# Turn-Based Gesture Interaction in Mobile Devices

Sanna Kallio[1], Panu Korpipää[1], Jukka Linjama[2], and Juha Kela[1]

[1] Finwe Ltd, Elektroniikkatie 8, 90570 Oulu, Finland
[2] Senseg, Valimotie 27, 00380 Helsinki, Finland
{sanna.kallio,juha.kela,panu.korpipaa}@finwe.fi,
jukka.linjama@senseg.com

**Abstract.** When properly designed, gesture interaction can bring usability benefits to mobile device users. This article introduces a new accelerometer-enabled method of mobile device gesture control, namely the turn-based interaction. Turning is a commonly understood concept in interaction with tangible objects. Applied in mobile devices as an abstracted virtual key command, it extends the variety of potentially useful gesture control use cases. This article compactly explains the essential factors of designing a successful sensor-enabled gesture interface for mobile devices. In the light of these design factors, two example use cases of turn-based interaction with a prototype are presented. The reliability of recognizing a turn gesture is verified quantitatively to confirm that the introduced method is technically feasible.

**Keywords:** Gesture interface, acceleration sensors, turn, gesture interaction, haptic interaction, feedback.

## 1 Introduction

Advances in the research of sensor technologies in the last decade have led to their deployment in variety of application domains. As mobile devices are used in a diverse and dynamic context, development of new sensor-enabled applications and, especially, interaction methods has lately been a subject of great interest. As a result, the mobile computing community has recently witnessed large-scale deployment of smartphones applying sensor-enabled gesture control as a complementary user interaction modality [1].

### 1.1 Gesture Input

Generally, movement-based interaction can bring several advantages to the user of a mobile device in use cases where traditional modalities are insufficient [2], [3], [4]. Gesture control is eyes-free, button-free, and silent. The user is not required to see the keyboard or display to interact. The modality is "keypad lock"-free, which is a specifically beneficial property in mobile phones. The user does not have to look at the phone, open the keypad lock, and press buttons to navigate and perform a control action. Moreover, gestures can be performed while wearing gloves.

S. Hailes, S. Sicari, and G. Roussos (Eds.): S-Cube 2009, LNICST 24, pp. 11–19, 2009.

Traditionally, many gesture recognition systems have been based on visual recognition. Although more and more mobile phones are equipped with camera, this approach has limited potential in mobile context. Another emerging approach is to use position and/or acceleration sensors embedded into mobile devices themselves. Acceleration sensors indicate the motion of the device in one to three dimensions. Thus, user can move the device and perform some controls by gestures [5]. Acceleration sensors measure both dynamic and static acceleration and can thus also be used to implement, e.g., tilt control.

### 1.2 Related Work

Common examples of small to medium-scale types of gestures, captured using acceleration sensors, include shaking the device [6], and swinging it from side to side [7]. However, both of these interaction methods can be considered quite noticeable to other people, regardless of scale. Tapping control, in turn, stands for a minimalist extreme in hand gestures for interacting with mobile devices [8], [9]. Simple accelerometer-based tilting and orientation controls have been discussed in the literature in many studies over the years [10], but also more recently, e.g. combining tilting with vibrotactile feedback [11], and switching between landscape and portrait display orientations [12], a use case that has gained a lot of commercial popularity lately. Tilting and orientation are unobtrusive, and very simple to implement movement-based interaction types applicable to well-selected use cases in mobile computing. Tilting combined with a set of free form gesture commands can also be used to control external entities such as a 3D design studio [4]. However, as far as is known, the literature to date has not addressed simple turning movement as a separate type of method for gesture interaction.

### 1.3 Aims

This study addresses the relevant factors in designing successful sensor-enabled gesture input and, on this basis, introduces new gesture interaction modality, namely, turn-based interaction. The most important design factors are the user effort in performing gestures, reliability of recognition, clarity of function, social acceptability [8], feedback during interaction, multimodality, and well-selected use cases that match the metaphors behind the interaction. Based on the addressed design factors, turn-based interaction is introduced and two use cases are presented. Method for detecting turn-gestures using acceleration sensor is contemplated and discussed in the light of experience. The emphasis is on the user-friendly timing of acceleration sequence as well as on the feedback design. To validate that the gesture detection method is feasible in practice, the reliability of the method is quantitatively evaluated based on collected user data. Finally, a video demonstration is given to illustrate the application of turn gestures with example use cases [13].

