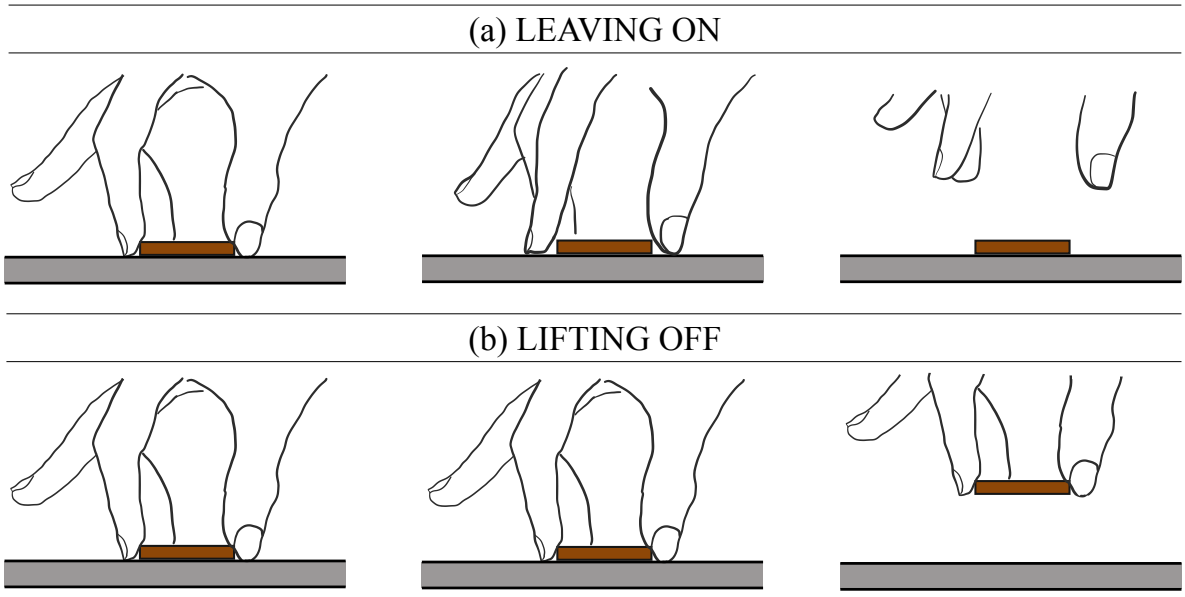


Figure 4.3: Finger micro-movements when leaving a token *on* the surface (a), and when lifting it *off* (b).

Making the system aware of whether a token is still *on* the surface, or if it has been lifted *off* it, is an important feature of tangible interaction. It allows users to lay out several tokens on the surface (as in, *e.g.*, Facet-streams [120]). Conductive tokens usually rely on the fact that the human body is a conductor. They thus become invisible to the system as soon as users no longer touch them. The system does not even know whether a token has been left on the surface or removed off it.

TOUCHTOKENS require users to both hold them by putting their fingers in the notches and touch the surface with those fingers. We hypothesized that the micro-movements made by the fingers at the time they leave the surface would have a distinct signature, depending on whether users were leaving tokens on the surface or were lifting them off. Figure 4.3 illustrates our hypothesis: when leaving a token on the surface, users are likely going to relax their grasp, while when lifting it off, they will likely maintain a

firm grip, potentially compressing the token a bit. In the former case, we should observe finger traces that move slightly away from the touch points' centroid. In the latter case, we should observe finger traces that either remain still or move slightly toward the touch points' centroid.

4.4.2 Squeezing Tokens

When squeezing a token, the user's fingers remain in contact with the surface throughout the corresponding micro-movements.

We hypothesized that when squeezing, we would observe touch traces that move toward the touch points' centroid, and away from it when un-squeezing. If successful, tokens can then be made to behave like a mouse with a button: quickly squeezing and releasing is equivalent to a click; keeping the token squeezed and moving it on the surface is equivalent to a drag. These can be used respectively to trigger discrete events, and to enter quasi-modes.

4.4.3 Bending Tokens

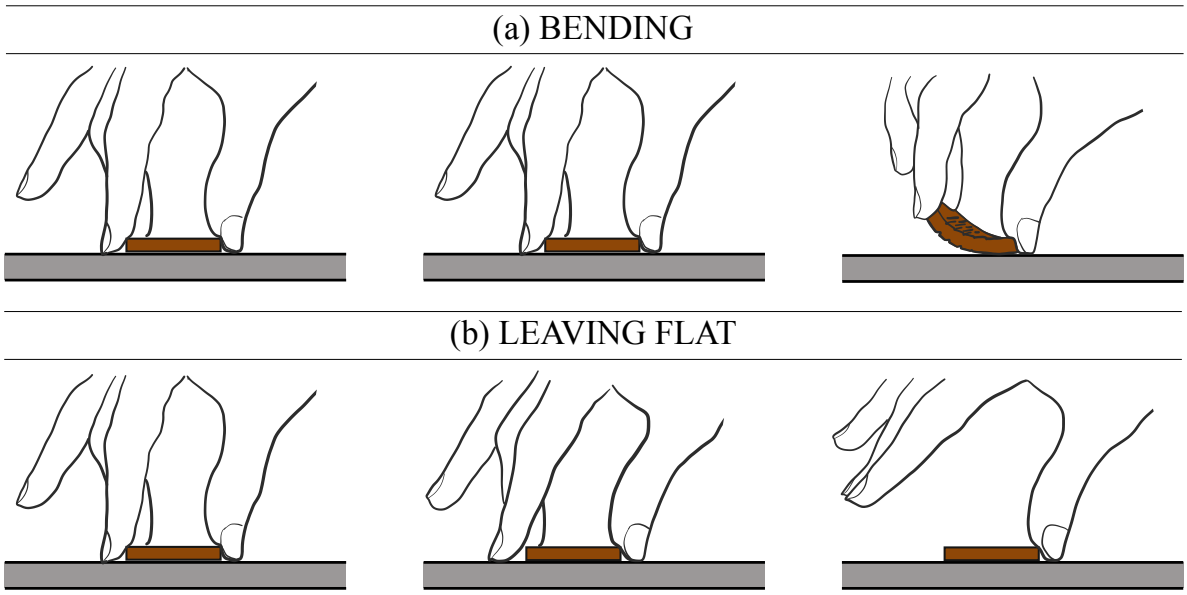


Figure 4.4: Micro-movements when (a) bending a token, (b) leaving it flat.

Bending a token leads to a state where users are keeping only one finger in contact with the surface (Figure 4.4-a). As all other token manipulations involve at least two fingers, the number of fingers could be a discriminating factor. However, it is too permissive,

as it may also match cases where users lift two fingers off, but leave the token flat on the surface (Figure 4.4-b). Again, micro-movements may help us detect actual bending actions. We hypothesize that users are likely going to keep their index and middle fingers in contact with the token's side when bending it, while they are going to relax their grip when leaving it flat. We should thus observe still traces before lift-off when bending, as opposed to traces that slightly move away from the centroid in the other case.

4.5 Collecting Touch Traces

We collected multi-touch traces of users performing the three types of manipulations described above. Our goal was to gather data about the different finger micro-movements, and to identify criteria that could enable us to recognize the corresponding manipulation events. We were particularly interested in the typical profile of point-to-centroid average distance time-series associated with these movements.

4.5.1 Participants & Apparatus

Twelve volunteers (2 female), 23 to 40 year-old (avg. 28.83, med. 28), participated in the data collection. They were seated at a desk, manipulating tokens on a tablet (Samsung SM-T810 Galaxy Tab S2: 237 × 169 mm display area / 2048 × 1536 pixels), laid flat on the desk. Participants were video-recorded.

4.5.2 Procedure

All participants performed the 3 manipulation events: *Click and Drag & Drop*, *Leave on vs. Lift off* and *Bend vs. Leave flat*. Presentation order was counterbalanced using a Latin Square. All events involved the flexible version of the 6 TOKENS introduced in the chapter 3: 2 circles, 2 squares, 1 triangle, 1 rectangle.

$Event_1$: *Click and Drag & Drop*

Participants had to perform 2 types of ACTIONS: *Click* or *Drag*. In the *Click* case, they had to grab the right token using 3 fingers, put it on a black cross, and then slide it toward a red circle located 130 mm away. Once the token was inside the circle, they had to perform a "click" on the token by compressing it sideways, and then release the pressure. Finally, they removed the token from the surface. In the *Drag* case, they had to:

compress the token right after having put it on the black cross, keep it compressed while moving it toward the red circle, and release the pressure before removing the token from the surface. We collected data involving sliding movements in 4 main DIRECTIONS: up, down, left, right. The tablet was placed in landscape mode for DIRECTION = {left, right}, and portrait mode for DIRECTION = {up, down}, so that the red circle would be at the same distance from the black cross in all conditions.

$Event_2$: *Leave on vs. Lift off*

Participants also had to move a token from a black cross to a red circle. However, once in the circle, participants had to perform one of two ACTIONS: *Leave on* or *Lift off*. In the first case, they had to lift their fingers off the surface but leave the token on it. In the second case, they had to lift their fingers, taking the token off the surface. We used the same 4 DIRECTIONS as in $Event_1$. We introduced an additional factor, FINGERCOUNT, to capture the two different manipulation styles described in section 3.1 in chapter 3: once a token has been identified with the 3-finger hold, users can keep manipulating it with 3 fingers, or they can relax their grasp and manipulate the token with only 2 fingers. Thus, we had 2 FINGERCOUNT conditions: participants either had to keep their 3 fingers in contact with the surface all along (*3-finger* condition), or they were asked to lift a finger off the surface after having put the token on the black cross, and to keep it lifted until the end of the trial (*2-finger* condition). Failure to comply in any given trial meant it had to be performed again.

$Event_2$: *Leave on vs. Lift off*

The tablet only displayed a black cross. Participants had to put the right token on the surface and perform one of two ACTIONS. In the *Bend* condition, they had to bend the token, keeping only their thumb in contact with the surface, and then unbend the token by putting the other two fingers back on the surface. In the *LeaveFlat* condition, they also had to lift two fingers off the tablet, only keeping the thumb in contact, but without bending the token, which remained flat on the tablet. They then had to put their two fingers back on the surface to end the trial.

For each event type, trials are first blocked by ACTION, then by DIRECTION within

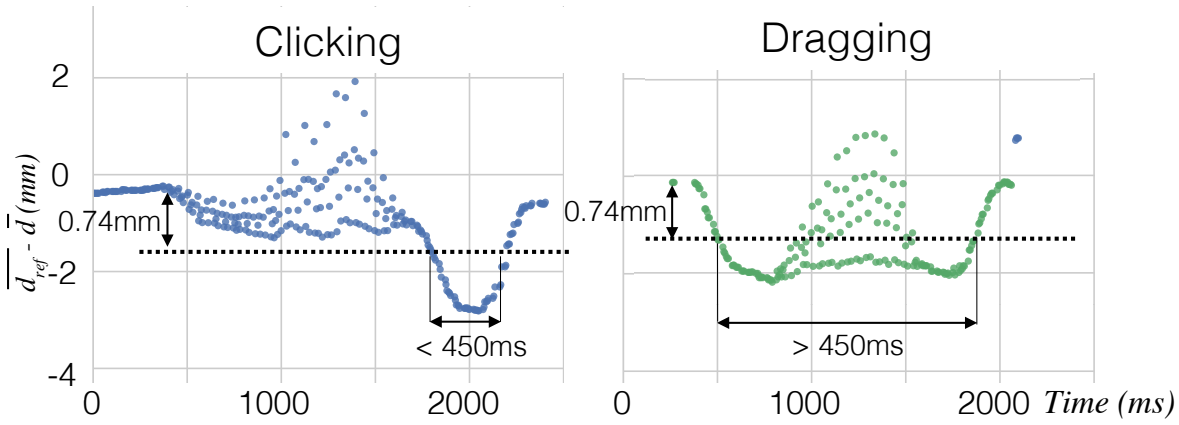


Figure 4.5: Using *Squeeze* mode for clicking (left) and dragging (right).

each ACTION ($Event_1$ and $Event_2$), and by FINGERCOUNT within each DIRECTION block ($Event_2$). Each condition is replicated 3 times. Block presentation order is counterbalanced across participants; trial presentation order within a block is random. The whole procedure consists of 252 trials (72 + 144 + 36), and lasts approximately one hour.

Finally, some indications about the robustness of our flexible token design: we used the same set of tokens throughout the entire experiment, that consisted of 3024 manipulations by 12 people (252 x 12). No token was broken, or deformed.

4.6 Recognizers

Our main hypothesis was that the micro-movements of interest to us could be observed by looking at the fingers' traces, that should move slightly toward, or away from, the token's center. To verify this, we analyzed, for all collected touch traces, the evolution over time of the average distance $\overline{d}$ of a touch point to the centroid of the corresponding multi-touch sample. In the following, we report the criteria we identified as the most successful for capturing these micro-movements. Parameter values (**in bold**) are determined in the next section.

1. *Squeeze:* a token is considered *squeezed* (Figure 4.5) when:

$$\forall i \in \{1..|B|\}, \quad \overline{d_{ref}} - \overline{d_i} > \mathbf{d_{sqz}}$$

where $\overline{d_{ref}}$ is the average distance in millimeters of a touch point to the centroid of the corresponding multi-touch sample when users register the token, and B is a buffer containing the successive values of $\overline{d}$ over the last $\mathbf{buffer_{sqz}}$ milliseconds.

