Passive yet Expressive TouchTokens

Rafael Morales González[1] **Caroline Appert**[1] **Gilles Bailly**[2,3] **Emmanuel Pietriga**[1]
[1]LRI, Univ. Paris-Sud, CNRS, INRIA, Université Paris-Saclay, Orsay, France
[2]LTCI, CNRS, Telecom ParisTech, Université Paris-Saclay, Paris, France
[3]Sorbonne Universités UPMC Univ Paris 06, CNRS, ISIR, Paris, France

ABSTRACT
TouchTokens are passive tokens that can be recognized on any capacitive surface based on the spatial configuration of the fingers that hold them. However, interaction with these tokens is confined to the basic two-state model of touch interaction as the system only knows the tokens' position and cannot detect tokens that are not touched. We increase the expressive power of TouchTokens by introducing laser-cut lattice hinges in their design, so as to make them flexible. A new recognizer, that analyzes the micro-movements of the fingers that hold the tokens, enables the system to detect when a token is left on the surface rather than taken off it. It can also detect *bend* events that can be mapped to command triggers, and a *squeezed* state that can be used for quasi-modal interaction.

ACM Classification Keywords
H.5.2 : User Interfaces - Input devices and strategies.

Author Keywords
Tangible interaction; Multi-Touch input; Micro-movements

INTRODUCTION
TouchTokens [9] provide a simple means to develop tangible interfaces. The approach relies on easy-to-make passive tokens that feature notches constraining how users grasp them. 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, users are limited in how they can manipulate these tokens, as is often the case with approaches based on capacitive sensing.

In this article, we aim at increasing the expressive power of TouchTokens by making the system able to detect: 1) when a token is left *on* or lifted *off* the surface, 2) when it is *squeezed* and 3) when it is *bent*. We achieve this without introducing any kind of instrumentation, thus preserving the simplicity of the original approach, which relies exclusively on passive tokens, and which works with any off-the-shelf capacitive surface. Our solution relies on the hardware side on making the tokens flexible by introducing lattice-hinges in their design, and on the software side on a novel recognizer that analyzes the micro-movements of the token-holding fingers that remain in contact with the surface.

After a short overview of related work, we describe the design of our flexible tokens, based on lattice hinges which can easily be obtained using fabrication processes such as laser cutting. We then report on a formative study in which we collected a sample of finger micro-movements that are representative of the manipulations afforded by our flexible tokens. Finally, we describe our recognizer, and evaluate its performance.

RELATED WORK
The most common approach to enabling tangible interaction on surfaces that use diffuse illumination technology consists in augmenting the objects with fiducial markers, and using a vision-based algorithm to identify them and track their location (see, *e.g.*, [5]). Other projects have investigated tangibles that reflect incoming light to the surface in a specific way in order to support more manipulations, such as TZee tangibles [14], which have the shape of a truncated pyramid and support gesturing on their sides, or Lumino blocks [1], which can be stacked. Diffuse illumination is a solution that is usually reserved to large setups such as tabletops.

Another approach involves augmenting tangibles with magnets. When coupled with a force-resistive screen, the system can detect pressure and gestures performed on top of the tokens [6]. When coupled with a surface augmented with a Hall sensor grid, the system can track tokens hovering over the surface [8]. GaussBricks [7], which also rely on a display equipped with Hall sensors, are bricks that can be assembled together to create larger objects featuring both deformable and rigid parts. While this approach enables very rich interactions, it requires augmenting the surface with specific sensors, and ensuring that the device's environment is free of any ferrous object that could interfere with the tangibles' magnetic field.

