Twist ‘n’ Knock: A One-handed Gesture for Smart Watches

Vikram Kamath Cannanure*
Carnegie Mellon University, Human Computer Interaction Institute

Xiang ‘Anthony’ Chen†
Carnegie Mellon University, Human Computer Interaction Institute

Jennifer Mankoff‡
Carnegie Mellon University, Human Computer Interaction Institute

ABSTRACT

Interacting with a smart watch requires a fair amount of attention, which can disrupt a user’s primary activity. While single-handed gestures have been developed for other platforms, they are cumbersome to perform with a watch. A simple interaction is needed that can be used to quickly and subtly access the watch at the user’s convenience. In this paper, we developed *Twist 'n' Knock*—a one-handed gesture that can quickly trigger functionality on a smart watch without causing unintended false positives. This gesture is performed by quickly twisting the wrist that wears the watch and then knocking on a nearby surface such as the thigh when standing or a table when sitting. Our evaluation with 11 participants shows that by chunking the twisting and knocking motion into a combined action, Twist ‘n’ Knock offers distinct features that produced only 2 false positives over a combined 22 hours of real world collection (11 users for 2 hours each). In structured tests, accuracy was 93%.

Keywords: One-handed interaction; gesture; smart watch.

Index Terms: [User Interfaces]: Input devices and strategies, Interaction styles.

1 INTRODUCTION

Humans are adept at focusing on a central task while still effectively interacting at the periphery. From an activity theory perspective, it has been argued that operations, like turning a doorknob, or moving a mouse can be performed at the edges of our attention (these then are secondary activities) [9]. For example, it is possible to move the steering wheel without losing focus on the road while driving.

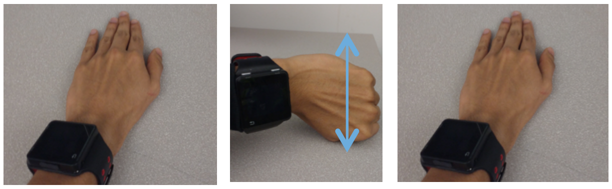


Figure 1: *Twist’n’Knock* is performed by making a fist and rotating the hand clockwise until the index finger can be tapped twice and then returning to stationary state.

Short interactions constitute almost half of the interactions with mobile phones [4], and are particularly appropriate to the periphery, especially if they are designed to minimize cognitive load [3]. These interactions may involve users glancing at a device to check the weather, email or skip a song in a playlist.

In some cases, even glancing is unnecessary: Hudson *et al.* show the user can respond to a cue on the waist-mounted device simply by quickly whacking it [7]. Another solution to this problem is a custom hands and eyes-free wearable device, such as the pendant-like devices as shown in *Sixth Sense* [11] or *Gesture Pendant* [21]. *Nailo* [13], the thumbnail trackpad and *Nod* [14] the wearable ring are commercial examples of such devices.

We build on this body of work by developing a gesture deployable on off-the-shelf watches that supports peripheral interaction. Because smart watches are attached to the wrist, it is critical to design a gesture that is unlikely to be triggered accidentally but still requires little attention to execute.

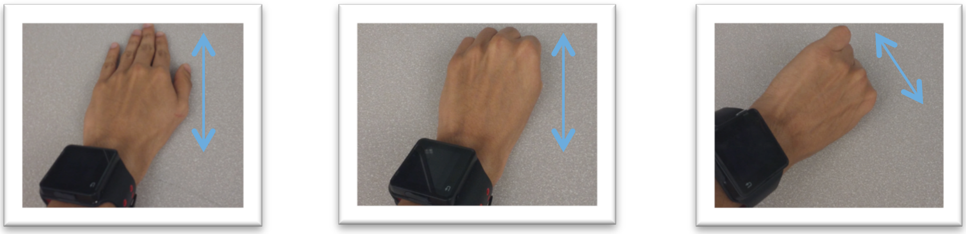
For this reason, commercial products often require repetition or exaggeration of actions such as twisting the device thrice [12] or moving the wrist quickly towards the chest [6]. Even with good levels of accuracy, these solutions are at times hard to perform.

