Touch Shadow Interaction and Continuous Directional User Interface for Smartphone

Sanjay Ghosh[1], Joy Bose[1], Rajkumar Darbar[2] and Punyashlok Dash[2]

[1] Samsung R&D Institute Bangalore, Karnataka, India
{sanjay.ghosh, joy.bose}@samsung.com,
[2] Indian Institute of Technology, Kharagpur
{rajdarbar.r, punyashlok.com}@gmail.com

Abstract: We propose a new touch screen interaction named as Touch Shadow, which is based on the contact area and the resultant shape of the fingertip when it touches the touch surface. The results of our feasibility evaluation established three degrees of freedom for the Touch Shadow interaction based on three fundamental parameters, the touch surface area, displacement of the touch centroid and direction of the press. For this interaction, we designed several user interface options that can support continuous interaction. Based on the review feedback from 6 UI designers, we finally selected three of them, the Bubble Shadow, Rectangular Shadow and Beam Shadow for user evaluation. A proof of concept prototype was developed for Android smartphones and was tested with 10 users for maps application scenario. The participants rated the experience of using Touch Shadow with continuous direction UI on the map to be high.

1   Introduction

Existing touch gestures are primarily two dimensional in nature, as they are limited by the position and movement of fingers or stylus across locations on the touch screen. Moreover, most of the existing touch interactions are committable or discrete by nature, and not exploratory or continuous. This means that, when a user performs any existing touch action, say, tap or long press on a button, the mapped event is bound to happen. Thus, existing touch interactions assume that the user is certain about which action to perform for which task. However, several applications with real life situations require the user to explore various possible options and then take a decision. Examples are: searching for a hotel using a map or choosing a particular family picture from the phone gallery. For such instances, the interaction as well as the user interface needs to be designed to support the cognitive mental model of ambiguity.

To support such a requirement, the interaction must be inherently continuous in nature, which would enable the user to give continuous input to the device using simple interactions, similar to that which is possible using a joystick. Designing such interactions would need the touch hardware to respond to more degrees of freedom (DOF) which a user can control using touch actions. The existing touch gestures only allow

limited degrees of freedom, limited by the touch coordinates on a touch screen. Although pinch, pan, and scroll among the current touch gestures are of continuous nature, these are also tightly coupled with the distance traveled by the user's fingers on the touch surface. The user is still required to move his finger on the 2D touch surface in order to perform these gestures. Position 3D touch or Pressure Touch [1], which is patented by Apple, is beyond just 2D touch and allows one additional degree of freedom along the Z-axis with three intermediate interaction levels, but it may still be considered as discrete interaction rather than a continuous one. Moreover, techniques such as Apple's 3D touch require specialized embedded hardware to detect the finger pressure during the touch. Our research objective was to develop a new touch interaction with above mentioned capabilities without employing additional hardware.

In this paper, we propose and evaluate a new touch interaction method for smartphones which we call as Touch Shadow, wherein the user can perform continuous input without lifting and moving his finger. Shadow here is a metaphor used to represent the variable surface area of the finger tip that touches the touch surface. When different levels of pressure are applied to the finger on a touch surface, the resultant surface area and shape of the finger contact slightly changes. Also, the bulge of the fingertip depends on the direction of the applied pressure. The advantage of the touch shadow interactions over the existing touch interaction includes the ability to continuously interact with UI elements in the display. In this study, our objective was to evaluate how clearly these slight changes could be detected and could be mapped to interactions. To that end, we developed a proof of concept prototype on an Android smartphone with a capacitive touch screen to demonstrate the concept. The detection of this interaction is based on the measurement of the total capacitance value from the surface area of the finger touch and hence no additional hardware was employed. We evaluate the feasibility of detection along three possible degrees of freedom, i.e. change in the touch area, change in direction and change in the touch centroid, on our prototype with 10 users. We also identified and implemented 3 UI elements using the interaction on a map application in Android, and evaluated with 10 users for ease of use. Thus, this study established the feasibility of the proposed interaction for smartphone applications.