## 2 Guidelines of Gesture Interaction Design

Complementary to the commonly applied usability criteria [14], gesture interaction has specific characteristics that affect the flavor of design parameters. Experience of movement-based interaction design has indicated that at least the following guidelines need to be considered in order to reach a properly balanced outcome.

1. User effort: the user task should become more effortless to perform than before, and the user experience should improve.
2. Multimodality: the gesture should be provided in addition to the traditional modality, if any, to perform the same task.
3. Reliability: it should not be possible to perform the gesture command by accident when not intending to do it, and when intending to perform the gesture, it should be consistently recognized correctly.
4. Feedback: it should be clearly, but not too obtrusively, indicated to the user whether the gesture was performed successfully.
5. Clarity of function: if there are multiple different gestures available in a device, their function should be clearly distinguishable from each other and clearly communicated to the user (e.g. by providing different feedback content).
6. Social acceptability: gestures should be as unnoticeable as possible, i.e. small in spatial scale [3].
7. Use case: the use of the new modality for a selected task should bring benefit to the user in terms of usability or joy of use.

## 3 Turn-Based Interaction

Turning an object is a commonly understood concept in interaction with tangible objects and is familiar to most people. The main contribution of this paper is in describing and evaluating a type of mobile device gesture interaction that is based on turning the device with regard to gravity.

Turn-based interaction is different from the common tilting and orientation-based interaction described in the literature [10], [11], [12]. The main difference is that an abstracted turn gesture is applied as a single discrete command instead of controlling an application with events from tilt angles or related orientation states. A single abstracted turn gesture command consists of a sequence of turn movements and orientations. It is important to handle a turn-based gesture as a single abstracted entity since this facilitates using turn gestures in a mobile device as virtual key-press commands that can be easily connected to perform various actions, in addition to the existing input methods. Moreover, as a separate abstraction, turn gestures can potentially be applied together with, and additionally to, other types of gestures. Table 1 presents a categorization of mobile device movement interaction types to clarify the main differences of the methods.

The suitability of movement interaction to a selected use case strongly depends on the type of gesture applied. Increased variety in the types of available common gestures enables mobile device usage to benefit from the new modality potentially more widely than before.

As a type of gesture, compared to the others in Table 1, turning has the primary advantage of being very effortless for a user to perform. It does not require accurate aimed motion from the user, nor excess concentration and attention focus. From a technical perspective, turn gestures can be recognized with a low event-based sampling rate, thus facilitating low power operation, which is essential in mobile devices containing limited battery resources.

**Table 1.** Categorization of movement interaction types applied in mobile devices

| Gesture type | Control type | Movement characteristics | Movement scale |
|---|---|---|---|
| Tilting | Stream of angles | Aimed angle | Small |
| Orientation | Discrete state | Keep/change state | Small |
| Free-form trainable | Discrete event | Match form | Small to large |
| Tapping / knocking | Discrete event | Aimed event | Tiny |
| Shaking / Swinging | Stream or discrete | Coarse match form | Medium to large |
| Turning | Discrete event | Coarse match form | Small |

## 4 Turn Gestures with Example Use Cases

Following the addressed design guidelines, two use cases were selected to demonstrate the use of the interaction method. Two gestures that utilize a turning movement of the device are applied for the selected tasks. Turn gestures function as virtual keys, which are named *TurnDown* and *DoubleTurn*, Figures 1 and 2 respectively. Virtual keys can be connected to any action in the mobile device, similarly to tapping commands in the Nokia 5500 phone [1]. Use cases 1 and 2 illustrate the use of turn gestures to control two different common tasks of a mobile phone user.