We contribute a simple single-handed gesture, *Twist’n’Knock,* with an extremely low false positive rate and a recognition accuracy of 93%. *Twist’n’Knock* is performed by quickly twisting the hand and knocking on a nearby surface, such as the thigh when standing or the table when sitting.

Our vision for the gesture is to couple it with other devices, allowing a watch to provide an important channel of control on the periphery of the main task, similar to the steering wheel in driving. For example, if a smart watch user is interacting with a music player while standing in a crowded bus, the single-handed gesture can be used to trigger the music player or switch a song. For a laptop, a *Twist’n’Knock* gesture on table could be used for authentication in combination with other features like location and the mac address of the device. Another use case in a home could be triggering a Bluetooth enabled lamp through a gesture. These sample scenarios and application help us showcase ideal situations where a single-handed gesture can be used to interact with the digital world.

*email: kmarkiv@cmu.edu
†email: xiangchen@acm.org
‡email: jmankoff@cs.cmu.edu

Graphics Interface Conference 2016
1-3 June, Victoria, British Columbia, Canada



Tapping Knocking Twist 'n' Knock

Figure 2: The images above show a user performing Tap, Knock and *Twist'n'Knock on the table*.

Our evaluation demonstrates that *Twist'n'Knock* outperforms existing techniques, with only 2 false positives over a combined 22 hours of real world data (collected from 11 users for 2 hours each). In a structured study, accuracy was 93%. We also demonstrate the gesture's value in three scenarios that couple watch input with other devices such as a music player on the mobile phone, authentication system on a laptop and a smart lamp.

Next we present related work on low-attention gestures for mobile devices (specifically tablets and phones) as well as work on improving the interaction experience with smart watches. Following that, we discuss our implementation of twist and knock, alongside two other simple gestures used effectively with mobile devices (tapping and knocking). Section 4 describes the evaluation we conducted and its results.

2 RELATED WORK

Brief and simple interactions with mobile phones and tablets have been an active research area in recent years. *Whack* gestures allow the user to interact with a mobile device by striking it with the palm or the heel of the hand, even when in a pocket. This allows users to quickly trigger the device without disrupting their routine. By using a series of *Whacks* as a signalling gesture, false positives are kept to a minimum. Serrano *et al* [20] developed *Bezel tap* to quickly activate a tablet from sleep mode by tapping. *Shakeunlock* uses rapid shaking to unlock a device [9]**.** In *Pocket Touch,* Saponas *et al* [19] showcase a hardware prototype that enables interaction in the pocket through gestures**.** Their technique is able to detect gestures over the pocket but they found a need to be able to 'unlock' the device for operation in any orientation. While these gestures demonstrate the potential and need for simple ways of triggering common actions, they are not designed for or tested on smart watches.

In the domain of smart watches and related arm-worn devices, attention has primarily focused on increasing the quality of the interactive experience. In *Pinch Watch*, Loclair *et al* [8] introduce a wearable device that uses the implicit tactile sensation of the thumb to allow for hands free interaction. The main goal was to allow for effortless multitasking and their hardware addendum provided solution a single handed gestures as input. *WatchIt* [16] had a similar motivation with a use of a wristband with a potentiometer. They used the wristband as an input device to scroll content on a connected display. They found that their technique was useful in eyes free usage scenarios. Pasquero *et al* [15] designed a haptic wristwatch for eyes free interaction by using a piezo electric actuator. Their goal was to come up with interactions that blended in with the user's activities through tactile feedback. Recent work from *Office Smartwatch* [5] talks about scenarios in which a wearable can be used in an office environment. They use a simple forearm gestures coupled with a QR code for access control. They spoke in their limitation that their users found it hard to scan the code before performing the gestures.

A hardware limitation of smart watches is low sensor frequency, as the operating system limits the sensor data to reduce battery consumption. This makes it difficult to detect gestures. Thus, most commercial Smart Watches have long or quick gestures to trigger actions. Fitbit [6], Apple watch [1] and Galaxy Gear [18] have the gesture to quickly move the arm upwards as if you were seeing the time. MotoActv [12] requires a series of repeated jerks to activate the watch. These gestures give a reasonable accuracy but they may be hard to perform in all situations. For example, it may not be socially appropriate to jerk your hand in a crowded bus.