2   Related work

There are several research prototypes which detect the finger pressure [2] or shear force [3] [4] to perform continuous interactions. Most of these employ relevant sensors below the touch screen to augment a device such as a smartphone. There are a few approaches which detect an equivalent of finger pressure in some form without actually using any pressure sensors. WatchMI [5] is one such prototype which detects different levels of continuous touch pressure through finger movements data recorded in the built-in IMU sensor in the Smartwatch. Contrastingly, there are also few prototypes which use various actions of the finger on 2D touch surface as a close proxy to the pressure, an approach used in our Touch Shadow interaction method. Wang et. al. [6] performed an empirical evaluation of the three key properties of the finger touch,

i.e. contact area, contact shape and contact orientation and proposed to use these for defining additional degrees of freedom for touch gestures on a 2D touch surface. This forms one of the inspirations for our research. Xiao et. al. [7] proposed to use the finger movement along the pitch and yaw relative to a touch surface to trigger actions on a smartphone and a smartwatch. Roudaut et al. [8] proposed MicroRolls gestures using the thumb, wherein the stationary thumb acquires six small magnitude rolling positions which can be detected on the touch surface. TapSense [9] prototype detects the portion of the user's finger including the tip, pad, nail, and knuckle that is used to make the touch interaction. Angle pose [10] estimated a finger's 3D pose and angle based on the model using cumulative capacitive values and used them for improved accuracy of touch. Work by Wang et al. [11] could detect finger orientation based on the directionality and the shape of the touch surface. Closely relevant to our work is that of Boring et al. [12] which uses the change in thumb's contact size along various threshold levels for mapping different user interaction. Here the elliptical size of thumb contact portion was considered virtually equivalent to the finger pressure.

Notably, most of the above mentioned works focused only on one of the additional aspects of finger touch other than just the location and movement along the x-y coordinate of the 2D touch surface. This additional aspect may either be the touch angle or pose of the finger or size of the touch surface or the slight movement of the finger. However, our Touch Shadow gesture is based on 3 aspects: change in touch area, change in direction and change in the touch centroid. Also unlike ours, most of the prior works do not primarily define the continuous aspect of the interaction.

3   Detection Parameters for Touch Shadow interaction

We here explain the fundamental detection parameters for detecting Touch Shadow.

3.1   Change in Area / Shape

When the user rolls or presses the finger, the fingertip bulges, resulting in a change in the area of the finger in contact with the screen. This change can be in the positive or negative direction, which represents one of the degrees of freedom. By measuring the change in this area or shape we mapped it to different interaction categories.

3.2   Displacement of the Centroid

The centroid of the ellipse formed by the touch surface of the finger with the touchscreen gets displaced towards the direction of the tangential force, resulting in an additional degree of freedom. If (x_1, y_1) are the initial touch coordinates and (x_2, y_2) the final touch coordinates, the displacement is given by $\sqrt{((x_2-x_1)^2 + (y_2-y_1)^2)}$.

3.3   Directionality

The direction of movement of the finger can also be identified and serves as an additional degree of freedom. The x and y touch coordinates of the touch point, along with

the touch major and touch minor of the ellipse created by the finger can be used to calculate the Displacement Angle of centroid Θ, given by $\tan^{-1} (y_2-y_1)/(x_2-x_1)$.

4   Touch Shadow detection feasibility

We performed a detection feasibility experiment to study the degrees of freedom that are possible for our Touch Shadow interaction. We used three Android devices with different levels of screen resolutions: Samsung S6, Samsung J7, and Motorola G3. The prototype, as shown in Fig. 1(a), had a circular touch sensitive area for users to move their finger with seven different tasks as mentioned in table 1.
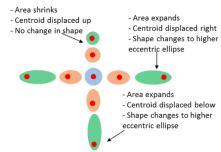

Fig. 1. (a) Prototype used for the feasibility evaluation.

Fig. 1. (b) Pattern of finger touch shapes, area and centroid displacements during evaluation.