### 4.1 Use Case 1

In use case 1, *TurnDown* is applied to mute the ringing of the phone. The procedure of the scenario for *TurnDown* in Figure 1 is the following:

1. The phone is “ringing”.
2. The user turns the phone display up, then display down and holds it still - sound and vibra feedback notify the recognized *TurnDown* gesture.
3. Ringing tone is muted.

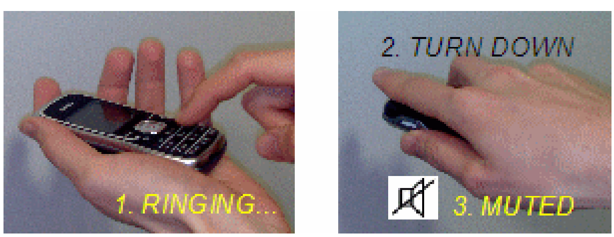


**Fig. 1.** Use case 1: *TurnDown* gesture use. Turn the phone display down to mute ringing.

### 4.2 Use Case 2

In use case 2, *DoubleTurn* is used to switch the screen lights on. The procedure of the scenario for *DoubleTurn* in Figure 2 is the following:

1. The user starts the gesture from display up and then turns the display down.
2. The user turns the display back up - sound and vibra feedback notify recognized *DoubleTurn* gesture.
3. Device wakes up and turns the display light on.

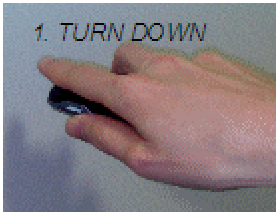



**Fig. 2.** Use case 2: DoubleTurn gesture use. Turn the phone display down and then up to switch the screen light on.

## 5 Input Methods

Two specific turn gestures, *TurnDown* and *DoubleTurn*, are used for controlling the device. The turn gestures are detected using accelerometer built inside the mobile device (STMicroelectronics, type LIS302DL) and Nokia sensor API is utilized to get data from the embedded sensors.

The algorithm designed is simply based on applying a threshold to the acceleration value within a time interval window. Performance optimization and tuning is done case-by-case. Basically, the *TurnDown* is a simple common movement pattern that could easily occur also in other than intended situations. However, in this case, the gesture detection needs to be active only when the phone is ringing. Binding the gesture to the controlled situation restricts misrecognitions. As a result, observing a single threshold crossing within a certain time interval window provides a reliable recognition in the use case 1.

In the use case 2, the detection must always be active. This increases the likelihood of false positives resulting from other daily user activities similar to the *DoubleTurn* gesture pattern. To develop, improve and optimize *DoubleTurn* algorithm, an extensive dataset was collected and analyzed.

## 6 Algorithm Optimization and Evaluation

The importance of reliability depends on a use case. When a gesture is used for a function that should never occur by accident, e.g. opening a keypad lock or calling, the corresponding gesture detection must have very high reliability. In practice, this is such a demanding requirement that these kinds of critical actions should not be selected to be controlled with gestures at all.

In the example use cases, false positives are not critical. In the use case 1, detecting the *TurnDown* gesture is only active while the phone is ringing. This narrows the application scope so that misrecognitions are only inconvenient. In the use case 2, the possible misrecognitions wake up the phone and turn the screen light on, which is inconvenient at most. However, should there be too frequent false positives, the battery consumption increases and the unintentional display light activations may become unwanted. More importantly, the gesture should be detected correctly when the user, any user, does intend to perform it (true positive), even if the user is simultaneously performing another movement activity, such as walking.

## 6.1 Data Collection

To address the reliability, user data was collected by logging the acceleration stream with phone logger application. The collected data was analyzed and the results were used to optimize and confirm the reliability of *DoubleTurn*. Turn gesture data was collected from 10 users, 6 female and 4 male. The gestures were performed while standing still and walking. The dataset contained 200 gesture repetitions in total. To analyze the occurrence of false positives, data containing movement activities (walking, climbing stairs, jogging, roller-skating) was collected from 8 users, in total 96 minutes. Figure 3 presents samples of 3-axis turn gesture data.