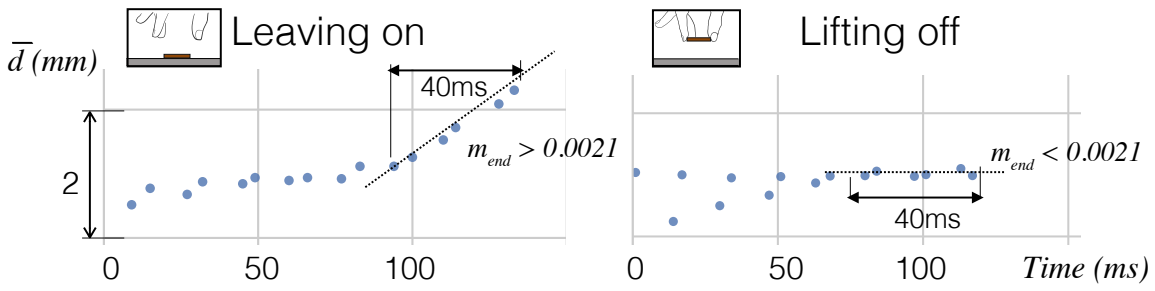


Figure 4.6: Leaving a token *on* the surface (left) or lifting it *off* (right).

2. *On/Off:* a token is considered as left *on* the surface when:

$$m_{end} > \mathbf{m_{on\_off}}$$

where m_{end} is the slope (it is computed using the Theil-Sen estimator [149]) of the evolution of $\overline{d}$ over the $\mathbf{buffer_{on\_off}}$ milliseconds preceding the instant where the last finger has been lifted off the surface ($count(fingers) = 0$). On the opposite, if $m_{end} \leq 0$ at this instant, the token is considered as lifted *off* the surface. Figure 4.6 illustrates the two cases.

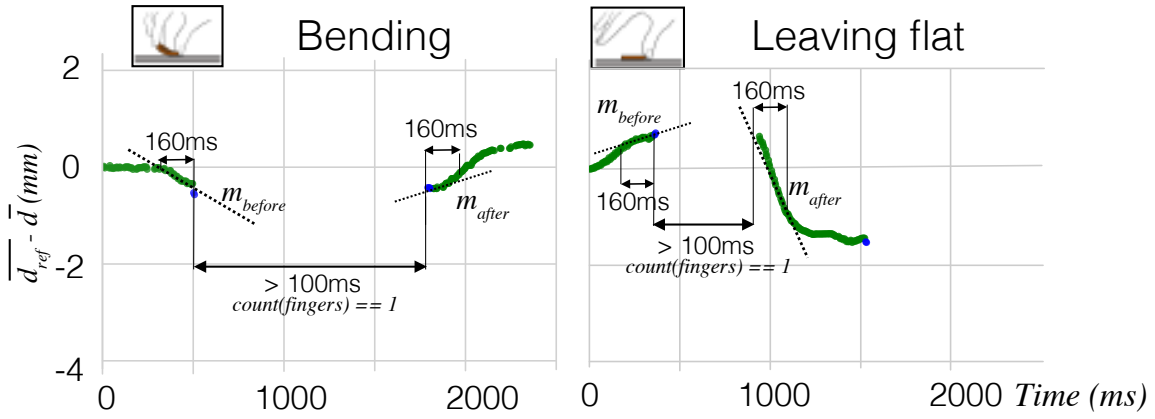


Figure 4.7: *Bending* a token (left) or leaving it flat (right).

3. *Bend:* a token is considered as having been *bent* when:

$$max(m_{before}, -m_{after}) < 0$$

where m_{before} (resp. m_{after}) is the slope of the evolution of $\overline{d}$ over the $\mathbf{buffer_{bend}}$ milliseconds preceding (resp. following) the instant where only one finger remains in contact with the surface ($count(fingers) = 1$) for at least 100ms, as illustrated in

Figure 4.7. The formula is basically a sign analysis: it checks whether $\overline{d}$ increases or decreases before and after the time span during which there is one single contact point. We initially considered analyzing only m_{before} to detect when users enter the *bent* state, but our tests revealed that this sample does not carry enough information to discriminate between *bending* and *leaving flat*. This entails that our recognizer considers *bent* as a discrete event, that gets triggered only once users have unbent the token.

We couple these criteria with state machines that take the number of contact points into account, making it very unlikely that any one event will get confounded with the other two:

- The criterion for *squeeze* is only evaluated when there are 3 contact points on the surface for at least 200ms. This is mainly to avoid confusion with cases where users bend the token, as they tend to compress it when unbending.
- The criterion for *on/off* is only evaluated when the number of contact points becomes null.
- The criterion for *bend* is only evaluated after a time span of 100ms during which there has been exactly 1 contact point.

The Java implementation of FLEXIBLE TOUCHTOKENS with support for both TUIO and Android is available at [1].

4.7 Recognizer Parametrization

For each of our three micro-movements, we measure the *accuracy* of our recognizer by running it on data collected for this micro-movement only. We then test its *robustness* to false positives by running it on data collected for the other two.

We use the leave-one-out cross-validation technique to parameterize the recognizers: for each participant, we set the parameters to values that maximize the overall recognition score for the 11 other participants. We then report the average score across all 12 participants (mean, median, standard dev.).

Squeezed mode is recognized in **96.9%** (median: 97.9 / std: 3.0) of all trials collected for $Event_1$ (with $\mathbf{d_{sqz}} \in [0.74, 0.75]$ and $\mathbf{buffer_{sqz}} = 100$). It is falsely detected in 1.8% of all trials for $Event_2$, and 2.1% for $Event_3$.

States *on* and *off* were properly distinguished in **90.1%** (median: 92.4 / std: 5.1) of all trials for $Event_2$ (with $\mathbf{m_{on\_off}} \in [0.0018, 0.0027]$ and $\mathbf{buffer_{on\_off}} = 40$). The distinction between states *on* and *off* also works well for $Event_3$, with only 7.6% of false positives. However, when tested on trials from $Event_1$, we observe 43% of false positives. A finer analysis reveals that the recognizer fails to detect state *off* right after leaving mode *squeezed*, which happens when users lift the token off while releasing the pressure applied on the token ($\overline{d}$ increases right before $count(fingers) = 0$). Making tokens flexible thus provides opportunities for performing micro-movements in general, but has the side-effect of introducing some ambiguity in this particular case. This is a limitation of our recognizer that we will further investigate. In the meantime, it can be handled by considering the state where $count(fingers) = 0$ right after having left mode *squeezed* as "uncertain", prompting users for input to resolve the ambiguity.

We also tested our *on/off* recognizer on *rigid* tokens during informal tests. We observed a recognition accuracy close to 90% suggesting that these micro-movements can also be detected on regular TOUCHTOKENS.

For $Event_3$, *Bent* events were detected in **91.1%** (median: 91.7 / std: 6.1) of all trials where ACTION = *Bend* (with $\mathbf{buffer_{bend}} \in [100, 160]$). In the remaining 8.9% trials, the recognizer detected either 0 or at least 2 *Bent* events (during the same trial). No *Bent* event is ever accidentally triggered for either $Event_1$ or $Event_2$, as the time intervals during which users have only one finger in contact with the surface are infrequent and very short. No *Bent* event is ever accidentally triggered, either, when ACTION = *LeaveFlat*.

4.8 Example Applications

TOUCHTOKENS allow developers to create various applications with tangibles. For example, developers can use TOUCHTOKENS for controlling parameters or filtering data in a visualization or they can be used as controllers in games. Our new vocabulary of events enables the development of even more powerful interfaces where tokens can be dragged (*squeeze*) or clicked (*bent*, *squeezed*), and where several tokens can be laid on the surface (*on*/*off* enabling the system to keep track of untouched tokens).

To demonstrate the practical feasibility and the potential of our new token design and its associated recognizer, we developed two application examples.

Crocodile Game

(a) User is squeezing the token for moving the crocodile

(b) User is performing the bend gesture

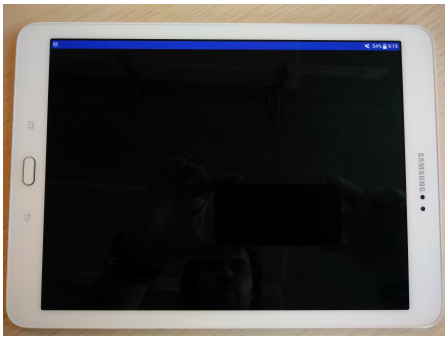
(c) When the user put the token back to the surface the recognizer detects a bent event, which activates the action *open mouth*

Figure 4.8: Crocodile interactive board game: a crocodile moves in the river in order to eat ducks while avoiding rocks.

To demonstrate the use of *On / Off*, *Squeeze* and *Bend* interactions, we implemented a crocodile game, inspired by the *hungry hungry hippos* game. A token acts as a controller for a crocodile that must eat ducks but avoid rocks. Users can move the crocodile by squeezing and moving the token around the surface, and they can open the crocodile's mouth by bending the token. They can also remove the token from the surface to hide the crocodile when there are too many rocks.

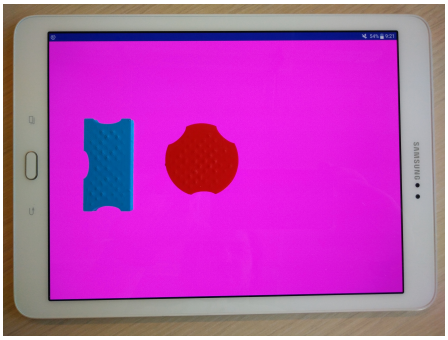
Mixer Color Game

In this application, we illustrate how tokens can be used to access and modify digital data. We especially illustrate how the *on / off state* can be useful for combining the effect of multiple tokens. The application is a tangible game, inspired by kid games for learning. We printed three different shape tokens where each token has a different primary color (circle is red, rectangle is blue and square is yellow). Users can play by placing and removing tokens on the tablet surface. Putting the token on the surface alters the color of the application background by making the color of the token mixed with the current color of the background. The background color remains unchanged when users are leaving the token on the surface. With this game, users can test different color combinations by *adding*, *leaving* and *removing* color tokens.

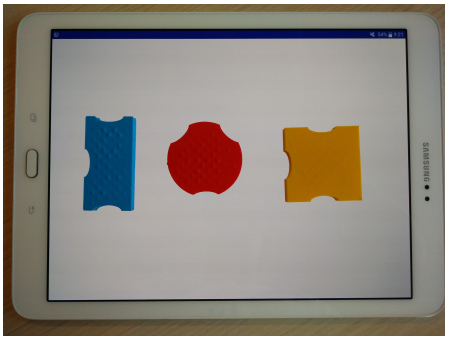


(a) Black background: no token.

(b) The user puts a blue token. The background turns blue.



(c) The user adds a red token. The background turns to purple, *i.e.*, the result from the mixing of red and blue.



(d) The user has placed all tokens on the surface. The background turns white, *i.e.*, the result from the mixing of all primary colors.

Figure 4.9: Interactive board game: users play mixing colors with passive tokens on the surface.

4.9 Summary

In this chapter, we have investigated an approach for increasing the vocabulary of interaction with passive tokens on capacitive surfaces. First, we have designed a new set of flexible tokens. Second, we have analyzed the finger traces when users are manipulating those new passive tokens. We hypothesized that there might be sufficient information to discriminate patterns for recognizing the following three types of manipulations:

1. *On* / *Off*: Users leave the token on the surface, or they lift it off.

2. *Squeeze*: Users compress the token keeping their fingers in contact with the surface. We classified this manipulation as click or drag depending on the duration the

token is kept in the squeeze state.

3. *Bend*: Users bend the token with index and middle fingers, keeping only one finger in contact with the surface for a time span, and then they put it back (horizontal) on the surface.

Our results showed that our algorithms can recognize these three manipulations with an accuracy of: *On/Off* 90.1%, *Squeezed* 96,9% and *Bent* 91.1%. Such high rates allow developers to use these manipulations for developing advanced applications. We presented a couple of proof-of-concept applications to illustrate the different roles that these manipulations can play.

CHAPTER 5

TOUCHTOKENBUILDER AND TOUCHTOKENTRACKER - INCREASING TANGIBLE SET

5.1 Introduction

In chapter 3, we introduced TOUCHTOKENS as a means to design low-cost tangible interfaces. The approach consists in fabricating passive tokens that can be made of any non-conductive material. However, designing passive tokens remains a rather complex process if we want to end with tokens that are both comfortable to grasp, and that minimize recognition conflicts.

Besides, as we discussed in section 2.3, tangible interfaces have been used in various domains such as music composition, collaboration, games or education among others. All of these interfaces feature physical tokens that aim at resembling actual objects from the targeted application area. Researchers have reported that variations in shape, size and material of tokens all play an important role in providing the right manipulation affordances and conveying the proper semantics. While it is quite straightforward to create variations in size, it is less easy to build arbitrarily-shaped tokens.

In this chapter, we present two tools that allow interface designers to build and recognize TOUCHTOKENS that feature arbitrary shapes. First, we introduce TOUCHTOKEN-BUILDER which helps interface designers to build tangibles based on the TOUCHTOKENS'

approach of placing notches in passive material. Second, we present TOUCHTOKENTRACKER which works with an output file from TOUCHTOKENBUILDER in order to provide developers with more information for tracking the tangible. We then present some proof-of-concept token sets designed with TOUCHTOKENBUILDER, and report on experiments that we conducted to evaluate TOUCHTOKENTRACKER's recognition accuracy for these token sets. Finally, we discuss the limitations of our approach and directions for future work.