Solutions based on capacitive sensing are more affordable, but usually more limited. The system will often only be able to track the tokens that users are touching. There are, however, a few exceptions that go beyond these limitations. CapStones and ZebraWidgets [3] are capacitive units that can be assembled to configure different conductive circuits, enabling more manipulations with the tangibles that can, for example, be stacked or feature moving parts. PUCs [13] widgets rely on the principle of mutual capacitance so as to be detected even

CHI 2017, May 06-11, 2017, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/3025453.3025894

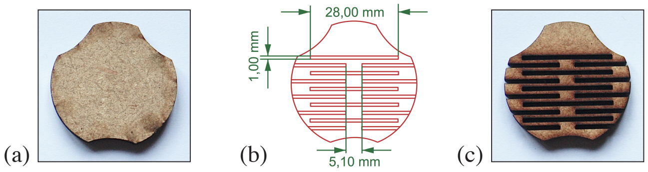

Figure 1. Making a TouchToken flexible: (a) original, rigid TouchToken (circle, 4cm in diameter), (b) schematics of lattice-hinges, (c) flexible TouchToken. Vector descriptions of all flexible TouchTokens available at https://www.lri.fr/~appert/touchtokens/index.html.

when users do not touch them. However, after a moment, PUCs get rejected by the adaptive filtering method of capacitive screens. To avoid this issue, PERCs [12] are equipped with sensors to capture the electrical field emitted by the capacitive screen, enabling them to know if they are on the surface or not, and communicate their state (*on* vs. *off* the surface) to the system via the Bluetooth protocol. Our contribution also aims at increasing the number of possible interactions with tokens but, as described in the next section, we do so without relying on any advanced design or embedded electronics.

MAKING TOUCHTOKENS MORE EXPRESSIVE

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 [2, 10]. This section describes our new tokens and introduces our hypotheses regarding the micro-movements we expect to observe.

Designing Flexible TouchTokens

Figure 1 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.

Detecting Tokens' *on/off* State

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 [4]). 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.

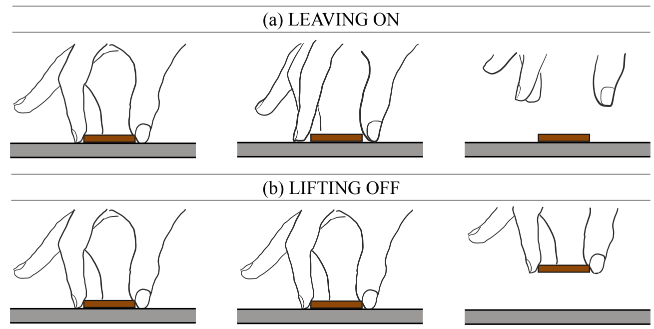

Figure 2. Finger micro-movements when leaving a token *on* the surface (a), and when lifting it *off* (b).

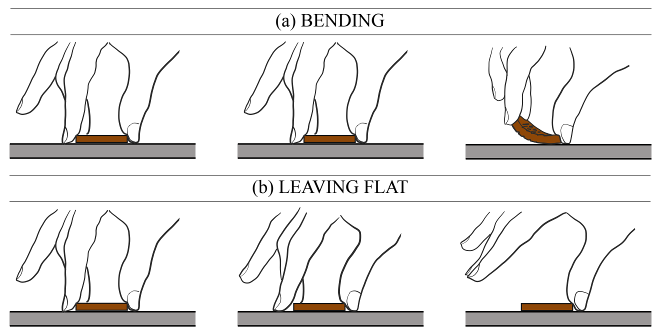

Figure 3. Micro-movements when (a) bending a token, (b) leaving it flat.

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 2 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.

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.

Bending Tokens

Bending a token leads to a state where users are keeping only one finger in contact with the surface (Figure 3-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 3-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.

COLLECTING TOUCH TRACES

We conducted a series of 3 experiments to collect multi-touch traces of users performing the three types of manipulations described above. Our goal was to refine hypotheses about the different finger micro-movements involved. We were particularly interested in the typical profile of point-to-centroid average distance time-series corresponding to these movements.

Participants & Apparatus

Twelve volunteers (2 female), 23 to 40 year-old (avg. 28.83, med. 28), participated in our experiment. 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.

Procedure

All participants ran the 3 experiments: *Click and Drag & Drop*, *Leave on vs. Lift off* and *Bend vs. Leave flat*, whose presentation order was counterbalanced using a Latin Square. All experiments involved the flexible version of the 6 TOKENS introduced in [9]: 2 circles, 2 squares, 1 triangle, 1 rectangle.