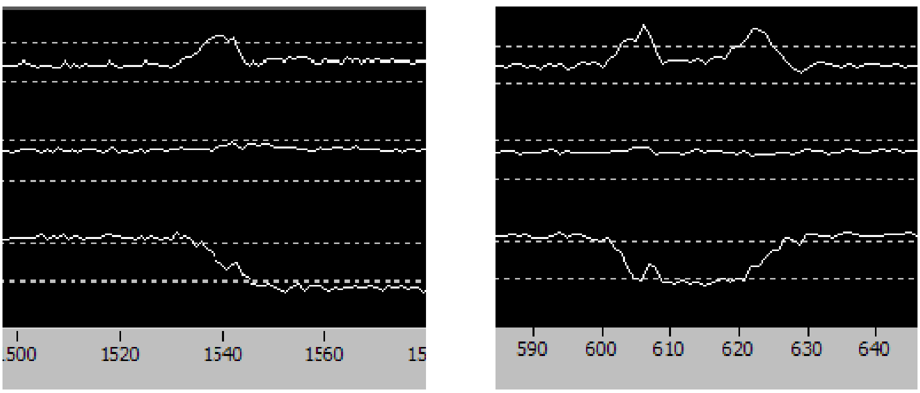


**Fig. 3.** An example of a *TurnDown* (left) and *DoubleTurn* (right) acceleration (X,Y,Z) signal trace performed in a stationary situation, applied to recognize the gestures

## 6.2 Algorithm Optimization

To minimize false positives during activity and eliminate the impact of user variation to *DoubleTurn* gesture, the algorithm was optimized based on the collected dataset. Firstly, to capture the variation on how users tend to perform the *DoubleTurn* gesture, the timing profile of the gesture was calculated from the dataset. Timing information was used to analyze how fast people generally perform the turn gestures. The recognition algorithm was then optimized accordingly so that the gesture is recognized independent of the user. Secondly, the acceleration data from activity dataset was carefully analyzed so that the type of the movements causing false positives could be identified. To further examine and validate the algorithm against these false positives, a new data set consisting of false positive data was collected. Finally, an additional maximum threshold limit was implemented to decrease the false positives during activities.

### 6.3 Recognition Results

Table 2 shows recognition results for basic dataset consisting of *DoubleTurn* gestures performed while standing still and walking. When standing still, a user-independent recognition accuracy is 100% and when walking and performing the *DoubleTurn* gesture concurrently, the recognition accuracy is 96%. The total recognition accuracy is 98%. The occurrence of false positives was calculated from the activity dataset. During 96 minutes, only 8 false positive occurred. Most of the false positives occurred during jogging and roller-skating.

**Table 2.** Recognition accuracy of *DoubleTurn* gesture

| Situation of performing *DoubleTurn* | User-independent average true positive % |
|---|---|
| Standing still | 100 |
| Walking | 96 |
| Total | 98 |

## 7 Feedback Design

In gesture interaction it is important to deliver information on the result of the performed control action to the user; otherwise, the user does not know whether the gesture was detected and may repeat it. Because the gestures are captured by acceleration sensor, they can also be performed eyes-free. As a concequence, a visual feedback may not be the best option.

The interaction feedback in the example use cases was designed to examine whether multiple feedback elements add value to the interaction. In addition to the feedback from the phone function performed as a result of a successful gesture, tactile (vibration) and sound feedback was applied to enforce the message.

The aim in the feedback indicator content design was minimalism. Subtle short vibration pulses were applied with rhythm as a parameter. The key benefit with this approach is that with proper content design, the same rhythm can be rendered with multiple modalities (touch, hearing, vision). Thus, in different contexts, when one particular modality is not available, the intended feedback metaphor can still be perceived.

## 8 Discussion

The results of the experiences are qualitatively discussed in this section. A formal usability study is beyond the scope of this study. Thus, for factors other than reliability, the discussion is based on the experiences gathered informally from the development team during the interaction design process, and on the prior experience of the authors.

### 8.1 Use Cases

Gestures can only be used for a restricted set of carefully selected control tasks, and the type of gesture should be selected to fit and benefit the task. Failure in matching the gesture type and the target task will lead to user confusion and disturbance, at the least. The selected example use cases demonstrated how turn gestures benefit the user by relieving the attention focus through button-free interaction. Multimodal operation allows using the gestures alternatively to the traditional way of performing the tasks.

