



PDF Download  
2470654.2470693.pdf  
31 March 2026  
Total Citations: 20  
Total Downloads: 828

 Latest updates: <https://dl.acm.org/doi/10.1145/2470654.2470693>

RESEARCH-ARTICLE

## Indirect shear force estimation for multi-point shear force operations

**SEONGKOOK HEO**, Korea Advanced Institute of Science and Technology, Daejeon, South Korea

**GEEHYUK LEE**, Korea Advanced Institute of Science and Technology, Daejeon, South Korea

**Open Access Support** provided by:

**Korea Advanced Institute of Science and Technology**

**Published:** 27 April 2013

[Citation in BibTeX format](#)

CHI '13: CHI Conference on Human Factors in Computing Systems  
April 27 - May 2, 2013  
Paris, France

**Conference Sponsors:**  
SIGCHI

# Indirect Shear Force Estimation for Multi-Point Shear Force Operations

Seongkook Heo, Geehyuk Lee

Department of Computer Science, KAIST

Daejeon, 305-701, South Korea

seongkook@kaist.ac.kr, geehyuk@gmail.com

## ABSTRACT

The possibility of using shear forces is being explored recently as a method to enrich touch screen interaction. However, most of the related studies are restricted to the case of single-point shear forces, possibly owing to the difficulty of independently sensing shear forces at multiple touch points. In this paper, we propose indirect methods to estimate shear forces using the movement of contact areas. These methods enable multi-point shear force estimation, where the estimation is done for each finger independently. We show the feasibility of these methods through an informal user study with a demo application utilizing these methods.

## Author Keywords

Shear force, multi-touch, force sensing, touch screen

## ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces – Input devices and strategies

## INTRODUCTION

A display with force vector detection capability was presented by Herot and Weinzapfel [6] in 1978 and also by Minsky [11] in 1984. With increased interest in touch screen interaction, the possibility of using force vectors has been revisited to increase the input dimensions of touch screen interaction. Heo and Lee [4] presented the concept of Force Gestures, which use normal and shear forces to enrich touch gestures. Harrison and Hudson [3] presented use scenarios that can be enabled, only if shear forces are used. Lee et al. [10] conducted an experiment to study the performance of using shear forces on a touch screen. However, the touch screen prototypes used in these previous studies were capable of sensing shear force at a single location only. Therefore, the use of multiple shear forces, such as pinching with shear forces, has not yet been studied.

Multi-touch input operations, such as pinching or rotating, are now commonly available on most smartphones and tablet computers. These multi-touch operations may be

further extended and enriched when a sensing method for shear forces at multiple points becomes possible. In order to detect shear forces at multiple touch locations, we could use a camera-based method that can detect local shear forces [9]. Further, as introduced by Holz and Baudisch [8], fingerprint sensors can be used to accurately measure finger deformations to estimate shear forces for each finger. However, camera-based methods and fingerprint sensor methods are not feasible for touch screen mobile devices, as they require space for a camera or are difficult to integrate with a display.

For multiple touch locations, we propose a method to estimate shear forces applied on the touch screen, instead of directly measuring shear forces. When a finger applies shear force on a screen, the fingertip is deformed because of the friction between the fingertip and the surface. Because of the deformation of the fingertip, the center of the contact area shifts slightly toward the direction of the shear force. In order to use this shift to estimate the shear force, one has to be able to discern whether the shift is due to the deformation of the fingertip or due to an actual finger movement. In this paper, we describe two different solutions to this problem.