Test action	Touch Major	Touch Minor	X co-ordinate	Y co-ordinate	Area
Normal Touch (Default Light Press)	2.93	2.93	737.23	1993.13	0.02
Radial Shadow (Medium Press)	4.83	3.42	738.28	2021.88	0.03
Radial Shadow (Strong Press)	5.86	5.86	717.19	2031.88	0.05
Left Shadow (Light Press)	2.93	2.93	657.07	1991.88	0.02
Max Left Shadow (Light Press)	4.88	3.42	605.04	2008.75	0.03
Right Shadow (Light Press)	3.42	3.42	766.05	2005	0.03
Max Right Shadow (Light Press)	3.42	1.95	919.34	2043.21	0.02
Up Shadow (Light Press)	1.95	1.95	717.19	1938.75	0.02
Max Up Shadow (Light Press)	1.95	1.95	720	1853.75	0.02
Bottom Shadow (Light Press)	3.42	3.42	730.20	2051	0.03
Max Bottom Shadow (Light Press)	5.37	3.91	734.06	2113.13	0.04

Table 1. The captured values for the touch ellipse for the feasibility experiment, averaged over 10 users, using a Samsung Galaxy S6.

We tested it with 10 users, all heavy users of smartphone applications in the age group of 24-35 years (Mean Age = 28.3 years; SD=2.8; 4 Female; 6 Male), and then took the average of the values to normalize for the size of the fingers and other parameters readings. We actually did not observe high within-subject variations. In fact, the readings for touch major, touch minor, and area for all the 10 users differed not

more than 10%. This also assured sufficient accuracy in terms of detecting the correct interaction. The results for Samsung S6 are tabulated in Table 1. We obtained similar results using the other devices as well. We recorded the values of the x and y coordinates of the touch point and the minor and major axes of the ellipse around the touch point.The key aspect of the results is as follows: the touch surface area expanded in a pattern similar to as shown in fig 1 (b) when the finger was rolled from the initial position to the left, right, or down position. It shrunk slightly when the finger was rolled up. From this, we conclude that there are 3 DOF identifiable for the Touch Shadow interaction: the direction (left, right, up, down) of the finger movement, the (x, y) touch coordinates and the surface area of touch (size).

5 Continuous Directional UI Elements for Touch Shadow

Based on the detection feasibility of the Touch Shadow interaction, we conceptualized 10 possible user interface options for the same, each of which was continuous in nature, i.e. their shape, size and the orientation should vary according to the change in the finger touch area. The 10 UI options each corresponded to a primary standard shape, such as rectangle, circle, ellipse, etc. We asked 6 UI designers to review all 10 UI options with respect to a set of 10 different hypothetical application use cases which required continuous interaction, the use cases being provided as stimulus in form of mobile wireframe illustrations. Few of these use cases were: exploring and selecting a music track, searching for a particular photo in a gallery, selecting a video, etc. We performed a subjective analysis of the feedback and reasoning received and selected the three most preferred continuous directional UI options, i.e. Bubble, Rectangular and Beam Shadow for further evaluation. Most of the participants found these three UI options to be appropriate for the majority of the application use cases.
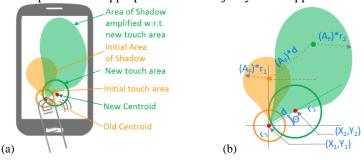

Fig. 2. Illustration of (a) a bubble shaped UI element created by the user in the direction of change of touch area, and (b) the area and direction of the UI element.

5.1 Bubble Shadow

A bubble shaped UI element is created in the direction of change of the touch area. Fig.2 shows the calculation of the touch area and direction of expansion for the gesture. The touch region is assumed to be a circle for the sake of easier understanding,

whose centre is at a distance d and angle Θ from the initial coordinates of the touch point. Here (x_1, y_1) and (x_2, y_2) are the initial and final coordinates of the centroid.

5.2   Rectangular Shadow

Here the UI element is in the shape of a rectangle. Here Θ assumes one of the following four values: 0, 90, 180, or 270 degrees, depending on the predominant cardinal direction of the finger displacement (up, bottom, left, right). As in the bubble UI, the rectangle amplified the displacement of the touch point by an amplification factor A_{F1}. So the movement of the center of the rectangle = $A_{F1} * \Delta d$.

5.3   Beam Shadow

In this case, the UI element is in the shape of a beam (similar to the light of a torch), where the angle Θ made by the beam is a value between 0 and 90 degrees in the predominant cardinal direction of the finger displacement, as before. The length of the arc r is proportional to the displacement of the centroid Δd, amplified by an amplification factor A_F. The arc angle is proportional to ΔA, the change of touch area.

Table 2 describes the 3 UI elements w.r.t. the 3 DOF parameters.