5.2 Functionality Overview

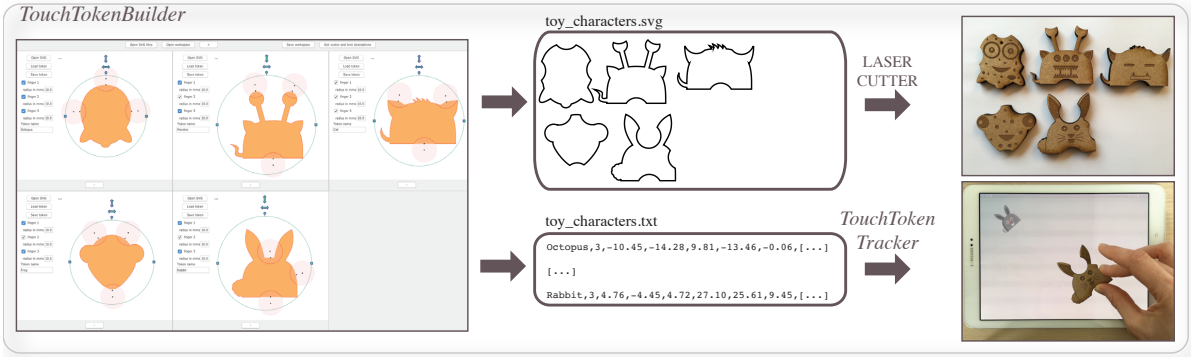


Figure 5.1: TOUCHTOKENTRACKER (left) assists users in placing grasping notches on arbitrarily-shaped tokens, warning them about spatial configurations that are too similar and could generate recognition conflicts. TOUCHTOKENBUILDER outputs both a vector and a numerical description of the tokens' geometry (middle). Those are used respectively to build the tokens (top-right), and to track them on any touchscreen using TOUCHTOKENTRACKER (bottom-right).

The objective of this project is to provide tools that enables the creation of arbitrarily-shaped TOUCHTOKENS.

Our first contribution, TOUCHTOKENBUILDER, is a software application that assists interface designers in placing notches on arbitrarily-shaped vector contours for creating conflict-free token sets. The application features a simple direct-manipulation interface and outputs two files: a vector-graphics description of all tokens in the set, ready to be fabricated using, *e.g.*, a laser cutter; and a numerical description of the geometry of each token.

Our second contribution, TOUCHTOKENTRACKER, is a software library that takes as input the numerical description produced by TOUCHTOKENBUILDER. While TOUCHTOKENS' original algorithm only provided developers with the ID of the recognized token and the user's finger coordinates, the new TOUCHTOKENTRACKER also enables tracking the tokens' full geometry (location, orientation and shape) throughout their manipulation on the multi-touch surface. In addition, tracking remains robust even when users lift a finger while manipulating tokens (leaving a minimum of two fingers in contact with the surface), as illustrated in Figure5.1 .

5.3 TouchTokenBuilder

TOUCHTOKENS feature three notches that suggest to users how those tokens should be grasped so as to enable effective recognition of those tokens by the system. The recognition algorithm introduced in chapter 3 only needs one unique template per token, called *universal template*. This template consists of a series of three coordinates that correspond to the expected finger contact point coordinates relative to the token's center. These simple templates have been demonstrated in chapter 3 to be sufficient to achieve a recognition accuracy of ~98% with the set of basic TOUCHTOKENS.

TOUCHTOKENS' approach is simple, but it requires designers to compute the coordinates of the templates' points (feeding the recognizer), and to specify the geometry of each token's contour with some vector-drawing tool (for fabricating the tokens), carving them accordingly to create the notches. As we commented previously, this can be a tedious process.

This section introduces TOUCHTOKENBUILDER, an application that makes the token design process easier. Building a token now simply consists of importing an SVG image, from which the token's outline will be derived, and positioning the three notches on that contour by dragging three circles that represent the user's finger tips.

5.3.1 Designing arbitrarily-shaped tokens

Figure 5.1 illustrates the general approach that a designer can follow when creating a set of tokens, in this case for a game where toy characters (octopus, monster, cat, frog and rabbit) are controlled with tangible tokens. He first identifies a set of SVG images he wants to use for the different tokens. In our scenario, those simply get

downloaded from the Web. In this particular example: `http://www.clipartlord.com/category/halloween-clip-art/monsters-clip-art/`, he then loads them in TOUCHTOKENBUILDER. For each SVG image, TOUCHTOKENBUILDER computes the outline of the entire geometry and creates a new cell in which it displays that outline, as illustrated in Figure 5.2. The outline is computed by processing all SVG elements in the image with the help of the Batik toolkit:[1] each element is turned into a Java2D shape, taking into account groupings and affine transforms; the union of all those shapes is computed; and then the outline is generated by marching along the contour of the resulting shape.

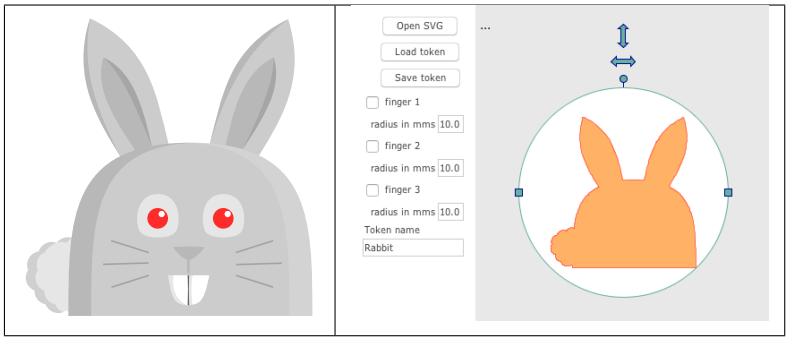


Figure 5.2: SVG image for the rabbit toy character (left) and the corresponding outline displayed in TOUCHTOKENBUILDER(right).

Each token outline can be manipulated using simple widgets to adjust its scale and orientation. As shown in Figure 5.2, a ring surrounds the token, featuring two square handles to resize the token, and a circular handle to rotate it. Two arrows, positioned above, let users flip the token vertically or horizontally. A panel on the left-hand side of each token cell enables users to position the three finger notches on the outline. Fingers are represented using semi-transparent red circles (Figure 5.1). Each of these circles can be dragged and resized, and acts as a carving tool: when a circle intersects the token's outline, it actually subtracts the intersecting area from the token, computes the corresponding universal-template point (*i.e.*, the estimated finger contact point), and detects potential sources of conflicts between tokens, as detailed later.

TOUCHTOKENBUILDER adapts each token's display size depending on screen resolution, so that it matches its actual physical size when fabricated. This helps designers informally evaluate how comfortable a given token is to grasp, by putting their fingers on the corresponding circles on screen. The SVG vector description output by TOUCHTOKENBUILDER declares the document size and view box parameters so that the coordinates

[1] `http://xmlgraphics.apache.org/batik/` (visited 2016-03-18)

are correctly interpreted by the fabrication device that will be used to make the tokens such as, *e.g.*, a laser cutter.

Once satisfied with his set, the designer can export the corresponding vector and numerical descriptions (Figure 5.1-middle). TOUCHTOKENBUILDER turns what was a tedious process (relying on vector graphics editing and geometrical computations) into a sequence of simple direct manipulations. It does not require users to manually draw or extract the token's contour. Most importantly, it enables users to very easily test alternative placements for the finger notches, as the computation of the carved contour is now fully automatic. Token design is achieved through a very simple interaction model, based exclusively on drag-and-drop, that avoids premature commitment [150], making the design process much more flexible. To further facilitate exploratory design by trial-and-error, TOUCHTOKENBUILDER also supports a per-object history of actions [151], enabling users to revert any graphical object such as, *e.g.*, a finger circle or a token manipulation handle, to one of its earlier positions.

5.3.2 Detecting conflicts

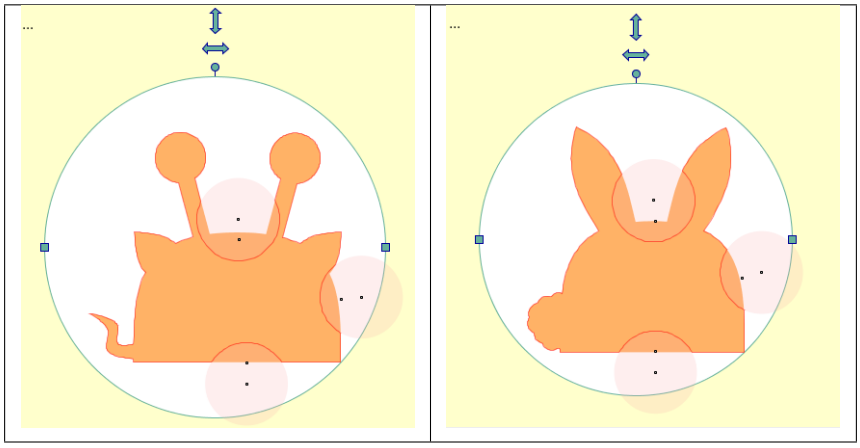


Figure 5.3: Two conflicting tokens.

As described above, TOUCHTOKENBUILDER makes it easy for designers to test different positions for the three notches that must be carved in each token. Finding correct positions for notches is not solely a question of comfort and aesthetics (avoid altering the original shape too much), however. It also involves preventing recognition conflicts between tokens in the set. To this end, TOUCHTOKENBUILDER provides immediate visual feedback when it detects conflicts, turning the corresponding tokens' background yellow (Figure 5.3). Addressing such conflicts can be achieved by moving one or more

notches along the contour, or adjusting the token's size, thereby causing the notches to move closer or farther away from the token's centroid. The token cell's background turns back to white as soon as the conflict has been resolved.

Warnings regarding potential conflicts are based on a heuristic derived from data collected in the second experiment reported in chapter 3. For all three notches of each trial, we computed the distance between the actual touch point and the template point $P_{template}$, located 5mm away along the normal at the notch's center (Figure 3.13).

Figure 5.4 summarizes the results: in ~98% of cases, this distance is less than 5mm for all three notches. Based on these observations, we define the tolerance area of a notch as a 5-mm radius circle around its corresponding template point. Two tokens are thus likely to cause confusion if they can accommodate the same multi-touch input within their respective tolerance areas.

TOUCHTOKENBUILDER relies on this notion of tolerance area to check each token pair (T_1, T_2) for conflicts by proceeding as follows. The first point of $template(T_2)$ is incrementally adjusted within its tolerance area, each step increases its distance and angle to the template point by 1mm and $\pi/12$, respectively (polar coordinates), and, after each increment, gets aligned with $template(T_1)$, in the same manner as TOUCHTOKENS's recognizer. If, once aligned, the two other points of $template(T_2)$ both fall within the tolerance area of their paired points in T_1, TOUCHTOKENBUILDER detects a conflict and warns the user. TOUCHTOKENBUILDER performs this verification continuously whenever a manipulation handle or a finger circle is moved, in order to provide immediate feedback when resolving conflicts. The method runs at interactive frame rates, but to avoid unnecessary computations, TOUCHTOKENBUILDER first tests if such an elaborate detection is necessary by aligning the two templates and computing the distance between the three pairs of points. If all three distances are greater than 1cm, there is no conflict, and TOUCHTOKENBUILDER does not perform any further check.

5.4 TouchTokenTracker

TOUCHTOKENTRACKER allows developers who make use of arbitrarily-shaped tokens in their application to track the full geometry of those tokens. Distributed as a library, it enables the development of applications that need to display contextual information around or below the token. Examples include information filtering using tangibles as

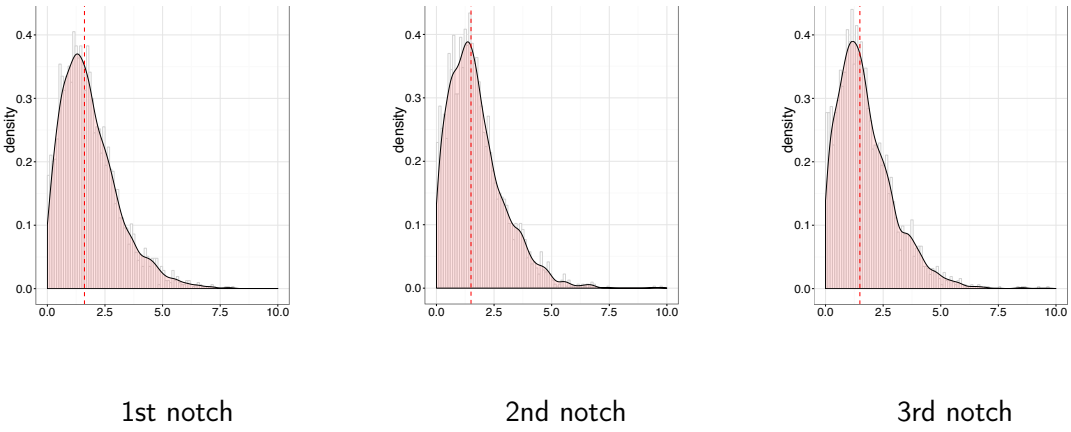
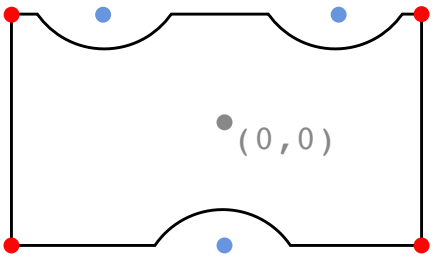


Figure 5.4: Distance (mm) between $P_{template}$ and P_{actual} (template and actual touch points) for all 3 notches. Red dashed lines show median values.




```
rectangle,3,-13.69,-14.87,
14.49,-14.87,0.74,14.74,0,0,
-25.03,-14.11,-25.03,14.11,
25.03,14.11,25.03,-14.11
```

Figure 5.5: The numerical description of a token includes: template points for recognition (blue), token center (gray), and the token's contour as a polyline (red).

see-through tools [147], and games (Figure 5.7-bottom demonstrates launching missiles from a tangible spaceship). TOUCHTOKENTRACKER's recognition algorithm relies on the three points provided in each token template, as the original TOUCHTOKENS recognizer described in section 3.3 already does. It considers two additional pieces of information, provided in the new templates output by TOUCHTOKENBUILDER: the token's center coordinates, and a description of its contour as a polyline (Figure 5.5). These are used to estimate the location and orientation of the token, which are then provided to developers through a simple API.