Experiment$_1$: *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.

Experiment$_2$: *Leave on vs. Lift off*. This task also involved moving 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 *Experiment*$_1$. We introduced an additional factor, FINGERCOUNT, to capture the two different manipulation styles described in [9]: 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

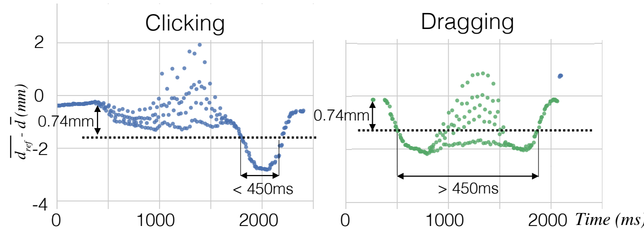

Figure 4. Using *Squeeze* mode for clicking (left) and dragging (right).

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.

Experiment$_3$: *Bend vs. Leave flat*. 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.

In each experiment, trials are first blocked by ACTION, then by DIRECTION within each ACTION (*Experiment*$_1$ and *Experiment*$_2$), and by FINGERCOUNT within each DIRECTION block (*Experiment*$_2$). 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.

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 hypothesis, 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) 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.

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

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

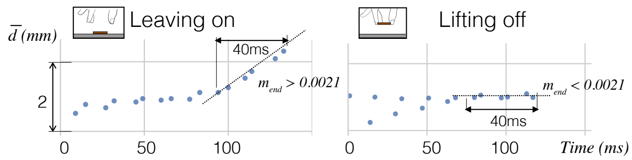

Figure 5. Leaving a token *on* the surface (left) or lifting it *off* (right).

where m_{end} is the slope[1] 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 5 illustrates the two cases.

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 6. 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.

RECOGNIZER PARAMETERIZATION

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 parameter 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 deviation).

Squeezed mode is recognized in **96.9%** (median: 97.9 / std: 3.0) of all trials in $Experiment_1$ (with $\mathbf{d_{sqz}} \in [0.74, 0.75]$ and $\mathbf{buffer_{sqz}} = 100$). It is falsely detected in 1.8% of all trials in $Experiment_2$, and 2.1% in $Experiment_3$.

[1]Computed using the Theil-Sen estimator [11].

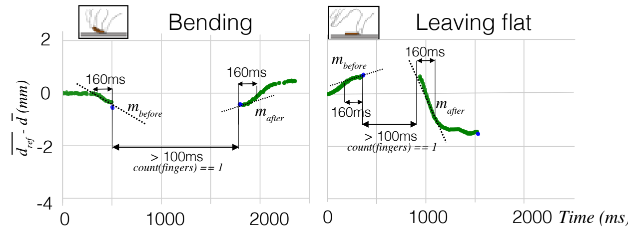

Figure 6. *Bending* a token (left) or leaving it flat (right).

States *on* and *off* were properly distinguished in **90.1%** (median: 92.4 / std: 5.1)[2] of all trials in $Experiment_2$ (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 in $Experiment_3$, with only 7.6% of false positives. However, when tested on trials from $Experiment_1$, 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.

In $Experiment_3$, *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 in either $Experiment_1$ or $Experiment_2$, 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*.

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.

CONCLUSION

As discussed in [9], TouchTokens can play different roles in an application. They can be used to control parameters or filter data in a visualization. They can be used as controllers in games, as data receptacles to hold any kind of content, and even as an access control mechanism. Our new events enable developing 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 them). This extended vocabulary can be used for different purposes, such as concurrently activating several filters, invoking commands on specific items or transferring data using drag-and-drop, click actions or contextual controls that take the tokens' relative layout into account.

[2]As a side note, we observed a recognition accuracy close to 90% for *on*/*off* states during informal tests using rigid tokens, suggesting that these micro-movements can also be detected on regular TouchTokens.