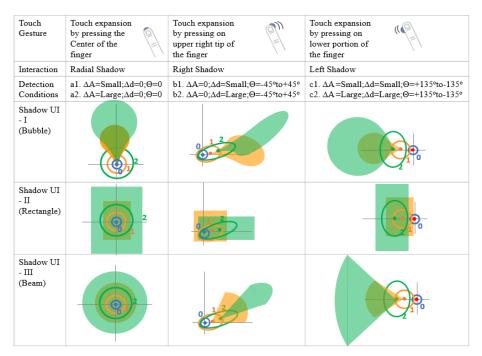| Touch Gesture	Touch expansion by pressing the Center of the finger	Touch expansion by pressing on upper right tip of the finger	Touch expansion by pressing on lower portion of the finger
Interaction	Radial Shadow	Right Shadow	Left Shadow
Detection Conditions	a1. ΔA=Small;Δd=0;Θ=0 a2. ΔA=Large;Δd=0;Θ=0	b1. ΔA=0;Δd=Small;Θ=-45°to+45° b2. ΔA=0;Δd=Large;Θ=-45°to+45°	c1. ΔA=Small;Δd=Small;Θ=+135°to-135° c2. ΔA=Large;Δd=Large;Θ=+135°to-135°
Shadow UI - I (Bubble)			
Shadow UI - II (Rectangle)			
Shadow UI - III (Beam)			

Table 2. Illustration of the modes of using the Touch Shadow gesture

In our prototype, we were able to detect at least 64 (4^3 for 3 DOF parameter with 4 levels for each DOF parameter) different instances of detection conditions, Only 6 of those 64 detection conditions are described in the three columns of table 2. At least

one of the 3 DOF parameters, i.e. ΔA, Δd and Θ should vary beyond the threshold limit for a valid instance of a detection condition. The three UI options, Bubble, Rectangular, and Beam are illustrated for these 6 different instances of detection conditions. For detection condition a1 and a2, the only ΔA varied continuously; while Δd and Θ were constant, a condition achieved by pressing the center of finger touch area. Similarly, other corresponding detection instances can be inferred from table 2.

6   Prototype implementation details

We implemented a prototype of the Touch Shadow UI, using a Samsung Galaxy S6 smartphone running Android version 7. For simplicity, we assumed the movement of only one of the following: up, bottom, left, right, or radial. For the simplicity of explaining the interaction detection logic and UI rendering logic, we provide the pseudo code only for the Bubble UI, with the bubble shape approximated by an ellipse.

IDENTIFY-DIRECTION-FINGER-MOVEMENT () // Identify direction

```
1   Tx , Ty = Threshold for min. movement of touch point centroid in the X axis; //Input
2   TA = Threshold for min. change in area of the touch ellipse; // Input
3   x1, y1, A1 = Initial touch coordinates, touch area; //using getX, getY, getArea functions
4   touchMajor1, touchMinor1 = Initial major and minor axes of touch ellipse;
5   x2, y2 = New touch coordinates; A2 = New touch area;
6   dx = x2 - x1, dy = y2 - y1; //movement of the centroid of touch point
7   dA = A2 - A1; //change in area of the touch ellipse
8   if ( abs(dx) < Tx and abs(dy) > Ty ) //movement in up-down direction
9       if (dy > 0) then direction = DIR_UP; else direction = DIR_BOTTOM;
10  if ( abs(dx) > Tx and abs(dy) < Ty ) //movement in left-right direction
11      if (dx > 0) then direction = DIR_RIGHT; else direction = DIR_LEFT;
12  if ( abs(dx) < Tx and abs(dy) < Ty ) //no movement, press down at same point
13      if (dA > TA) then direction = DIR_RADIAL;
14  return direction;
```