To recognize tokens, TOUCHTOKENTRACKER identifies the best alignment between the points of each candidate token's template and the actual touch points. Aligning template points with touch points requires translating and rotating template points so as to (1) make the centroids of the touch and template points coincide, and (2) align this centroid with the first pair of matched touch and template points.

As illustrated in Figure 5.6, TOUCHTOKENTRACKER stores the pairing between a touch point and its corresponding template point. It also stores the initial locations of the touch points and the token's initial orientation, which is the rotation angle used to align the first pair of points with the centroid. Using this information, it can estimate the current orientation and location when users move and rotate the token (Figure 5.6-c-right). In case the user lifts a finger off the surface to adopt a 2-finger pinch grasp that facilitates some manipulations (as in Figure 5.7), TOUCHTOKENTRACKER computes a third artificial touch point, assuming that the relative placement between touch points and between the template points are consistent (Figure 5.6-c-left). Keeping track of the three notches' locations can be useful to implement some interactions like the missiles launched by the spaceship in Figure 5.7-bottom.

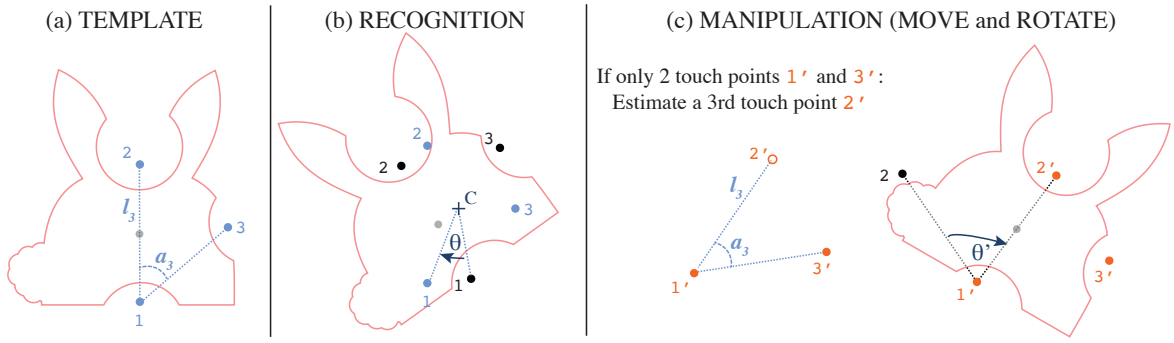


Figure 5.6: Tracking strategy. (a) Token template. (b) Recognizing a token: TOUCHTOKENS's algorithm aligns template points (blue) with touch points (black) by translating template points to make their centroid C coincide with the touch points' own centroid, and by rotating them to make the first pair of matched points aligned with C. TOUCHTOKENTRACKER stores the pairing between touch and template points, the rotation angle θ, and the touch points. (c) If the user lifts one finger off the surface, TOUCHTOKENTRACKER computes a third touch point, assuming that its relative placement is consistent with that of its corresponding template point. Using two touch points, TOUCHTOKENTRACKER computes the relative change in orientation θ' since the token was first recognized.

All these data are made available to developers through three simple callbacks:

- tokenDown
- tokenMoved
- tokenUp

We also provide methods for giving more information about the geometry of the token and its state:

- getTouchPoints
- getNotchPoints
- getContourShape
- getTokenCenter
- getInitialOrientation
- getRelativeOrientation

5.5 Proof-of-concept Applications

This section presents three proof-of-concept applications that we developed using TOUCHTOKENBUILDER and TOUCHTOKENTRACKER.

All three applications feature tokens that have much more elaborate shapes than those tested in section 3.5. In each case, the tokens could be designed quickly, simply by loading SVG images fetched from the Web in TOUCHTOKENBUILDER, positioning the notches, exporting the tokens' vector description and feeding it to a laser cutter. As shown in Figure 5.7, some tokens were engraved, adding details such as, *e.g.*, the filament in the light bulb or the hole in the key, using an external vector drawing application.

The first proof-of-concept example is about controlling a virtual toy character using its tangible counterpart. The second example is a simplified house automation control system. Users can switch the light on/off, get information about energy consumption, turn on video-surveillance, play music, and lock the house. The last example is a bi-manual warship game. Users manipulate one of the ships with their dominant hand, and select a weapon with their other hand.

Figure 5.7: The three proof-of-concept token sets: toy characters (top), home automation (middle) and spaceship game (bottom).

5.6 Experiments

In this section, we report on experiments that we conducted to evaluate TOUCHTOKENTRACKER's accuracy for these tokens sets. Our proof-of-concept examples illustrate token sets for representative application areas. We use them as a mini-benchmark for validation purposes, showing that all token sets are indeed conflict-free in terms of recognition, and that TOUCHTOKENTRACKER can accurately identify the tokens' location and orientation.

To this end, we ran one experiment per token set {*Toys*, *Home*, *Space*}. Each experiment has two factors: TOKEN and ORIENTATION. *Toys* and *Home* each feature five tokens, while *Space* features four tokens. ORIENTATION can take five different values: $\{-\pi/3, -\pi/6, 0, +\pi/6, +\pi/3\}$. Values outside $\{-\pi/3, +\pi/3\}$ were not considered, as they would have been beyond the limits of users' range of motion.

Figure 5.8: A trial in our experiment: the participant has to dock the corresponding physical token inside the displayed silhouette.

At the start of each trial, a token's silhouette was displayed in the middle of the screen, with a specific orientation (Figure 5.8). Participants were asked to dock the corresponding physical token inside the silhouette and wait for the background to turn blue before lifting the token off the surface and proceed to the next trial. Participants had to hold the token still for 1 second. TOUCHTOKENTRACKER's algorithm would then run and log the ID of the recognized token, its estimated location and orientation.

Nine volunteers (one female), aged 23 to 33 year-old (average 26.5, median 26), participated in our study. Each of them performed the three experiments (one per token set) in a row. We counterbalanced token-set presentation order using a Latin-square, assigning three participants to each of the three orders. For each token set, participants ran 5 trials per TOKEN, testing the 5 ORIENTATION values. The experiment was approximately 10-minute long. It started with 5 practice trials (randomized TOKEN $\times$ ORIENTATION conditions), followed by 25 measure trials (20 for *Space*) presented in random order. The experiment ran on a tablet (Samsung SM-T810 Galaxy Tab S2) with a 237.3 $\times$ 169 mm display area and a resolution of 2048 $\times$ 1536 pixels. Participants were standing up, holding the tablet during the whole experiment.

5.7 Results

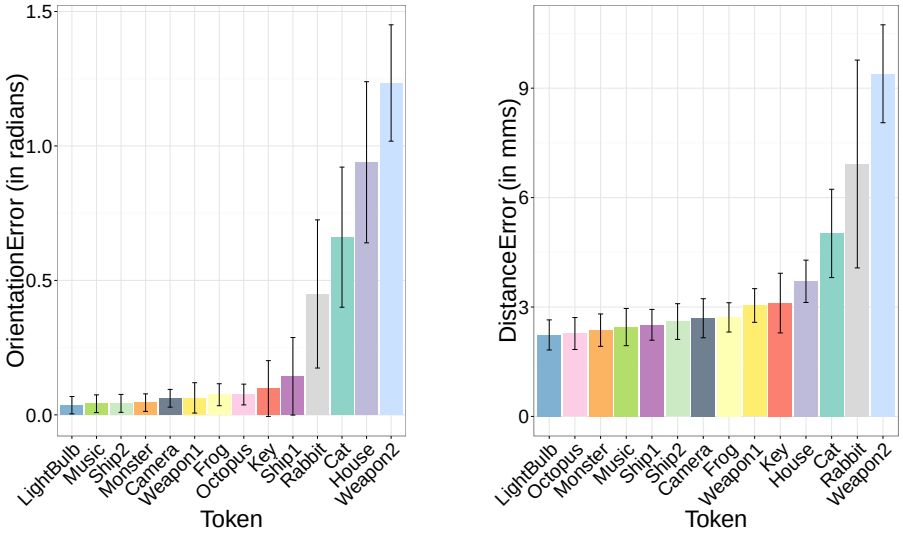


Figure 5.9: *OrientationError* (left) and *DistanceError* (right) per TOKEN condition. Error bars represent the 95% confidence interval.

We considered the following three measures to capture TOUCHTOKENTRACKER's accuracy;

1. *RecognitionError*: a binary measure whose value is 0 in case the token is accurately recognized and 1 otherwise.
2. *OrientationError*: a continuous measure of the absolute difference (in radians) between the silhouette's orientation and the physical token's orientation as estimated by TOUCHTOKENTRACKER.
3. *DistanceError*: a continuous measure of the distance (in millimeters) between the silhouette's center and the physical token's center, as estimated by TOUCHTOKENTRACKER.

We observed an overall recognition accuracy of 98%. The recognizer failed to identify the correct token in only 12 of the 630 trials: 6 times with the CAT, 3 times with the CAMERA, and once with the KEY, the RABBIT and WEAPON1. A Friedman rank sum test revealed that the difference between the different TOKEN conditions regarding recognition accuracy is actually significant ($\tilde{\chi}^2(13) = 38$, $p < 0.001$). We attribute this

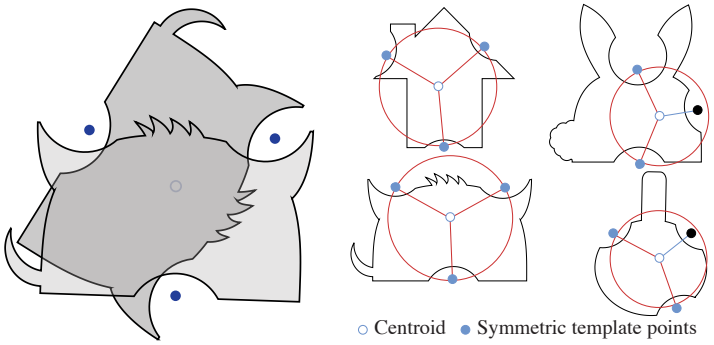


Figure 5.10: Token orientation: (left) an example of ambiguity; (right) error-prone tokens: HOUSE, RABBIT, CAT, WEAPON2.

difference to the fact that the CAT token requires users to spread the index and middle fingers a bit too much. Participants might have placed their index and middle fingers closer together so as to make their grasp more comfortable, thus not exactly coinciding with the notches.

Figure 5.9 illustrates that for 10 of the 14 tokens, *OrientationError* is less than 0.15 ($\frac{\pi}{20}$) and *DistanceError* is less than 3.1mm. However, TOUCHTOKENTRACKER's estimations are much less accurate for the other 4 tokens: HOUSE, CAT, RABBIT and WEAPON2. This result is not really surprising: these tokens feature at least two template points that are symmetric relative to the axis defined by the third point and the centroid, which implies that there is more than one solution for the recognizer's best-alignment algorithm.

Figure 5.10 illustrates how two orientations can match the same template points. It also shows that the four problematic tokens do feature at least two template points that can cause confusion. As the token's center location is derived from the token's orientation (Figure 5.6), it is not surprising that *DistanceError* is also larger for the four problematic tokens than for the other ten, whose orientation was properly estimated by TOUCHTOKENTRACKER. We computed a linear regression to predict *DistanceError* from *OrientationError*. We found a significant relation (F(1,616), $p < 0.001$) with r^2=0.53. We acknowledge this limitation of our approach, which is due to the fact that it relies on passive tokens and thus on what can be inferred from the three finger contact points only. However, this limitation can be alleviated by eliminating a range of unlikely orientations that fall out of users' range of motion, possibly warning users if the manipulated token still features an axis of symmetry.

5.8 Summary

Taken together, TOUCHTOKENBUILDER and TOUCHTOKENTRACKER enable designers to build low-cost tangible interfaces using TOUCHTOKENS while addressing several limitations of the original approach. Each of the two tools, however, still has its own limitations, several of which can be addressed in future work.

First, TOUCHTOKENBUILDER lets users freely position notches on the tokens and warns them about potential conflicts, providing support for a more exploratory design process. But TOUCHTOKENBUILDER does not make suggestions about notch positioning, leaving it to the designer to find positions that (1) remain comfortable to grasp, and (2) do not alter the token contours too much, so as to preserve enough semantics in their shape. Designers have to rely on their personal judgment, visually adjusting finger circles using direct manipulation, and putting their fingers on the screen to evaluate grasp comfort informally. We plan to investigate how TOUCHTOKENBUILDER could provide more support by, e.g., automatically identifying contour sections that would be good candidates for hosting notches. This implies defining heuristics that reflect how much a shape is altered (for instance, quantify the amount of detail removed if a portion is carved in), and to run empirical studies to identify the space of comfortable grasps. TOUCHTOKENBUILDER could also provide more support for FLEXIBLE TOUCHTOKENS by suggesting the lattice hinges pattern depending on the geometry and the material of the tokens.

Second, TOUCHTOKENTRACKER provides estimates of the token's location and orientation, but those can be wrong in some cases. As shown earlier, we can eliminate high-amplitude errors, but there will still remain some uncertainty. This latter limitation results from the trade-off between accuracy and ease-of-implementation in approaches such as ours: relying on fully passive tokens makes building tangible interfaces easy, but requires the system to infer a lot from very few input data, which are limited in our case to the fingers' contact points.