## RELATED WORK

Various properties of touch, such as the force, velocity, and size of contact, are used to augment touch interfaces. One commonly studied feature is the normal force applied to the screen. Normal force is one of the key pieces of information that a tablet stylus provides; therefore, a majority of studies on normal force are based on a stylus interface [13, 14]. For a finger-based touch screen, Miyaki and Rekimoto [12] attached a force-sensitive resistor on the back of a mobile device and used vertical force for continuous zooming and scrolling. Rosenberg and Perlin [16] introduced a touch device capable of detecting vertical forces on multiple locations. Another method uses the velocity of a finger tap using an accelerometer embedded in the device. Hinckley and Song [7] presented scenarios that became possible only by combining touch and motion. Heo and Lee [5] explored possibilities of distinguishing between a gentle tap and a strong tap. The use of contact area sizes [1] was also studied. Roudaut et al. [17] showed that a touch movement caused by a finger roll can be recognized and used for additional touch gestures. Recently, using shear force [3, 4, 10] is being studied actively, because it augments touch inputs by using two dimensions, while other features, such

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2013, April 27–May 2, 2013, Paris, France.

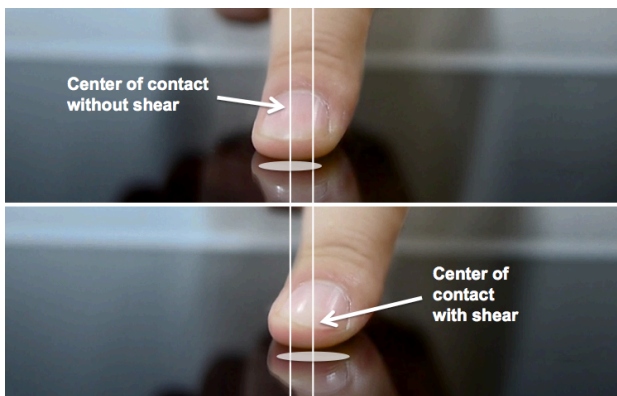
Copyright © 2013 ACM 978-1-4503-1899-0/13/04...\$15.00.

as the velocity or the strength force, augment touch inputs by using only one dimension.

**MULTI-POINT SHEAR FORCE ESTIMATION**

In order to avoid slipping, normal force must be applied to the screen at the same time as shear force. When we apply shear force and normal force on a screen at the same time, the bottom of the fingertip does not move due to friction, and the fingertip is deformed instead. The result is that the center of the contact area shifts slightly, as shown in Figure 1. This shift is roughly proportional to the applied shear force. A capacitive touch screen determines only the center of a contact area, and, therefore, it reports such a shift as a touch movement. (For the rest of this document, we will use “a touch movement” to specifically refer to a touch movement reported by a touch screen.)

Using the direction and the amplitude of a touch movement, we can estimate the direction and amount of the shear force applied by a finger. Because a touch movement for each finger is independently measured and reported by the touch screen, it is possible to estimate shear forces at multiple locations.



**Figure 1. The shift of the center of a finger contact area: (top) before and (bottom) after applying a shear force.**

**Detection of a Shear Force Event**

In order to use a touch movement to estimate shear force, we should be able to discriminate between a shear force event and a slide event. In this paper, we consider two methods for this discrimination.

The first method, which we call the *Force method*, measures normal force, using force-sensitive resistors (FSRs) under the screen. This method works because normal force is necessary when applying shear force; however, it requires additional force sensors under the touch screen. On the other hand, adding sensors under the screen is easier than adding sensors around the screen in order to sense shear forces, as in the cases of [4, 10]. In addition, some mobile phones in the market already come with FSRs under the touch screen [15].

The second method, which we call the *Area method*, uses contact area information from a touch screen to indirectly estimate the normal force. Using a contact area to infer the

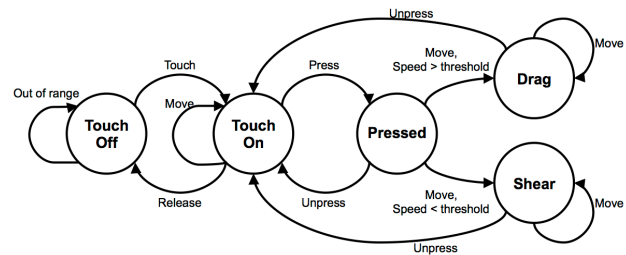
normal force [2] or to enable an alternative input [1] has already been studied. However, estimating normal force this way is not very reliable, because the size of the contact area is not only dependent on an applied force, but also on the pose and the location of the finger, as described in FatThumb [1]. Despite its inaccuracy, this method is a possible option, because detection of a shear force event does not require the estimation of the normal force to be very accurate. In addition, this method may be a more attractive option than the Force method, because this method does not require additional sensors.