In conclusion, prior work on mobile devices suggests the value of a single-handed gesture that can be used to activate a device. Watch research however has not addressed this need to date. Existing watches accomplish this with exaggerated actions that may not be appropriate in all situations.

3 IMPLEMENTING TWIST AND KNOCK

To address the need to quickly access the smart watch with minimal attention, we developed a simple gesture (*Twist'n'Knock*) that works across different postures (standing vs. sitting).

Drawing from the gestures reviewed above that have been developed on mobile phones and tablets, we compare tapping (a singe touch), knocking (two light taps) to *Twist'n'Knock*. The gesture is implementable in software on an off-the-shelf smart watch and depends only on the 3-axis accelerometer.

We hypothesize that combining knocking and twisting into one single gesture can limit confusion with users' other common wrist motion. In this section, we describe the features and implementation of *Twist'n'Knock*, along with competing gestures (tapping and knocking) (see Fig. 2).

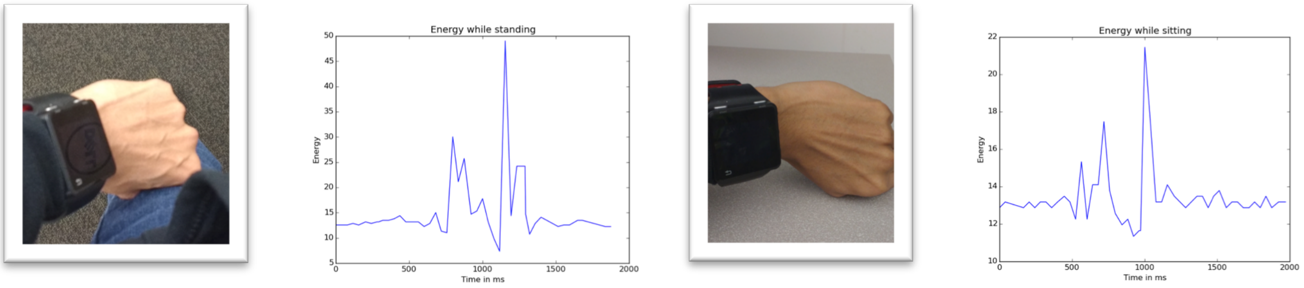


Figure 3: The graphs above show the sequence of energy for knocking segmented for a second while standing (left two images) and sitting (right two images)

3.1 Features

To facilitate the development of appropriate features, one author collected 4 hours of real world data while riding the bus, talking on the phone, typing, and exercising.

Although the accelerometer has a frequency of 25Hz, the watch records data at its highest frequencies based on frequent motion and it stops when the hand is stationary. This resulted in an average frequency of 6Hz in our data.

Based on this data, we extract *energy* (the sum of accelerometer readings) and rotation (the angle across acceleration vectors), which gives us *yaw, pitch* and *roll*. Most of the features were noisy because of jitter in the accelerometer readings. We cleaned energy with a high pass filter. An exponential filter was used to smooth yaw, pitch and roll.

3.2 Implementation of gestures

For completeness, and to enable comparison with prior work, we implemented recognition of three gestures: Knocking, Tapping, and *Twist'n'Knock*.

3.2.1 Knocking

A *knock* is detected based on accelerometer energy, specifically when two large spikes occurred within an interval of 1 second followed by a stationary state for half a second. Spikes during knocking are observed as large peaks. We detect them by subtracting current energy from a moving average during each timestep (3). Thresholds are determined by using trial and error in the pilot data. A regular knock should have an energy threshold of two standard deviations above the mean.

Fig 3 illustrates the knock feature performed by a user while sitting and standing after the data is normalized using the moving average and a low pass filter. Notice that there is more noise while standing because the user moves the wrist more between taps while standing.

Two or more spikes are used to identify a knock. The *knocking* gesture is recognized when such spikes occur in the absence of any *roll* around the X Axis (defined using an empirically determined threshold). We determined this by passing an energy filter on the roll to exclude any high roll cases.