CHAPTER 6

CONCLUSIONS

6.1 Contributions

In this section, we restate the contributions presented in the introduction and summarize how each of these contributions was achieved.

Our primary contribution is TOUCHTOKENS, a novel interaction technique that combines tangible and gesture-based input using passive tokens on a regular multi-touch surface (Chapter 3). By offering some tangibility to multi-touch gestures, TOUCHTOKENS act as guides for learning and discovering specific touch patterns. They also provide a means to design low-cost tangible interfaces, allowing interface designers to move objects from the virtual world into the physical world. Besides, in comparison with other technologies for building tangible interfaces, TOUCHTOKENS are low-cost and very easy to build. TOUCHTOKENS are fully passive, and get recognized by an algorithm, which is able to identify at least six touch patterns with a high level of accuracy (~98%). This recognizer is fast, robust and does not require any kind of training or calibration.

In Chapter 4, we increased the expressive power of TOUCHTOKENS by making them flexible so that users can deform them. In particular, users can squeeze and bend them. To achieve this, we introduced laser-cut lattice hinges in the construction of the tokens. This new token design provides enough elasticity to make the tokens easy to deform without requiring too high force, while ensuring that they revert to their original

shape when users stop applying force to them. We also designed an algorithm that can recognize those specific manipulations (bend and squeeze) by analyzing users' finger traces. This approach that consists in analyzing users' finger micro-movement during token manipulations allows us to also discriminate whether users left a token on the surface or they took it off when they remove their fingers from the surface. These new interactions allow developers to implement more advanced features to their system like click actions, drag and drop operations, as well as the possibility of implementing solutions that rely on laying down several tokens on the surface.

Finally, in Chapter 5, we introduced two tools to facilitate the development of custom-made TOUCHTOKENS and increase the number and variety of TOUCHTOKENS that interface designers can consider in their applications. TOUCHTOKENBUILDER allows interface designers to create conflict-free sets of arbitrarily-shaped tokens using a simple direct-manipulation interface. It allows them to freely position notches on the tokens and warns them about potential conflicts, providing support for a more exploratory design process. TOUCHTOKENTRACKER augments the initial TOUCHTOKENS recognizer in order to also provide tokens' full geometry like location, orientation and shape during token manipulation. In addition, the library allows developers to keep tracking the tokens even when users lift a finger while manipulating tokens.

6.2 Limitations

We discuss below the limitations of the current implementation of TOUCHTOKENS that motivate future work.

One limitation could be the number of passive tokens. Our TOUCHTOKENS recognizer successfully discriminates a set of six passive tokens (Chapter 3) which is sufficient for a wide range of applications. However, further investigations are necessary to study larger sets of passive tokens. While the TOUCHTOKENS gestures involves three fingers in contact with the surface, it could be interesting to consider variants with four or five fingers to increase the gesture vocabulary.

TOUCHTOKENS have currently been tested for single-user applications and the recognizer may fail if two users place two tokens at the same time because it only analyzes the first three contact points without considering the distance and the location of the contact points. Further investigations should consider multi-user scenarios. It

would also be interesting to study whether we can identify users from their touch pattern or the dynamics of their micro-movements.

We also observed that our recognizer for flexible TOUCHTOKENS fails to detect state OFF right after leaving mode squeezed. It happens when users lift the token off while releasing the pressure applied on the token. Indeed, making tokens flexible provides opportunities for performing micro-movements in general, but it also has the side-effect of introducing some ambiguity in this particular case. Further investigations should consider this specific scenario.

The construction of flexible TOUCHTOKENS currently depends on the shape and the material of the tokens. For instance, our current lattice hinges design works well with wood tokens, but does not work with acrylic tokens. Moreover, some participants mentioned that $Triangle_5$ and $Rectangle_5$ were less conformable when performing squeeze gestures than the other tokens. Further investigations should provide recommendations for the construction of flexible TOUCHTOKENS depending on their geometry and material.

More generally, TOUCHTOKENBUILDER should provide more support for the design of TOUCHTOKENS. In addition to suggestions about the lattice hinges pattern for flexible TOUCHTOKENS, TOUCHTOKENBUILDER could provide suggestions regarding notch positioning. Further investigations should consider models to automatically identifying contour sections that would be good candidates for hosting notches. This implies identifying positions for the notches that remain comfortable to grasp and defining heuristics that reflect how much a shape is altered (for instance, quantify the amount of details removed if a portion is carved in).

6.3 Future Work

Our future research will start by addressing the limitations mentioned above. For instance, we plan to improve our recognizer to detect state *OFF* right after leaving mode squeezed. A possible solution that we want to explore is to adapt the values of the parameters involved in the criterion for discriminating *On* and *off* states (such as how fast the finger traces are moving away from the token center) depending on the current state of the token (*i.e.*, whether it is squeezed or not).

We also plan to generalize lattice hinges pattern to tokens with different shapes and materials (acrylic and 3D printed material) by conducting different empirical studies. We also plan to consider another approach. Currently, tokens were designed so that the forces were directed towards the center of the tokens. We will now also consider designs where forces are directed in different locations of the tokens. It might be especially interesting for tokens such as the crocodile token. When pressing the body of the crocodile, the applied forces could semi-open its mouth (similar to the mechanism involved in a pair of scissors). The semantic meaning of the token can thus be encoded in its deformation.

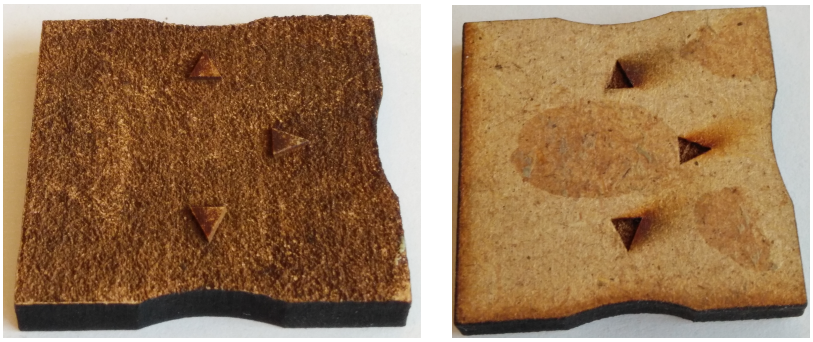


Figure 6.1: The two tokens that can be stacked. (Left) Below token. (Right) Above token.

We believe that we can push even further the number of different manipulations that we can perform with passive tokens. For example, we plan to investigate whether we can recognize *stacked* TOUCHTOKENS to increase the size of the gesture vocabulary. We hypothesize that we can design tokens that can be clamped together. The idea is that the radial distribution of the contact points around the token center (centroid) for a single token or a stacked token will be the same, but the distance from the touch points to the token center will increase (see Figure 6.2) because the slightly larger top token will constrain users' grasp.

To test this hypothesis, we recently built couples of tokens that can be stacked on top of each other. We used a laser cutter technique that creates three protuberant triangles (3mm depth) on the below token illustrated in Figure 6.1-left. This technique consists of performing multiple engraving passes on the same area to remove some material and thus reduce the token thickness in this area. For the token that is below, we used the same technique in order to remove all material but at the locations of the triangles (Figure 6.1-right) so that both tokens can be clamped together. Additional empirical studies are necessary to collect data and augment our recognizer.

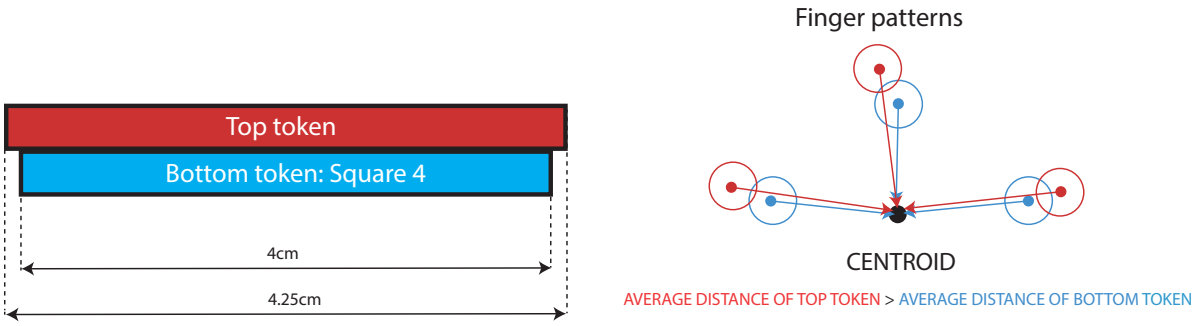


Figure 6.2: Approach for two stack tokens and the result finger patterns. The distance of the top token should be higher than the bottom token

Finally, we plan to investigate how passive tokens can play as haptic guides to help users to learn chording gestures. The idea is that users can implicit learn (complex) chording gestures by using TOUCHTOKENS. In the same spirit of marking menus [152], novice users could rely on passive tokens to execute specific touch patterns and progressively learn them. Once confident enough, they could decide to execute the right finger configurations without haptic guidance. We are currently designing and implementing a user study to test users' ability to learn touch patterns with TOUCHTOKENS.

BIBLIOGRAPHY

[1] C. Appert, "Touchtoken api, pdf description and code," (Date last accessed 19-June-2017). [Online]. Available: https://www.lri.fr/~appert/touchtokens/

[2] E. Pietriga, F. del Campo, A. Ibsen, R. Primet, C. Appert, O. Chapuis, M. Hempel, R. Muñoz, S. Eyheramendy, A. Jordan, and H. Dole, "Exploratory visualization of astronomical data on ultra-high-resolution wall displays," in *Proceedings of the Astronomical Telescopes and Instrumentation conference: Software and Cyberinfrastructure for Astronomy III, 9913*. SPIE, June 2016, pp. 0W:1–0W:15. [Online]. Available: http://dx.doi.org/10.1117/12.2231191

[3] U. Hinrichs, S. Carpendale, N. Valkanova, K. Kuikkaniemi, G. Jacucci, and A. V. Moere, "Interactive public displays," *IEEE Computer Graphics and Applications*, vol. 33, no. 2, pp. 25–27, 2013.

[4] M. Wu, C. Shen, K. Ryall, C. Forlines, and R. Balakrishnan, "Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces," in *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, ser. TABLETOP '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 185–192. [Online]. Available: http://dx.doi.org/10.1109/TABLETOP.2006.19

[5] Y. Li, "Gesture-based interaction: A new dimension for mobile user interfaces," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI '12. New York, NY, USA: ACM, 2012, pp. 6–6. [Online]. Available: http://doi.acm.org/10.1145/2254556.2254560

[6] I. Chatterjee, R. Xiao, and C. Harrison, "Gaze+gesture: Expressive, precise and targeted free-space interactions," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ser. ICMI '15.

New York, NY, USA: ACM, 2015, pp. 131–138. [Online]. Available: http://doi.acm.org/10.1145/2818346.2820752

[7] C. Appert and S. Zhai, "Using strokes as command shortcuts: Cognitive benefits and toolkit support," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 2289–2298. [Online]. Available: http://doi.acm.org/10.1145/1518701.1519052

[8] C. North, T. Dwyer, B. Lee, D. Fisher, P. Isenberg, G. Robertson, and K. Inkpen, "Understanding multi-touch manipulation for surface computing," in *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, ser. INTERACT '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 236–249. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03658-3_31

[9] W. S. Yu, H. van Duinen, and S. C. Gandevia, "Limits to the control of the human thumb and fingers in flexion and extension," *Journal of Neurophysiology*, vol. 103, no. 1, pp. 278–289, 2010. [Online]. Available: http://jn.physiology.org/content/103/1/278

[10] S. Li and C. T. Leonard, "The effect of enslaving on perception of finger forces," *Experimental Brain Research*, vol. 172, no. 3, pp. 301–309, 2006.

[11] M. S. Horn, R. J. Crouser, and M. U. Bers, "Tangible interaction and learning: the case for a hybrid approach," *Personal and Ubiquitous Computing*, vol. 16, no. 4, pp. 379–389, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00779-011-0404-2

[12] H. Ishii, "Tangible bits: Beyond pixels," in *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction*, ser. TEI '08. New York, NY, USA: ACM, 2008, pp. xv–xxv. [Online]. Available: http://doi.acm.org/10.1145/1347390.1347392

[13] Y. Jansen, P. Dragicevic, and J.-D. Fekete, "Tangible remote controllers for wall-size displays," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 2865–2874. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208691

[14] J. Patten and H. Ishii, "A comparison of spatial organization strategies in graphical and tangible user interfaces," in *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, ser. DARE '00. New York, NY, USA: ACM, 2000, pp. 41–50. [Online]. Available: http://doi.acm.org/10.1145/354666.354671

[15] S. Brave, H. Ishii, and A. Dahley, "Tangible interfaces for remote collaboration and communication," in *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '98. New York, NY, USA: ACM, 1998, pp. 169–178. [Online]. Available: http://doi.acm.org/10.1145/289444.289491

[16] S. C. and B. L., "Is gesture-based interaction a way to make interfaces more intuitive and accessible?" ser. HCI '09, Cambridge: British Computer Society, 2009.

[17] S. Zhai, P. O. Kristensson, C. Appert, T. H. Andersen, and X. Cao, "Foundational Issues in Touch-Screen Stroke Gesture Design - An Integrative Review," *Foundations and Trends in Human-Computer Interaction*, vol. 5, no. 2, pp. 97–205, Dec. 2012. [Online]. Available: https://hal.inria.fr/hal-00765046

[18] R. Aigner, D. Wigdor, H. Benko, M. Haller, D. Lindbauer, A. Ion, S. Zhao, and J. T. K. V. Koh, "Understanding mid-air hand gestures: A study of human preferences in usage of gesture types for hci," Tech. Rep., November 2012.