Both methods have their advantages and disadvantages. In the Force method, more accurate pressure sensing is possible; therefore, discrimination of a shear force event will be more reliable. However, detecting pressed states independently at multiple locations is not easy, even with an array of force sensors under a touch screen. In the Area method, the main advantage is that it does not require any additional sensors and it can be applied to the current touch screens in the market. Another important advantage is that normal forces can be independently estimated at multiple touch locations. However, as previously mentioned, discriminating a shear force event from a slide event is relatively unreliable.

**Algorithm Details**

The shear force estimation algorithms of the two methods are almost the same. The only difference is in identifying a shear force event.

In the Force method, the algorithm performs a calibration at the moment the finger comes into contact with the screen. This is done to offset the effect of gravity, which changes as the orientation of the device changes. When the force applied by the finger on the touch screen exceeds a predefined threshold, the system transitions to the Pressed state (see Figure 2). In the Area method, a gravity calibration step is not necessary, and a contact area is used to determine when to transition to the Pressed state.



**Figure 2. State transition diagram**

We assumed that a finger could not slip in the Pressed state due to friction, and that a touch movement in this state could be used to estimate a shear force vector. In a pilot test, however, we observed that participants could make a slide operation even in the Pressed state. Because a shear force (Shear) operation and a forceful slide (Drag) operation have clearly different intentions, we needed to distinguish between these two inputs. The problem was that the touch

movement is similar in both cases. One of the data features that turned out to be useful for distinguishing these two cases was the speed of the touch movement. When a touch movement is faster than a certain threshold (approx. 200 mm/s), the movement is regarded as the start of a Drag operation, and a transition to the Drag state is triggered. Otherwise, it is regarded as the start of a Shear operation, and a transition to the Shear state is triggered. The algorithm records the touch location upon transition to the Shear state and uses it as the origin of the shear force vector. While in the Shear state, the value of the shear force vector changes as the touch location changes. Because Drag is not a common touch operation and it is not easy to slide fingers while applying pressure [4], we decided to use Drag operations for four directional gestures only. In addition, a slow Drag cannot be detected and is recognized as a Shear operation.

When two fingers are touching the screen, the state of each touch is determined independently in the Area method. In the Force method, however, two fingers share the same state, because it is difficult to separately determine the normal forces applied by the two fingers.

In a pilot test, we observed that a finger slides slowly on the touch screen during a Shear operation. This sliding invalidates the origin of the shear force vector, which was set upon transition to the Shear state. Therefore, it was necessary to update the origin continuously while a finger touch is in Shear state. In the final algorithm, the update of the origin is done by a simple low-pass filter of the shear force vector; that is, the origin moves in the direction of the vector by 1% of its magnitude every 0.03 ms.

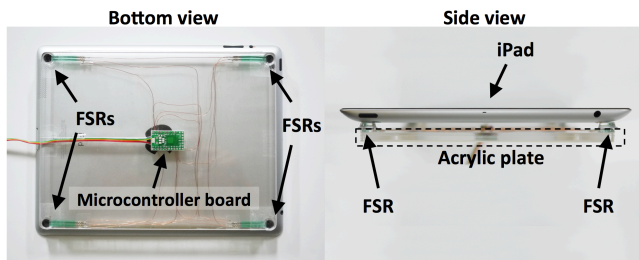


Figure 3. Bottom view and side view of the prototype

### PROTOTYPE IMPLEMENTATION

We implemented a prototype to test the feasibility of the proposed methods. The main part of the prototype is a third generation iPad, which detects and handles touch events and performs required calculations. Under the iPad is a sensing plate, which is a 7 mm thick acrylic plate with four FSRs (with a sensing range of 0.2 N to 10 N). A Teensy board with an ATMEGA32 microcontroller performs 10-bit analog-digital conversions and transfers sensor data to the iPad. Normal force is calculated by summing up the sensor values from the four FSRs. Figure 3 shows the bottom view and the side view of the prototype. The acrylic plate can be omitted when sensors can be placed directly under the screen glass.