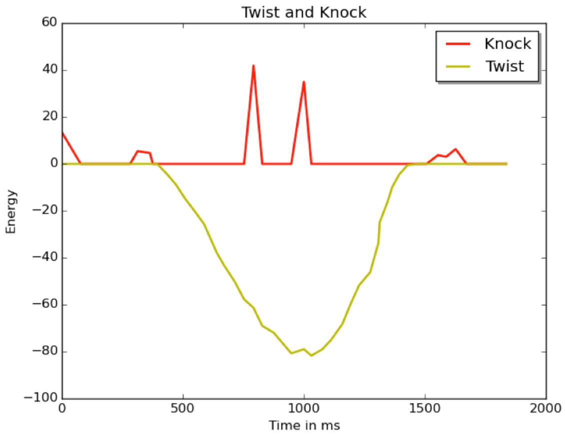


Figure 4: A visualization indicating twist (lower quadrant) and the spikes in a knock (upper quadrant)

3.2.2 Tapping

A gentle *tap* on the table or on the body is classified as a tap. Similar to Whack Gestures [7], a tap is based on a change in the exponential decay of the Z values. On tapping a surface, the hand experiences an opposite force in the reverse direction. This is dissipated to the watch and is noticed as a sharp spike in the acceleration. Since the watch face is outward, we notice the spikes in the Z direction.

We saw a high number of false positives in the pilot data, which is consistent with previous literature [17], which received 1.5 false positives per minute. Nonetheless, we retained this gesture for comparison because of its use in past work.

3.2.3 Twist'n'Knock

Twist'n'Knock is recognized when the two sharp spikes occur during a twist (overlapping it in time). Fig 4 illustrates the gesture. *Twist'n'Knock* is a conjoint gesture, which depends on independent recognition of twisting and knocking.

We have already described how knocks are identified, and the same algorithm is used for *Twist'n'Knock,* absent the constraint on rolling. *Twist'n'Knock* showed larger spikes than other knocks, which we hypothesize is because it involved work against gravity. Additionally, due to the twist, a knock in twist and knock involved work against gravity in the Y direction leading for large values of y. This was validated on a training set and we used this on top of the knock algorithm to further segment a twist.

Twist is determined using *roll*, calculated by measuring the angle between acceleration in Y and Z. This data is smoothed with an exponential filter with an empirically derived alpha value of 0.3. A sharp drop (shown in Fig 3) in roll can be used to extract a twist.

Table 1: Results of user study. Accuracy (Recall) was measured against our *gesture data*. False Positives were tested against the *raw data* collected in the field. Accuracy cannot be tested on that data because participants were not asked to do any gestures during their two hours in the field. Data was collected from a total of 11 participants.

	Recall (*gesture data*)	**Precision (*gesture data*)**	**False Positives (*raw data*)**
Tapping	91%	89.21%	2,559 (51/hr)
Knocking	92%	98.9%	264 (12/hr)
Twist'n'Knock	93%	100%	2 (0.1/hr)

Upon twisting, the user returns the arm to a natural state and this is anti clockwise motion, which when passed through a low pass filter leads to a sequence of empty values whose presence can be used to segment a twist (Fig 3). We determine the twist magnitude by integrating the values across the gesture, and segment it based on the empty values before and after the twist.

4 EVALUATION

To evaluate our solution, we conducted a user study with 11 participants (9 male, average age of 26, aged 20 to 40). Users were asked to wear the watch on their left hand (only one user was left handed). This is because most people wear their watch on the left hand. We asked the left-handed user to also use the left hand for consistency. They were instructed to tap once for tapping, knock twice for knocking, and twist in the clockwise direction and knock twice for *Twist'n'Knock*. The natural anticlockwise direction is easier to perform on the table but isn't convenient while standing.

We conducted the experiment in two phases. In the *raw data* phase, the participant was asked to wear the watch for 2 hours and to go about their daily activities without explicitly doing any gestures. The raw data was expected to contain no relevant gestures, and was used to test for false positives. Each participant engaged in a diverse range of actions, which gave us a wide variety of scenarios that can be tested. Some of the tasks that participants performed were lecturing, walking, and running, typing and reading. We used this real world data to look for accidental triggers.

In the second phase, to collect *gesture data*, participants were asked to perform the three gestures, tapping, knocking and *Twist'n'Knock,* in two conditions (sitting and standing). They were directed to perform these gestures on the thighs while standing and on a table while sitting (Fig 3). The protocol was to repeat the 3 gestures at an interval of 5 seconds for 10 trials each. Thus each user conducted 10 trials x 3 gestures x 2 conditions, giving a total of 60 data points per user. We used this gesture data to train and test our recognizer's accuracy.