[19] I. E. Sutherland, "Sketchpad: A man-machine graphical communication system," in *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference*, ser. AFIPS '63 (Spring). New York, NY, USA: ACM, 1963, pp. 329–346. [Online]. Available: http://doi.acm.org/10.1145/1461551.1461591

[20] T. P. Moran, P. Chiu, and W. van Melle, "Pen-based interaction techniques for organizing material on an electronic whiteboard," in *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '97. New York, NY, USA: ACM, 1997, pp. 45–54. [Online]. Available: http://doi.acm.org/10.1145/263407.263508

[21] C. G. Wolf, J. R. Rhyne, and L. K. Briggs, "Communication and information retrieval with a pen-based meeting support tool," in *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, ser. CSCW '92. New York, NY, USA: ACM, 1992, pp. 322–329. [Online]. Available: http://doi.acm.org/10.1145/143457.143539

[22] A. Holzinger, M. Höller, M. Schedlbauer, and B. Urlesberger, "An investigation of finger versus stylus input in medical scenarios," in *Proceedings of the ITI 2008 30th International Conference on Information Technology Interfaces, Dubrovnik, Croatia, June 23-26, 2008*, 2008, pp. 433–438. [Online]. Available: http://dx.doi.org/10.1109/ITI.2008.4588449

[23] D. Vogel and P. Baudisch, "Shift: A technique for operating pen-based interfaces using touch," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 657–666. [Online]. Available: http://doi.acm.org/10.1145/1240624.1240727

[24] M. Annett, F. Anderson, W. F. Bischof, and A. Gupta, "The pen is mightier: Understanding stylus behaviour while inking on tablets," in *Proceedings of Graphics Interface 2014*, ser. GI '14. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2014, pp. 193–200. [Online]. Available: http://dl.acm.org/citation.cfm?id=2619648.2619680

[25] C. Forlines and R. Balakrishnan, "Evaluating tactile feedback and direct vs. indirect stylus input in pointing and crossing selection tasks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 1563–1572. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357299

[26] A. Cockburn, D. Ahlström, and C. Gutwin, "Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse," *Int. J. Hum.-Comput. Stud.*, vol. 70, no. 3, pp. 218–233, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.ijhcs.2011.11.002

[27] H. Tu, X. Ren, and S. Zhai, "A comparative evaluation of finger and pen stroke gestures," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 1287–1296. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208584

[28] C. Appert and S. Zhai, "Using Strokes as Command Shortcuts: Cognitive Benefits and Toolkit Support," in *International conference on Human factors in computing systems*, Boston, MA, United States, 2010, pp. 2289–2298. [Online]. Available: https://hal.inria.fr/inria-00538372

[29] A. C. Long, Jr., J. A. Landay, L. A. Rowe, and J. Michiels, "Visual similarity of pen gestures," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '00. New York, NY, USA: ACM, 2000, pp. 360–367. [Online]. Available: http://doi.acm.org/10.1145/332040.332458

[30] O. Bau and W. E. Mackay, "Octopocus: A dynamic guide for learning gesture-based command sets," in *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '08. New York, NY, USA: ACM, 2008, pp. 37–46. [Online]. Available: http://doi.acm.org/10.1145/1449715.1449724

[31] A. Wilson and S. Shafer, "Xwand: Ui for intelligent spaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '03. New York, NY, USA: ACM, 2003, pp. 545–552. [Online]. Available: http://doi.acm.org/10.1145/642611.642706

[32] X. Cao and R. Balakrishnan, "Evaluation of an on-line adaptive gesture interface with command prediction," in *Proceedings of Graphics Interface 2005*, ser. GI '05. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society, 2005, pp. 187–194. [Online]. Available: http://dl.acm.org/citation.cfm?id=1089508.1089540

[33] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '07. New York, NY, USA: ACM, 2007, pp. 159–168. [Online]. Available: http://doi.acm.org/10.1145/1294211.1294238

[34] S. Heo, J. Gu, and G. Lee, "Expanding touch input vocabulary by using consecutive distant taps," in *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 2597–2606. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557234

[35] C. Holz and P. Baudisch, "The generalized perceived input point model and how to double touch accuracy by extracting fingerprints," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 581–590. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753413

[36] R. L. Potter, L. J. Weldon, and B. Shneiderman, "Improving the accuracy of touch screens: An experimental evaluation of three strategies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '88. New York, NY, USA: ACM, 1988, pp. 27–32. [Online]. Available: http://doi.acm.org/10.1145/57167.57171

[37] A. Sears and B. Shneiderman, "High precision touchscreens: Design strategies and comparisons with a mouse," *Int. J. Man-Mach. Stud.*, vol. 34, no. 4, pp. 593–613, Apr. 1991. [Online]. Available: http://dx.doi.org/10.1016/0020-7373(91)90037-8

[38] D. Wigdor, H. Benko, J. Pella, J. Lombardo, and S. Williams, "Rock & rails: Extending multi-touch interactions with shape gestures to enable precise spatial manipulations," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1581–1590. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979173

[39] T. Moscovich, "Contact area interaction with sliding widgets," in *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '09. New York, NY, USA: ACM, 2009, pp. 13–22. [Online]. Available: http://doi.acm.org/10.1145/1622176.1622181

[40] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen, "Lucid touch: A see-through mobile device," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '07. New York, NY, USA: ACM, 2007, pp. 269–278. [Online]. Available: http://doi.acm.org/10.1145/1294211.1294259

[41] A. Olwal, S. Feiner, and S. Heyman, "Rubbing and tapping for precise and rapid selection on touch-screen displays," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 295–304. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357105

[42] A. Goguey, G. Casiez, D. Vogel, F. Chevalier, T. Pietrzak, and N. Roussel, "A three-step interaction pattern for improving discoverability in finger identification techniques," in *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology*,

ser. UIST'14 Adjunct. New York, NY, USA: ACM, 2014, pp. 33–34. [Online]. Available: http://doi.acm.org/10.1145/2658779.2659100

[43] C. Stewart, M. Rohs, S. Kratz, and G. Essl, "Characteristics of pressure-based input for mobile devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 801–810. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753444

[44] C. Harrison, J. Schwarz, and S. E. Hudson, "Tapsense: Enhancing finger interaction on touch surfaces," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 627–636. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047279

[45] A. Sugiura and Y. Koseki, "A user interface using fingerprint recognition: Holding commands and data objects on fingers," in *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '98. New York, NY, USA: ACM, 1998, pp. 71–79. [Online]. Available: http://doi.acm.org/10.1145/288392.288575

[46] C. Holz and P. Baudisch, "Fiberio: A touchscreen that senses fingerprints," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '13. New York, NY, USA: ACM, 2013, pp. 41–50. [Online]. Available: http://doi.acm.org/10.1145/2501988.2502021

[47] H. Benko, A. D. Wilson, and P. Baudisch, "Precise selection techniques for multi-touch screens," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '06. New York, NY, USA: ACM, 2006, pp. 1263–1272. [Online]. Available: http://doi.acm.org/10.1145/1124772.1124963

[48] S. Boring, D. Ledo, X. A. Chen, N. Marquardt, A. Tang, and S. Greenberg, "The fat thumb: Using the thumb's contact size for single-handed mobile interaction," in *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services Companion*, ser. MobileHCI '12. New York, NY, USA: ACM, 2012, pp. 207–208. [Online]. Available: http://doi.acm.org/10.1145/2371664.2371711

[49] S. Heo and G. Lee, "Force gestures: Augmenting touch screen gestures with normal and tangential forces," in *Proceedings of the 24th Annual*

ACM Symposium on User Interface Software and Technology, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 621–626. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047278

[50] H. Benko, T. S. Saponas, D. Morris, and D. Tan, "Enhancing input on and above the interactive surface with muscle sensing," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '09. New York, NY, USA: ACM, 2009, pp. 93–100. [Online]. Available: http://doi.acm.org/10.1145/1731903.1731924

[51] M. Moyle and A. Cockburn, "The design and evaluation of a flick gesture for 'back' and 'forward' in web browsers," in *Proceedings of the Fourth Australasian User Interface Conference on User Interfaces 2003 - Volume 18*, ser. AUIC '03. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2003, pp. 39–46. [Online]. Available: http://dl.acm.org/citation.cfm?id=820086.820097

[52] M. Jain and R. Balakrishnan, "User learning and performance with bezel menus," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 2221–2230. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208376

[53] Y. Li, "Gesture search: A tool for fast mobile data access," in *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '10. New York, NY, USA: ACM, 2010, pp. 87–96. [Online]. Available: http://doi.acm.org/10.1145/1866029.1866044

[54] A. Roudaut, E. Lecolinet, and Y. Guiard, "Microrolls: Expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 927–936. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518843

[55] D. Bonnet, C. Appert, and M. Beaudouin-Lafon, "Extending the vocabulary of touch events with thumbrock," in *Proceedings of Graphics Interface 2013*, ser. GI '13. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2013, pp. 221–228. [Online]. Available: http://dl.acm.org/citation.cfm?id=2532129.2532166

[56] N. Matsushita and J. Rekimoto, "Holowall: Designing a finger, hand, body, and object sensitive wall," in *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '97. New York, NY, USA: ACM, 1997, pp. 209–210. [Online]. Available: http://doi.acm.org/10.1145/263407.263549

[57] G. Bailly, E. Lecolinet, and Y. Guiard, "Finger-count & radial-stroke shortcuts: 2 techniques for augmenting linear menus on multi-touch surfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 591–594. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753414

[58] D. Freeman and R. Balakrishnan, "Tangible actions," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '11. New York, NY, USA: ACM, 2011, pp. 87–96. [Online]. Available: http://doi.acm.org/10.1145/2076354.2076373

[59] D. Schmidt, M. K. Chong, and H. Gellersen, "Idlenses: Dynamic personal areas on shared surfaces," in *ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '10. New York, NY, USA: ACM, 2010, pp. 131–134. [Online]. Available: http://doi.acm.org/10.1145/1936652.1936678

[60] ——, "Handsdown: Hand-contour-based user identification for interactive surfaces," in *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, ser. NordiCHI '10. New York, NY, USA: ACM, 2010, pp. 432–441. [Online]. Available: http://doi.acm.org/10.1145/1868914.1868964

[61] X. Cao, A. D. Wilson, R. Balakrishnan, K. Hinckley, and S. E. Hudson, "Shapetouch: Leveraging contact shape on interactive surfaces," in *2008 3rd IEEE International Workshop on Horizontal Interactive Human Computer Systems*, Oct 2008, pp. 129–136.

[62] C. T. Dang, M. Straub, and E. André, "Hand distinction for multi-touch tabletop interaction," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '09. New York, NY, USA: ACM, 2009, pp. 101–108. [Online]. Available: http://doi.acm.org/10.1145/1731903.1731925

[63] M. Khalilbeigi, J. Steimle, J. Riemann, N. Dezfuli, M. Mühlhäuser, and J. D. Hollan, "Objectop: occlusion awareness of physical objects on interactive tabletops," in

Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces. ACM, 2013, pp. 255–264.

[64] P. Dietz and D. Leigh, "Diamondtouch: A multi-user touch technology," in *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '01. New York, NY, USA: ACM, 2001, pp. 219–226. [Online]. Available: http://doi.acm.org/10.1145/502348.502389

[65] J. Rekimoto, "Smartskin: An infrastructure for freehand manipulation on interactive surfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '02. New York, NY, USA: ACM, 2002, pp. 113–120. [Online]. Available: http://doi.acm.org/10.1145/503376.503397

[66] G. J. Lepinski, T. Grossman, and G. Fitzmaurice, "The design and evaluation of multitouch marking menus," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2233–2242. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753663

[67] E. Ghomi, S. Huot, O. Bau, M. Beaudouin-Lafon, and W. E. Mackay, "Arpège: Learning multitouch chord gestures vocabularies," in *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '13. New York, NY, USA: ACM, 2013, pp. 209–218. [Online]. Available: http://doi.acm.org/10.1145/2512349.2512795

[68] J. Wagner, E. Lecolinet, and T. Selker, "Multi-finger chords for hand-held tablets: Recognizable and memorable," in *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 2883–2892. [Online]. Available: http://doi.acm.org/10.1145/2556288.2556958

[69] C. Harrison, R. Xiao, J. Schwarz, and S. E. Hudson, "Touchtools: Leveraging familiarity and skill with physical tools to augment touch interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 2913–2916. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557012

[70] Y. Luo and D. Vogel, "Pin-and-cross: A unimanual multitouch technique combining static touches with crossing selection," in *Proceedings of the 28th Annual*

ACM Symposium on User Interface Software & Technology, ser. UIST '15. New York, NY, USA: ACM, 2015, pp. 323–332. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807444

[71] T. Moscovich and J. F. Hughes, "Indirect mappings of multi-touch input using one and two hands," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '08. New York, NY, USA: ACM, 2008, pp. 1275–1284. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357254

[72] T. Yoshikawa, B. Shizuki, and J. Tanaka, "Handywidgets: Local widgets pulled-out from hands," in *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '12. New York, NY, USA: ACM, 2012, pp. 197–200. [Online]. Available: http://doi.acm.org/10.1145/2396636.2396667