### A DESIGN EXAMPLE: BUG HUNTER

In order to show the feasibility of the two proposed methods, we implemented a simple computer game called Bug Hunter. The goal of the game is to kill all ants, while ignoring other bugs such as snails and caterpillars. In the game, traditional touch screen operations such as Slide and Pinch are used to pan across and to zoom in/out of the game field. Instead of a weapon, shear force operations are used for attack. The shear force gestures used in the game are shown in Figure 4. Users can swing a light saber by using a Shear operation, create a valley by using a Shear Pinch operation, dig a hole to trap bugs by using a Shear Spread operation, and call for air support by using a Drag operation.

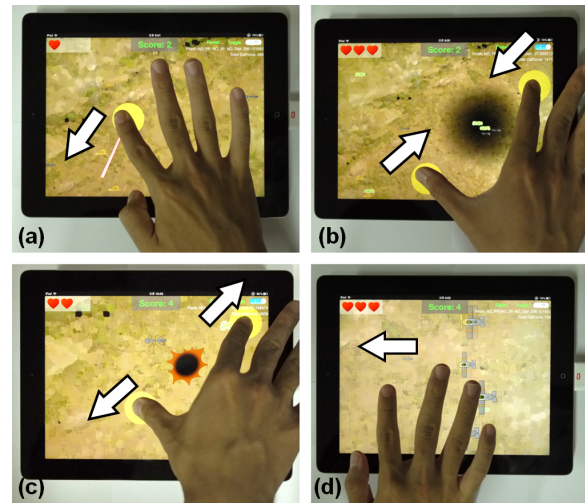


Figure 4. Shear force operations used in Bug Hunter: (a) Shear (b) Shear Pinch (c) Shear Spread (d) Drag

We recruited eight participants (three females and five males, mean age = 20.6) for an informal evaluation session. After a short demonstration, we allowed the participants to play the game once with the Force method and another time with the Area method. In the demonstration, we clearly explained the difference between the two methods. To eliminate ordering effects, four participants played with the Area method first, and the other four played with the Force method first. We asked them to speak freely when they encounter any inconvenience or have a problem. We recorded their comments and observed them while they were playing the game. After the experiment, participants were asked to answer 5-point Likert scale questions about intuitiveness, ease of learning, fatigue, ease of use, level of enjoyment, and conflicts between touch operations. They were also asked to choose a preferred method and to describe the reason for their preference.

In the results, all participants answered that there was a significant difference in usability between the two methods. All but one participant answered that they preferred the Force method, because there were less conflicts between Push and other operations. In the questionnaire, participants gave higher ratings to the Force method than to the Area method in all questions, especially in “easy to use” and

“less conflict.” Most errors were made when a user tried to perform a Shear operation with the Area method, which often misidentified Shear operations as Slide operations. These errors were more frequent with upward shear force operations, where the finger is slightly lifted and the contact area is smaller. Some participants commented that the Area method could be acceptable after a period of familiarization; however, the Force method would still be preferable.

**DISCUSSION**

Figure 5 shows the extended design space of touch screen operations that will be made possible by using multiple shear forces. The columns of the table are the different types of operations: Slide operations, Shear operations, and composite operations, which are combinations of different types of operations, including Drag operations. The rows are the numbers of contact fingers. There are two different two-finger cases: Cooperative and Independent. Cooperative indicates that the movements of two fingers are cooperative like pinch or rotate operations. Independent indicates that the movement of one finger is not related to the movement of the other.