4.1 Analysis

We split the *gesture data* randomly, by user, into three unequal samples for training, validation and test. Data from three users was selected for training, three for validation, and five for testing. We used the training set to determine initial values for the filters. We used the validation set to determine the accuracy of these preliminary values. Upon reaching a high accuracy, we combined the training and validation to determine the final thresholds.

We used the same thresholds for both sitting and standing. In some cases, like tapping, the magnitude of taps varied largely between sitting and standing and then we used one deviation below the mean for the threshold. We used the final thresholds to evaluate accuracy on the test data of 5 users. The final threshold was also used to determine the false positives in the *raw data* collected for each user.

4.2 Results

We obtained an accuracy of 91% for tapping, 92% for knocking and 93% for *Twist'n'Knock* on the test data, as reported in the first column of Table 1 with a kappa of 0.835.

When tested against the *real world data*, the *Twist'n'Knock* recognizer recorded only 2 false positives in the entire dataset of 22 hours across all 11 participants, both for a single user. She had mentioned that she had removed the watch while taking a shower. We speculate that this could have triggered the activity. There were no false positives in the rest of the data even when the user took notes in left hand or presented to an audience.

In contrast, tapping had a false positive rate of 51/hr (total of of 2,559 false positives) and knocking a false positive rate of 12/hr (total of 264 false positives). We noticed that tapping was triggered in outdoor activities or in animated indoor actions like giving a presentation. Knocking provided lesser false positives compared to tapping but even 12 false positives per hour is impractical.

False positives are important because they could accidentally trigger an action when it is not intended. We attribute *Twist'n'Knock's* success to the fact that the initial twist acts as a filtering mechanism to prevent accidental triggers. Perhaps this is because twisting the wrist in the clockwise direction is unnatural (although not uncomfortable).

From a qualitative perspective, our users mentioned that the action was strange to them when performed on a table. However, they did not report that it was uncomfortable, and did not have difficulty performing the activity repeatedly for ten trials both while sitting and standing.

4.3 Discussion and Limitations

Our algorithms have been evaluated for standing and sitting which are the common scenarios we assumed as per our use case. We found that (among other things) standing *versus* sitting at the table affected magnitude. In addition, gestures performed while standing were more prone to noise as there is less support for the hand, illustrated in the graph on Fig 3. The surface being used also seemed to affect how hard people would tap – they would tap harder on a softer surface such as the body.

Our watch model had a small hardware flaw that wrongly triggered a stationary state between gestures. This occurred more for Knock and Tap gestures because they required very little motion. To address this, users were asked to shake their hand before starting the experiment.

Our experiment demonstrates that *Twist'n'Knock* performs very well in terms of false positives, and strongly in terms of accuracy. However, users reported that it felt strange to them specifically while performing the gesture on the table (though not uncomfortable). A more common, intuitive gesture such as knocking or tapping has a very high false positive rate, making it impractical in a field deployment. *Twist'n'Knock* provides low false positive rate and users were able to perform it successfully after training.

We asked the users to rotate the wrist at 90 degrees when performing while sitting and standing for consistency. Future work could explore other positions, or support variable positions depending on context. This could improve the ergonomic aspects of the gesture. Additionally, user specific model can be built based on additional features like location and sound to improve accuracy in the real world.

5 CONCLUSION

Everyday interactions with common objects are performed in the periphery of one's attention. This helps keep focus on the central task and simultaneously perform our daily actions. Interaction with mobile devices disrupts the user's attention, as it requires considerable fine motor and visual attention.

In this paper, we described *Twist and Knock* a simple single-handed gesture to trigger an action on a smart watch. Since the gesture can be performed on the body or an object, it demands very little attention from the user.

We demonstrated that using common motions in combination can reduce false positives in the field. Our gesture was based on the combination of knocking and twisting in an uncommon but not uncomfortable manner. Adding an uncommon combination of motions acts as a filter and greatly prevents accidental triggers.