REFERENCES

1. Patrick Baudisch, Torsten Becker, and Frederik Rudeck. 2010. Lumino: Tangible Blocks for Tabletop Computers Based on Glass Fiber Bundles. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, 1165–1174. DOI: http://dx.doi.org/10.1145/1753326.1753500

2. David Bonnet, Caroline Appert, and Michel Beaudouin-Lafon. 2013. Extending the Vocabulary of Touch Events with ThumbRock. In *Proceedings of Graphics Interface 2013 (GI '13)*. Canadian Information Processing Society, 221–228. http://dl.acm.org/citation.cfm?id=2532129.2532166

3. Liwei Chan, Stefanie Müller, Anne Roudaut, and Patrick Baudisch. 2012. 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 (CHI '12)*. ACM, 2189–2192. DOI: http://dx.doi.org/10.1145/2207676.2208371

4. Hans-Christian Jetter, Jens Gerken, Michael Zöllner, Harald Reiterer, and Natasa Milic-Frayling. 2011. Materializing the Query with Facet-streams: A Hybrid Surface for Collaborative Search on Tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, 3013–3022. DOI: http://dx.doi.org/10.1145/1978942.1979390

5. Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: Exploring the Synergy Between Live Music Performance and Tabletop Tangible Interfaces. In *Proc. TEI '07*. ACM, 139–146. DOI:http://dx.doi.org/10.1145/1226969.1226998

6. Jakob Leitner and Michael Haller. 2011. 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 (CHI '11)*. ACM, 2985–2994. DOI: http://dx.doi.org/10.1145/1978942.1979385

7. Rong-Hao Liang, Liwei Chan, Hung-Yu Tseng, Han-Chih Kuo, Da-Yuan Huang, De-Nian Yang, and Bing-Yu Chen. 2014. GaussBricks: Magnetic Building Blocks for Constructive Tangible Interactions on Portable Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, 3153–3162. DOI: http://dx.doi.org/10.1145/2556288.2557105

8. Rong-Hao Liang, Kai-Yin Cheng, Liwei Chan, Chuan-Xhyuan Peng, Mike Y. Chen, Rung-Huei Liang, De-Nian Yang, and Bing-Yu Chen. 2013. GaussBits: Magnetic Tangible Bits for Portable and Occlusion-free Near-surface Interactions. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems (CHI EA '13)*. ACM, 2837–2838. DOI: http://dx.doi.org/10.1145/2468356.2479537

9. Rafael Morales González, Caroline Appert, Gilles Bailly, and Emmanuel Pietriga. 2016. TouchTokens: Guiding Touch Patterns with Passive Tokens. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 4189–4202. DOI: http://dx.doi.org/10.1145/2858036.2858041

10. Anne Roudaut, Eric Lecolinet, and Yves Guiard. 2009. 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 (CHI '09)*. ACM, 927–936. DOI:http://dx.doi.org/10.1145/1518701.1518843

11. Pranab Kumar Sen. 1968. Estimates of the Regression Coefficient Based on Kendall's Tau. *J. Amer. Statist. Assoc.* 63, 324 (1968), 1379–1389. DOI: http://dx.doi.org/10.1080/01621459.1968.10480934

12. Simon Voelker, Christian Cherek, Jan Thar, Thorsten Karrer, Christian Thoresen, Kjell Ivar Øvergård, and Jan Borchers. 2015. PERCs: Persistently Trackable Tangibles on Capacitive Multi-Touch Displays. In *Proc. UIST '15*. ACM, 351–356. DOI: http://dx.doi.org/10.1145/2807442.2807466

13. Simon Voelker, Kosuke Nakajima, Christian Thoresen, Yuichi Itoh, Kjell Ivar Øvergård, and Jan Borchers. 2013. PUCs: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-touch Displays. In *Proc. ITS '13*. ACM, 101–104. DOI: http://dx.doi.org/10.1145/2512349.2512791

14. Cary Williams, Xing Dong Yang, Grant Partridge, Joshua Millar-Usiskin, Arkady Major, and Pourang Irani. 2011. 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 (CHI '11)*. ACM, 1363–1372. DOI: http://dx.doi.org/10.1145/1978942.1979143