Number of contact fingers	Two Fingers	Cooperative	Pinch / Spread Rotation	Shear Pinch / Spread Shear Rotation	Touch / Press + Pivot / Drag	Heterogeneous
		Independent			Drag Pinch / Spread Drag Rotation	Homogeneous
Two Fingers	Independent		Two Slides	Two Shears	Shear + Slide	Heterogeneous
					Two Drags	Homogeneous
One Finger			Slide	Shear [3, 4, 9]	Drag [4]	Explored
			Slide operations	Shear operations	Composite operations	Unexplored
Movement type						

**Figure 5. The extended design space of touch screen operations made possible by using multiple shear forces.**

In Figure 5, the first column has been the subject of extensive research [18]. In the second and third columns, however, only the one-finger case is getting research focus and only recently [3, 4, 10]. The remaining cases marked in gray have not yet been explored systematically, possibly due to the lack of a hardware platform or technique to support multi-point shear force sensing. More complex touch screen interactions will become possible when this unexplored space can be fully utilized.

In the Bug Hunter example, we mapped Shear operations to new application-specific interactions only in order to emphasize new possibilities. Shear operations can be used to enable more general functions, such as rate-controlled panning or zooming.

**CONCLUSION**

We designed two methods to indirectly estimate shear forces at multiple points, and we showed their feasibility by implementing a hardware prototype and a demo application that utilizes multi-point shear force operations. In the example game application, we demonstrated a few novel

interaction techniques using multi-point shear forces. We also presented a summary of new design space that will be enabled by multi-point shear force sensing methods.

**ACKNOWLEDGEMENTS**

This work was supported by the IT R&D program of MKE/KEIT [10039161].

**REFERENCES**

- Boring, S., Ledo, D., Chen, X., Marquardt, N., Tang, A., Greenberg, S. The Fat Thumb: Using the Thumb’s Contact Size for Single-Handed Mobile Interaction. *MobileHCI '12*.
- Davidson, P.L., Han, J.Y. Extending 2D object arrangement with pressure-sensitive layering cues. *UIST '08*.
- Harrison, C., Hudson, S. Using Shear as a Supplemental Two-Dimensional Input Channel for Rich Touchscreen Interaction. *CHI '12*.
- Heo, S., Lee, G. Force gestures: augmenting touch screen gestures with normal and tangential forces. *UIST '11*.
- Heo, S., Lee, G. Forcetap: extending the input vocabulary of mobile touch screens by adding tap gestures. *MobileHCI '11*.
- Herot, C., Weinzapfel, G. One-point touch input of vector information for computer displays. *SIGGRAPH '78*.
- Hinckley, K., Song, H. Sensor synaesthesia: touch in motion, and motion in touch. *CHI '11*.
- Holz, C., Baudisch, P. The generalized and perceived input point model and how to double touch accuracy by extracting fingerprints. *CHI '10*.
- Kamiyama, K., Vlack K., Mizota, T., Kajimoto, H., Kawakami N., Tachi S. Vision-based sensor for real-time measuring of surface traction fields. *Computer Graphics and Applications* 25, 1, IEEE (2005), pp.68–75.
- Lee, B., Lee, H., Lim, S., Lee, H., Han, S., Park, J. Evaluation of human tangential force input performance. *CHI '12*.
- Minsky, M. Manipulating simulated objects with real-world gestures using a force and position sensitive screen. In *Proc. 11th annual conference on Computer graphics and interactive techniques*, 1984, pp.195–203.
- Miyaki, T., Rekimoto, J. GraspZoom: zooming and scrolling control model for single-handed mobile interaction. *MobileHCI '09*.
- Ramos, G., Boulos, M., Balakrishnan, R. Pressure widgets. *CHI '04*.
- Ramos, G., Balakrishnan, R. Pressure marks. *CHI '07*.
- Research In Motion™, SurePress™ Technology <http://us.blackberry.com/specifications/communication/surepress-touch-screen.html>
- Rosenberg, I., Perlin, K. The UnMousePad: an interpolating multi-touch force-sensing input pad. *SIGGRAPH '09*.
- Roudaut, A., Lecolinet, E., Guiard, Y. MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. *CHI '09*.
- Westerman, W., Hand tracking, finger identification, and chordic manipulation on a multi-touch surface. University of Delaware, 1999.