[73] X. Bi, C. Chelba, T. Ouyang, K. Partridge, and S. Zhai, "Bimanual gesture keyboard," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '12. New York, NY, USA: ACM, 2012, pp. 137–146. [Online]. Available: http://doi.acm.org/10.1145/2380116.2380136

[74] C. Foucault, M. Micaux, D. Bonnet, and M. Beaudouin-Lafon, "SPad: A Bimanual Interaction Technique for Productivity Applications on Multi-touch Tablets," in *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. Toronto, Canada: ACM, Apr. 2014, pp. 1879–1884. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00997856

[75] M. Negulescu, J. Ruiz, and E. Lank, "Zoompointing revisited: Supporting mixed-resolution gesturing on interactive surfaces," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '11. New York, NY, USA: ACM, 2011, pp. 150–153. [Online]. Available: http://doi.acm.org/10.1145/2076354.2076382

[76] D. P. Käser, M. Agrawala, and M. Pauly, "Fingerglass: Efficient multiscale interaction on multitouch screens," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1601–1610. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979175

[77] S. Butscher, K. Hornbæk, and H. Reiterer, "Spacefold and physiclenses: Simultaneous multifocus navigation on touch surfaces," in *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, ser. AVI '14. New York, NY, USA: ACM, 2014, pp. 209–216. [Online]. Available: http://doi.acm.org/10.1145/2598153.2598177

[78] J. Wagner, S. Huot, and W. Mackay, "BiTouch and BiPad: Designing Bimanual Interaction for Hand-held Tablets," in *CHI'12 - 30th International Conference on Human Factors in Computing Systems - 2012*, ACM SIGCHI. Austin, United States: ACM Press, May 2012. [Online]. Available: https://hal.inria.fr/hal-00663972

[79] B. Ullmer and H. Ishii, "Emerging frameworks for tangible user interfaces," *IBM Syst. J.*, vol. 39, no. 3-4, pp. 915–931, Jul. 2000. [Online]. Available: http://dx.doi.org/10.1147/sj.393.0915

[80] L. Angelini, D. Lalanne, E. van den Hoven, A. Mazalek, O. Abou Khaled, and E. Mugellini, "Tangible meets gestural: Comparing and blending post-wimp interaction paradigms," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '15. New York, NY, USA: ACM, 2015, pp. 473–476. [Online]. Available: http://doi.acm.org/10.1145/2677199.2683583

[81] G. W. Fitzmaurice, H. Ishii, and W. A. S. Buxton, "Bricks: Laying the foundations for graspable user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '95. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 442–449. [Online]. Available: http://dx.doi.org/10.1145/223904.223964

[82] O. Zuckerman, T. Grotzer, and K. Leahy, "Flow blocks as a conceptual bridge between understanding the structure and behavior of a complex causal system," in *Proceedings of the 7th International Conference on Learning Sciences*, ser. ICLS '06. International Society of the Learning Sciences, 2006, pp. 880–886. [Online]. Available: http://dl.acm.org/citation.cfm?id=1150034.1150162

[83] O. Zuckerman, S. Arida, and M. Resnick, "Extending tangible interfaces for education: Digital montessori-inspired manipulatives," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI

'05. New York, NY, USA: ACM, 2005, pp. 859–868. [Online]. Available: http://doi.acm.org/10.1145/1054972.1055093

[84] P. Wellner, "Interacting with paper on the digitaldesk," *Commun. ACM*, vol. 36, no. 7, pp. 87–96, Jul. 1993. [Online]. Available: http://doi.acm.org/10.1145/159544.159630

[85] J. Underkoffler and H. Ishii, "Urp: A luminous-tangible workbench for urban planning and design," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '99. New York, NY, USA: ACM, 1999, pp. 386–393. [Online]. Available: http://doi.acm.org/10.1145/302979.303114

[86] J. Patten, H. Ishii, J. Hines, and G. Pangaro, "Sensetable: A wireless object tracking platform for tangible user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '01. New York, NY, USA: ACM, 2001, pp. 253–260. [Online]. Available: http://doi.acm.org/10.1145/365024.365112

[87] M. Sugimoto, K. Hosoi, and H. Hashizume, "Caretta: A system for supporting face-to-face collaboration by integrating personal and shared spaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '04. New York, NY, USA: ACM, 2004, pp. 41–48. [Online]. Available: http://doi.acm.org/10.1145/985692.985698

[88] B. Ullmer and H. Ishii, "The metadesk: Models and prototypes for tangible user interfaces," in *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '97. New York, NY, USA: ACM, 1997, pp. 223–232. [Online]. Available: http://doi.acm.org/10.1145/263407.263551

[89] B. Ullmer, H. Ishii, and R. J. K. Jacob, "Token+constraint systems for tangible interaction with digital information," *ACM Trans. Comput.-Hum. Interact.*, vol. 12, no. 1, pp. 81–118, Mar. 2005. [Online]. Available: http://doi.acm.org/10.1145/1057237.1057242

[90] B. Ullmer, H. Ishii, and D. Glas, "mediablocks: Physical containers, transports, and controls for online media," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH

'98. New York, NY, USA: ACM, 1998, pp. 379–386. [Online]. Available: http://doi.acm.org/10.1145/280814.280940

[91] S. Klum, P. Isenberg, R. Langner, J.-D. Fekete, and R. Dachselt, "Stackables: Combining tangibles for faceted browsing," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI '12. New York, NY, USA: ACM, 2012, pp. 241–248. [Online]. Available: http://doi.acm.org/10.1145/2254556.2254600

[92] D. Merrill, E. Sun, and J. Kalanithi, "Sifteo cubes," in *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '12. New York, NY, USA: ACM, 2012, pp. 1015–1018. [Online]. Available: http://doi.acm.org/10.1145/2212776.2212374

[93] M. Le Goc, P. Dragicevic, S. Huron, J. Boy, and J.-D. Fekete, "SmartTokens: Embedding Motion and Grip Sensing in Small Tangible Objects," in *User Interface Software and Technology Symposium*. Charlotte, NC, United States: ACM, Nov. 2015. [Online]. Available: https://hal.inria.fr/hal-01211270

[94] E. S. Martinussen and T. Arnall, "Designing with rfid," in *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, ser. TEI '09. New York, NY, USA: ACM, 2009, pp. 343–350. [Online]. Available: http://doi.acm.org/10.1145/1517664.1517734

[95] S. Kubicki, S. Lepreux, and C. Kolski, "Rfid-driven situation awareness on tangisense, a table interacting with tangible objects," *Personal and Ubiquitous Computing*, vol. 16, no. 8, pp. 1079–1094, 2012. [Online]. Available: http://dx.doi.org/10.1007/s00779-011-0442-9

[96] A. N. Antle, A. F. Wise, and K. Nielsen, "Towards utopia: Designing tangibles for learning," in *Proceedings of the 10th International Conference on Interaction Design and Children*, ser. IDC '11. New York, NY, USA: ACM, 2011, pp. 11–20. [Online]. Available: http://doi.acm.org/10.1145/1999030.1999032

[97] M. Pyykkönen, J. Riekki, M. Jurmu, and I. Sanchéz Milara, "Activity pad: Teaching tool combining tangible interaction and affordance of paper," in *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '13. New York, NY, USA: ACM, 2013, pp. 135–144. [Online]. Available: http://doi.acm.org/10.1145/2512349.2512810

[98] A. Bianchi and I. Oakley, "Designing tangible magnetic appcessories," in *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, ser. TEI '13. New York, NY, USA: ACM, 2013, pp. 255–258. [Online]. Available: http://doi.acm.org/10.1145/2460625.2460667

[99] S. Hwang, M. Ahn, and K.-y. Wohn, "Maggetz: Customizable passive tangible controllers on and around conventional mobile devices," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '13. New York, NY, USA: ACM, 2013, pp. 411–416. [Online]. Available: http://doi.acm.org/10.1145/2501988.2501991

[100] S. Hwang, A. Bianchi, M. Ahn, and K. Wohn, "Magpen: Magnetically driven pen interactions on and around conventional smartphones," in *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, ser. MobileHCI '13. New York, NY, USA: ACM, 2013, pp. 412–415. [Online]. Available: http://doi.acm.org/10.1145/2493190.2493194

[101] J. Leitner and M. Haller, "Geckos: Combining magnets and pressure images to enable new tangible-object design and interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 2985–2994. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979385

[102] R.-H. Liang, K.-Y. Cheng, L. Chan, C.-X. Peng, M. Y. Chen, R.-H. Liang, D.-N. Yang, and B.-Y. Chen, "Gaussbits: Magnetic tangible bits for portable and occlusion-free near-surface interactions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. New York, NY, USA: ACM, 2013, pp. 1391–1400. [Online]. Available: http://doi.acm.org/10.1145/2470654.2466185

[103] R.-H. Liang, H.-C. Kuo, L. Chan, D.-N. Yang, and B.-Y. Chen, "Gaussstones: Shielded magnetic tangibles for multi-token interactions on portable displays," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '14. New York, NY, USA: ACM, 2014, pp. 365–372. [Online]. Available: http://doi.acm.org/10.1145/2642918.2647384

[104] D. Vogel and G. Casiez, "Conté: Multimodal input inspired by an artist's crayon," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software*

and Technology, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 357–366. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047242

[105] B. Hartmann, M. R. Morris, H. Benko, and A. D. Wilson, "Augmenting interactive tables with mice & keyboards," in *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '09. New York, NY, USA: ACM, 2009, pp. 149–152. [Online]. Available: http://doi.acm.org/10.1145/1622176.1622204

[106] M. Weiss, J. Wagner, Y. Jansen, R. Jennings, R. Khoshabeh, J. D. Hollan, and J. Borchers, "Slap widgets: Bridging the gap between virtual and physical controls on tabletops," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 481–490. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518779

[107] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner, "The reactable: Exploring the synergy between live music performance and tabletop tangible interfaces," in *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*, ser. TEI '07. New York, NY, USA: ACM, 2007, pp. 139–146. [Online]. Available: http://doi.acm.org/10.1145/1226969.1226998

[108] P. Baudisch, T. Becker, and F. Rudeck, "Lumino: Tangible blocks for tabletop computers based on glass fiber bundles," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 1165–1174. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753500

[109] T. Bartindale and C. Harrison, "Stacks on the surface: Resolving physical order using fiducial markers with structured transparency," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '09. New York, NY, USA: ACM, 2009, pp. 57–60. [Online]. Available: http://doi.acm.org/10.1145/1731903.1731916

[110] C. Williams, X. D. Yang, G. Partridge, J. Millar-Usiskin, A. Major, and P. Irani, "Tzee: Exploiting the lighting properties of multi-touch tabletops for tangible 3d interactions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 1363–1372. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979143

[111] D. Avrahami, J. O. Wobbrock, and S. Izadi, "Portico: Tangible interaction on and around a tablet," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 347–356. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047241

[112] S. Kratz, T. Westermann, M. Rohs, and G. Essl, "Capwidgets: Tangible widgets versus multi-touch controls on mobile devices," in *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '11. New York, NY, USA: ACM, 2011, pp. 1351–1356. [Online]. Available: http://doi.acm.org/10.1145/1979742.1979773

[113] N.-H. Yu, S.-S. Tsai, I.-C. Hsiao, D.-J. Tsai, M.-H. Lee, M. Y. Chen, and Y.-P. Hung, "Clip-on gadgets: Expanding multi-touch interaction area with unpowered tactile controls," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 367–372. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047243

[114] L. Chan, S. Müller, A. Roudaut, and P. Baudisch, "Capstones and zebrawidgets: Sensing stacks of building blocks, dials and sliders on capacitive touch screens," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 2189–2192. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208371

[115] N.-H. Yu, S.-S. Tsai, I.-C. Hsiao, D.-J. Tsai, M.-H. Lee, M. Y. Chen, and Y.-P. Hung, "Clip-on gadgets: Expanding multi-touch interaction area with unpowered tactile controls," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 367–372. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047243

[116] A. Wiethoff, H. Schneider, M. Rohs, A. Butz, and S. Greenberg, "Sketch-a-tui: Low cost prototyping of tangible interactions using cardboard and conductive ink," in *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, ser. TEI '12. New York, NY, USA: ACM, 2012, pp. 309–312. [Online]. Available: http://doi.acm.org/10.1145/2148131.2148196

[117] R. Blagojevic and B. Plimmer, *CapTUI: Geometric Drawing with Tangibles on a Capacitive Multi-touch Display*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 511–528. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40483-2_37

[118] S. Voelker, K. Nakajima, C. Thoresen, Y. Itoh, K. I. Overgård, and J. Borchers, "Pucs: Detecting transparent, passive untouched capacitive widgets on unmodified multi-touch displays," in *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '13. New York, NY, USA: ACM, 2013, pp. 101–104. [Online]. Available: http://doi.acm.org/10.1145/2512349.2512791

[119] D. Bouchard and S. Daniels, "Tiles that talk: Tangible templates for networked objects," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '15. New York, NY, USA: ACM, 2015, pp. 197–200. [Online]. Available: http://doi.acm.org/10.1145/2677199.2680607

[120] H.-C. Jetter, J. Gerken, M. Zöllner, H. Reiterer, and N. Milic-Frayling, "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*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 3013–3022. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979390

[121] C. Valdes, D. Eastman, C. Grote, S. Thatte, O. Shaer, A. Mazalek, B. Ullmer, and M. K. Konkel, "Exploring the design space of gestural interaction with active tokens through user-defined gestures," in *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 4107–4116. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557373

[122] J. Patten, B. Recht, and H. Ishii, "Audiopad: A tag-based interface for musical performance," in *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*, ser. NIME '02. Singapore, Singapore: National University of Singapore, 2002, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?id=1085171.1085175

[123] G. Levin, "The table is the score: An augmented-reality interface for real-time, tangible, spectrographic performance," in *In Proceedings of the International Computer Music Conference*, 2006.