Our evaluation demonstrated that *Twist'n'Knock* is far superior to other options, with 2 false positives over our large field data set and with accuracy 93%. Other gestures are impractical in terms of false positives and less accurate.

For future work we would like to experiment with other gestures that have very low false positive rates. Adding multiple gestures could allow the user to trigger multiple different actions. We also plan to implement several applications and explore the use of *Twist'n'Knock* in a true field deployment.

6 ACKNOWLEDGEMENTS

We would like to thank all participants for their time for taking part in the study. We also thank the support staff that contributed to the previous versions of the paper. This work was funded in part by NSH IIS 1217929.

7 REFERENCES

[1] Apple Watch (2015). http://www.apple.com/watch.

[2] Ashbrook, D. (2010), Enabling mobile microinteractions. PhD Thesis. (2010), Georgia Institute of Technology.

[3] Bakker, S., van den Hoven, E. and Eggen, B. (2014). Peripheral interaction: characteristics and considerations. Personal and Ubiquitous Computing 19(1):239-254.

[4] Banovic, N., Brant, C., Mankoff, J., and Dey, A. K. (2014) ProactiveTasks: the short of mobile device use sessions. In *Proceedings of Mobile HCI 2014:* 243-252

[5] Bernaerts, Y., Druwa, M., Steensels, S., Vermeulen, J. and Schaning, J. (2014). The office smartwatch. In *Proceedings of Designing Interactive Systems (DIS 2014)*. pp. 41-44.

[6] Fitbit.com. (2015). Fitbit official site for activity trackers & more. http://www.fitbit.com/.

[7] Hudson, S., Harrison, C., Harrison, B. and LaMarca, A. (2010). Whack gestures. In *Proceedings of Tangible and Embedded Interaction* (*TEI '10)*. 109-112.

[8] Loclair, C., Gustafson, S. and Baudisch, P. (2010). PinchWatch: a wearable device for one-handed microinteractions. In *Proceedings of MobileHCI*. 2010.

[9] Findling, R. D., Muaaz, M., Hintze, D., Mayrhofer, R. (2014). Shakeunlock: Securely unlock mobile devices by shaking them together. In *Proceedings of MoMM 2014,* pp. 165-174.

[10] Matthews, T., Rattenbury, T., and Carter, S. (2007) Defining, designing, and evaluating peripheral displays: An analysis using activity theory. *Human-Computer Interaction* 22(1-2): 221-261 (2007)

[11] Mistry, P., and Maes, P. (2009). SixthSense: a wearable gestural interface. In *Proceedings of SIGGRAPH ASIA 2009* Sketches (p. 11).

[12] Motoactv.com, 2015. MOTOACTV - Motorola Mobility, Inc. USA. https://motoactv.com/.

[13] Nailo. http://nailo.media.mit.edu/

[14] Nod. https://nod.com/

[15] Pasquero, J., Stobbe, S. and Stonehouse, N. (2011). A haptic wristwatch for eyes-free interactions. In *Proceedings of the Conference on Human Interaction (CHI '11)*. pp. 3257-3266.

[16] Perrault, S., Lecolinet, E., Eagan, J. and Guiard, Y. (2013). Watchit. simple gestures and eyes-free interaction for wristwatches and bracelets. In *Proceedings of the Conference on Human Computer Interaction (CHI 2013*). Pp.1451-1460

[17] Ronkainen, S., Häkkilä, J., Kaleva, S., Colley, A. and Linjama, J. 2007. Tap input as an embedded interaction method for mobile devices. In *Proceedings of Tangible and Embedded Interaction (TEI '07),* pp. 263-270.

[18] Samsung Electronics America,. 2015. http://www.samsung.com/us/mobile/wearable-tech.

[19] Saponas, T., Harrison, C. and Benko, H. (2011). PocketTouch: through-fabric capacitive touch input. In *Proceedings of User Interface Systems and Technology (UIST 2011)*. pp. 303-308

[20] Serrano, M., Lecolinet, E. and Guiard, Y. (2013). Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. In *Proceedings of the Conference on Human Computer Interaction (CHI 2013)*. pp. 3027-3036

[21] Starner, T., Auxier, J., Ashbrook, D., and Gandy, M. (2000). The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *Proceedings of ISWC' 00*. pp. 87-94.