### 8.2 Reliability

The reliability evaluation confirmed that the *DoubleTurn* gesture can be recognized user independently with a high accuracy, even when the user is on the move. False positives occurred infrequently enough, having an insignificant effect on power consumption and on the user experience.

### 8.3 Feedback Indicators

During the feedback development process, four users experimented with vibration pulses in various phases of interaction. The sound pulses were then added, experimenting with different pitch and character. An essential goal of the feedback content design was that it is not disturbing, but still clearly noticeable. To help the user distinguish the two gestures, a different number and rhythm of feedback pulses was provided for each gesture.

It was found that when sound and vibration are in synchrony, they enhance each other. The actual vibration pulse details are not important, as sound, when perceivable, grabs the attention. Vibration, on the other hand, adds to the perception of the sound as being more distinct and clear, compared to having just subtle sound without vibration.

## 9 Conclusion

Turn-based interaction was presented as a new type of method for gesture interaction with mobile devices. The accelerometer-enabled method was introduced and evaluated through example use cases and related user experiences. An essential element in the design was providing feedback indicators during interaction. The reliability of recognizing a turn-based gesture was verified quantitatively with user data. Turn-based interaction can benefit the mobile device user by extending the variety of usable gesture control use cases, potentially enhancing the user experience.

## References

1. Nokia Corporation. 5500 phone (2006), `http://europe.nokia.com/5500`
2. Pirhonen, A., Brewster, S., Holguin, C.: Gestural and Audio Metaphors as a Means of Control for Mobile Devices. In: Proc. CHI 2002, pp. 291–298. ACM Press, New York (2002)

3. Brewster, S., Lumsden, J., Bell, M., Hall, M., Tasker, S.: Multimodal 'eyes-free' interaction techniques for wearable devices. In: Proc. SIGCHI conference on human factors in computing systems, pp. 473–480. ACM Press, New York (2003)
4. Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., Di Marca, S.: Accelerometer based gesture control for a design environment. Personal and Ubiquitous Computing, 1–15 (2006); Online First Springer
5. Linjama, J., Häkkilä, J., Ronkainen, S.: Gesture Interfaces for Mobile Devices – Minimalist Approach for Haptic Interaction. Position paper in CHI 2005 Workshop Hands on Haptics, Portland, Oregon, April 3-4 (2005)
6. Levin, G., Yarin, P.: Bringing sketching tools to keychain computers with an acceleration-based interface. In: Proc. CHI 1999, pp. 268–269. ACM Press, New York (1999)
7. Sawada, H., Uta, S., Hashimoto, S.: Gesture recognition for human-friendly interface in designer - consumer cooperate design system. In: Proc. IEEE International Workshop on Robot and Human Interaction, pp. 400–405 (1999)
8. Linjama, J., Kaaresoja, T.: Novel, minimalist haptic gesture interaction for mobile devices. In: Proc. NordicCHI 2004, pp. 457–458 (2004)
9. Ronkainen, S., Häkkilä, J., Kaleva, S., Colley, A., Linjama, J.: Tap input as an embedded interaction method for mobile devices. In: Proc. TEI 2007, pp. 263–270. ACM Press, New York (2007)
10. Rekimoto, J.: Tilting operations for small screen interfaces. In: Proc. ACM Symposium on User Interface Software and Technology, pp. 167–168 (1996)
11. Oakley, I., Ängeslevä, J., Hughes, S., O'Modhrain, S.: Tilt and Feel: Scrolling with Vibrotactile Display. In: Proc. Eurohaptics, pp. 316–323 (2004)
12. Hinckley, K., Pierce, J., Horvitz, E., Sinclair, M.: Foreground and background interaction with sensor-enhanced mobile devices. ACM Transactions on Computer-Human Interaction 12(1), 31–52 (2005)
13. Video available, `http://www.finwe.fi/video`
14. Shneiderman, B., Plaisant, C.: Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4th edn., p. 672. Addison-Wesley, Reading (2005)