[124] S. Jordà, "The reactable: Tangible and tabletop music performance," in *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '10. New York, NY, USA: ACM, 2010, pp. 2989–2994. [Online]. Available: http://doi.acm.org/10.1145/1753846.1753903

[125] B. Schiettecatte and J. Vanderdonckt, "Audiocubes: A distributed cube tangible interface based on interaction range for sound design," in *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction*, ser. TEI '08. New York, NY, USA: ACM, 2008, pp. 3–10. [Online]. Available: http://doi.acm.org/10.1145/1347390.1347394

[126] K. Kobayashi, M. Hirano, A. Narita, and H. Ishii, "A tangible interface for ip network simulation," in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '03. New York, NY, USA: ACM, 2003, pp. 800–801. [Online]. Available: http://doi.acm.org/10.1145/765891.766000

[127] S. Voelker, C. Cherek, J. Thar, T. Karrer, C. Thoresen, K. I. Overgård, and J. Borchers, "Percs: Persistently trackable tangibles on capacitive multi-touch displays," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ser. UIST '15. New York, NY, USA: ACM, 2015, pp. 351–356. [Online]. Available: http://doi.acm.org/10.1145/2807442.2807466

[128] U. von Zadow, D. Bösel, D. D. Dam, A. Lehmann, P. Reipschläger, and R. Dachselt, "Miners: Communication and awareness in collaborative gaming at an interactive display wall," in *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*. New York, NY, USA: ACM, 11 2016, pp. 235–240. [Online]. Available: http://dx.doi.org/10.1145/2992154.2992174

[129] J. Arita, J. H. Seo, and S. Aldriedge, "Soft tangible interaction design with tablets for young children," in *ACM SIGGRAPH 2014 Posters*, ser. SIGGRAPH '14. New York, NY, USA: ACM, 2014, pp. 54:1–54:1. [Online]. Available: http://doi.acm.org/10.1145/2614217.2614267

[130] C. Magerkurth, M. Memisoglu, T. Engelke, and N. Streitz, "Towards the next generation of tabletop gaming experiences," in *Proceedings of Graphics Interface 2004*, ser. GI '04. School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society, 2004, pp. 73–80. [Online]. Available: http://dl.acm.org/citation.cfm?id=1006058.1006068

[131] E. Brule, G. Bailly, A. Brock, F. Valentin, G. Denis, and C. Jouffrais, "Mapsense: Multi-sensory interactive maps for children living with visual impairments," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: ACM, 2016, pp. 445–457. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858375

[132] L. Xie, A. N. Antle, and N. Motamedi, "Are tangibles more fun?: Comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces," in *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction*, ser. TEI '08. New York, NY, USA: ACM, 2008, pp. 191–198. [Online]. Available: http://doi.acm.org/10.1145/1347390.1347433

[133] R. J. W. Sluis, I. Weevers, C. H. G. J. van Schijndel, L. Kolos-Mazuryk, S. Fitrianie, and J. B. O. S. Martens, "Read-it: Five-to-seven-year-old children learn to read in a tabletop environment," in *Proceedings of the 2004 Conference on Interaction Design and Children: Building a Community*, ser. IDC '04. New York, NY, USA: ACM, 2004, pp. 73–80. [Online]. Available: http://doi.acm.org/10.1145/1017833.1017843

[134] J. Marco, E. Cerezo, and S. Baldassarri, "Playing with toys on a tabletop active surface," in *IDC*, 2010.

[135] S. Do-Lenh, F. Kaplan, and P. Dillenbourg, "Paper-based concept map: The effects of tabletop on an expressive collaborative learning task," in *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, ser. BCS-HCI '09. Swinton, UK, UK: British Computer Society, 2009, pp. 149–158. [Online]. Available: http://dl.acm.org/citation.cfm?id=1671011.1671028

[136] B. Schneider, P. Blikstein, and W. Mackay, "Combinatorix: A tangible user interface that supports collaborative learning of probabilities," in *Proceedings*

of the 2012 ACM International Conference on Interactive Tabletops and Surfaces, ser. ITS '12. New York, NY, USA: ACM, 2012, pp. 129–132. [Online]. Available: http://doi.acm.org/10.1145/2396636.2396656

[137] A. Loparev, L. Westendorf, M. Flemings, J. Cho, R. Littrell, A. Scholze, and O. Shaer, "Bacpack: Exploring the role of tangibles in a museum exhibit for bio-design," in *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '17. New York, NY, USA: ACM, 2017, pp. 111–120. [Online]. Available: http://doi.acm.org/10.1145/3024969.3025000

[138] D. Freeman, H. Benko, M. R. Morris, and D. Wigdor, "Shadowguides: Visualizations for in-situ learning of multi-touch and whole-hand gestures," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '09. New York, NY, USA: ACM, 2009, pp. 165–172. [Online]. Available: http://doi.acm.org/10.1145/1731903.1731935

[139] H. Olafsdottir, T. Tsandilas, and C. Appert, "Prospective motor control on tabletops: Planning grasp for multitouch interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. ACM, 2014, pp. 2893–2902. [Online]. Available: http://doi.acm.org/10.1145/2556288.2557029

[140] H. Olafsdottir and C. Appert, "Multi-touch gestures for discrete and continuous control," in *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, ser. AVI '14. New York, NY, USA: ACM, 2014, pp. 177–184. [Online]. Available: http://doi.acm.org/10.1145/2598153.2598169

[141] J. M. Loomis and S. J. Lederman, "Tactual perception," in *Handbook of Perception and Human Performance*, K. R. Boff, L. Kaufman, and J. P. Thomas, Eds. John Wiley & Sons, 1986.

[142] H. Koike, W. Nishikawa, and K. Fukuchi, "Transparent 2-d markers on an lcd tabletop system," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. ACM, 2009, pp. 163–172. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518728

[143] B. Ullmer, H. Ishii, and R. J. K. Jacob, "Tangible query interfaces: Physically constrained tokens for manipulating database queries," in *Proceedings of INTERACT âĂŹ03*, 2003, pp. 279–286.

[144] J. Rekimoto, B. Ullmer, and H. Oba, "Datatiles: A modular platform for mixed physical and graphical interactions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '01. ACM, 2001, pp. 269–276. [Online]. Available: http://doi.acm.org/10.1145/365024.365115

[145] M. Bostock, V. Ogievetsky, and J. Heer, "D3 data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.185

[146] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose, "Toolglass and magic lenses: The see-through interface," in *Proceedings of the 20th Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '93. ACM, 1993, pp. 73–80. [Online]. Available: http://doi.acm.org/10.1145/166117.166126

[147] W. Büschel, U. Kister, M. Frisch, and R. Dachselt, "T4 - transparent and translucent tangibles on tabletops," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, ser. AVI '14. ACM, 2014, pp. 81–88. [Online]. Available: http://doi.acm.org/10.1145/2598153.2598179

[148] L. E. Holmquist, J. Redström, and P. Ljungstrand, "Token-based access to digital information," in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, ser. HUC '99. Springer-Verlag, 1999, pp. 234–245. [Online]. Available: http://dl.acm.org/citation.cfm?id=647985.743869

[149] P. K. Sen, "Estimates of the regression coefficient based on kendall's tau," *Journal of the American Statistical Association*, vol. 63, no. 324, pp. 1379–1389, 1968.

[150] T. R. G. Green and M. Petre, "Usability analysis of visual programming environments: a "cognitive dimensions" framework," *JVLC*, vol. 7, no. 2, pp. 131–174, 1996.

[151] C. Appert, O. Chapuis, E. Pietriga, and M.-J. Lobo, "Reciprocal drag-and-drop," *ACM Trans. Comput.-Hum. Interact.*, vol. 22, no. 6, pp. 29:1–29:36, Sep. 2015. [Online]. Available: http://doi.acm.org/10.1145/2785670

[152] G. Kurtenbach and W. Buxton, "The limits of expert performance using hierarchic marking menus," in *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, ser. CHI '93. New York, NY, USA: ACM, 1993, pp. 482–487. [Online]. Available: http://doi.acm.org/10.1145/169059.169426

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Objets Passifs pour une Entrée Multi-points Riche

Mots clés: Interaction Homme-Machine, Interaction gestuelle, Interaction tangible

Résumé: L'entrée multi-points offre un canal d'interaction très expressif pour les dispositifs équipés d'une technologie tactile multi-points. Cependant, alors que la taille du canal de communication est, en théorie, trés grande, la plupart des systèmes n'en font, en pratique, qu'un usage très limité. Cet état de fait est probablement dû à la difficulté de gérer un grand nombre de gestes multi-points pour deux raisons principales: (1) les limites cognitives et motrices des humains et (2) les difficultés techniques pour l'élaboration de systèmes de reconnaissance robustes. Cette thèse étudie une nouvelle technique d'entrée, TouchTokens, pour enrichir le vocabulaire de gestes multi-points, en se basant sur la position relative des points de contact et des objets (tokens) passifs. Un TouchToken est un "token" passif avec des encoches qui indiquent à l'utilisateur comment l'attraper, et qui est donc associé à une configuration de doigts qui lui est propre.

Nous commençons par présenter le principe avec des tokens rigides de forme basique. L'algorithme de reconnaissance et la conception des tokens sont issus des conclusions d'une étude formative dans laquelle nous avons collecté et analysé des schémas de points de contact lorsque les utilisateurs tiennent des tokens de taille et de forme variable. Cette première étude montre que les utilisateurs ont des stratégies individuelles cohérentes, mais que ces stratégies dépendent de l'utilisateur. Ces conclusions nous ont mené à l'élaboration de tokens avec des encoches afin que les utilisateurs attrapent un même token toujours de la même façon. L'expérience que nous avons menée sur ce nouvel ensemble de tokens démontre que nous pouvons les reconnaître avec un niveau de robustesse supérieur à 95%.

La conception initiale des TouchTokens ne supporte qu'un ensemble d'interactions se limitant au modèle à deux états de l'interaction directe. Dans un second projet, nous décrivons une technique de fabrication avec une découpeuse laser qui permet de faire des tokens flexibles que les utilisateurs peuvent, par exemple, courber ou compresser en plus de les faire glisser sur la surface. Nous augmentons notre reconnaisseur pour analyser les micro-mouvements des doigts pendant la manipulation du token afin de reconnaître ces manipulations. Cette approche basée sur l'analyse des micro-mouvements des doigts nous permet également de discriminer, lorsque l'utilisateur enlève ses doigts de la surface, le cas où il enlève le token de la surface, du cas où le token est resté sur la surface. Nous rapportons sur les expériences que nous avons menées pour déterminer la valeur des paramètres de nos différents reconnaisseurs, et tester leur robustesse. Nous obtenons des taux de reconnaissance supérieurs à 90% sur les données collectées.

Nous finissons cette thèse par la présentation de deux outils qui permettent de construire et reconnaître des tokens de forme arbitraire, TouchTokenBuilder and TouchTokenTracker. TouchTokenBuilder est une application logicielle qui permet de placer des encoches sur des contours vectoriels de forme arbitraire, et qui alerte en cas de conflit de reconnaissance entre tokens. TouchTokenTracker est une librairie logicielle qui prend cette description numérique en entrée, et qui permet aux développeurs de traquer la géométrie.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

université PARIS-SACLAY | ÉCOLE DOCTORALE Sciences et technologies de l'information et de la communication (STIC)

Title: Rich Multi-touch Input with Passive Tokens

Keywords: Human-Computer Interaction, Gesture-based interaction, Tangible interaction

Abstract: This thesis investigates a novel input technique for enriching the gesture vocabulary on a multi-touch surface based on fingers' relative location and passive tokens.

The first project, TouchTokens, presents a novel technique for interacting with multi- touch surfaces and tokens. The originality is that these tokens are totally passive (no need for any additional electronic components) and their design features notches that guide users' grasp. The purpose of the notches is to indicate a finger spatial configuration (touch pattern) that is specific to the token. When users hold a token and place it on the surface, touching them simultaneously, the system can recognize the resulting touch patterns with a very high level of accuracy (>95%). This approach works on any touch- sensitive surface and makes it possible to easily build low-cost interfaces that combine no-conductive tangibles and gestural input.

This technique supports a new multi-touch input that the system can recognize. However, the interaction is limited to the two-state model of touch interaction as the system only knows the tokens' position and cannot detect tokens that are not touched. In the second project of the thesis, we introduce a laser-cut lattice hinge technique for making the tokens flexible. We then develop a new recognizer that analyzes the micro-movements of the fingers while user are holding and deforming those tokens on the surface. We run three experiments to design and calibrate algorithms for discriminating the three following types of manipulations: (1) when a token is left on the surface rather than taken off it (On/Off); (2) when a token has been bent, and (3) when it is squeezed. Our results show that our algorithms can recognize these three manipulations with an accuracy of: On/Off 90.1%, Bent 91.1% and Squeezed 96,9%.

The thesis concludes with the presentation of two tools, TouchTokenBuilder and TouchTokenTracker, for facilitating the development of tailor-made tangibles using a simple direct-manipulation interface. TouchTokenBuilder is a software application that assists interface designers in placing notches on arbitrarily-shaped vector contours for creating conflict-free token sets and warning them about potential conflicts. It outputs two files: a vector-graphics description of all tokens in the set and a numerical description of the geometry of each token. TouchTokenTracker is a software library that takes as input the numerical description produced by TouchTokenBuilder, and enables developers to track the tokens' full geometry (location, orientation and shape) throughout their manipulation on the multi-touch surface.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France