PLOT-UI-ELEMENT () // Plot the UI element based on identified direction

```
1   T1 = Polling interval in millseconds; // Input
2   A1, A2 = Amplification factors for UI, based on the model of device used; //input
3   direction = Identified direction of finger movement; //input
4   for(;;)
5       CLEAR_CANVAS();
6       touchMajor1, touchMinor1, x1, y1, A1 = GET-TOUCH-PARAMETERS();
7       r1 = touchMajor1 * A1;    r2 = touchMinor1 * A1;
8       if(direction ==DIR_LEFT)
9           c = abs(x2-x1)*A2; PLOT-UI-ELLIPSE(x1-c, y1, r1, r2);
10      if(direction ==DIR_RIGHT)
11          c = abs(x2-x1)*A2; PLOT-UI-ELLIPSE(x1+c, y1, r1, r2);
12      if(direction ==DIR_UP)
13          c = abs(y2-y1)*A2;  PLOT-UI-ELLIPSE(x1, y1+c, r1, r2);
14      if(direction ==DIR_BOTTOM)
15          c = abs(y2-y1)*A2;   PLOT-UI-ELLIPSE(x1, y1-c, r1, r2);
16      if(direction ==DIR_RADIAL)  PLOT-UI-ELLIPSE(x1, y1, r1, r2);
17      sleep(T1); // Poll every T1 millisec
18  return;
```

7   User Interface Evaluation

We conducted an evaluation with 10 users, the same set of people involved earlier in initial feasibility evaluation. They used the prototype with the 3 UI elements, bubble, rectangular and beam, overlaid on a map for a task to explore hotels around a fixed location (convention center). The users were asked to place their index finger on the location and explore the hotels in the region under one of the 3 UI elements using the continuous directional Touch Shadow interaction. Based on the changing UI shape, the interface displayed information such as hotel distance, cost, rating and discounts as in Fig. 3. For the prototype, we simulated only dummy values in the map interface. Post user trials, we collected their subjective feedback in terms of usefulness of the Touch Shadow interaction and ease of use of the 3 UI elements.

Fig. 3 Wireframe illustration of the Bubble Shadow UI on a maps application.

Majority of the users found the rectangular UI to be the easiest to control with Touch Shadow interaction, since the rectangular UI overlay covered appropriate portions of the rectangular smartphone display. A user suggested adapting default aspect ratios of the rectangular UI based on the smartphone used. Many users found the Beam UI to be most relevant for scenarios where a pivot point of interest is important, as in the test use case of finding hotels around a location. Users found the Rectangular UI relevant when a portion of display interface or an area was important. For some users, finger occlusion was the primary concern while using the Beam UI element, so they preferred the bubble UI considering it to be a superior variant of the Beam UI. Overall, almost all the users rated their experience of the Touch Shadow interaction to be high and found this to be highly useful for application use cases.

8   Conclusion

We proposed and evaluated a new interaction method called Touch Shadow, with the ability to continuously interact with UI elements enabling the user to continuously explore options and then choose. In future, we plan to implement the Touch Shadow interaction and corresponding UI elements for other key smartphone applications.

References

1. Apple, Inc. Multi-touch gesture dictionary, US Patent 20070177803 (2008).
2. Clarkson, E.C., Patel, S.N., Pierce, J.S., and Abowd, G.D.: Exploring continuous pressure input for mobile phones. Georgia Institute of Technology, 2006.
3. Heo, S., and Lee, G.: Force gestures: augmenting touch screen gestures with normal and tangential forces. In: 24th annual ACM symposium on User interface software and technology, pp. 621-626, (2011).
4. Harrison, C., and Hudson, S.: Using shear as a supplemental two-dimensional input channel for rich touchscreen interaction, In: SIGCHI Conference on Human Factors in Computing Systems, pp. 3149-3152, (2012).
5. Yeo, H.S., Lee, J., Bianchi, A. and Quigley, A.: WatchMI: pressure touch, twist and pan gesture input on unmodified smartwatches. In: 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 394-399, 2016.
6. Wang, F., and Ren, X., Empirical evaluation for finger input properties in multi-touch interaction. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 1063-1072, (2009).
7.   Xiao, R., Schwarz, J., and Harrison, C.: Estimating 3D Finger Angle on Commodity Touchscreens. In: International Conference on Interactive Tabletops & Surfaces, pp. 47-50, (2015).
8. Roudaut, A., Lecolinet, E., and Guiard, Y.: MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 927-936, (2009).
9. Harrison, C., Schwarz, J., and Hudson, S.E.: TapSense: enhancing finger interaction on touch surfaces. In: 24th annual ACM symposium on User interface software and technology, pp. 627-636, (2011).
10. Rogers, S., Williamson, J., Stewart, C., and Murray-Smith, R.: AnglePose: robust, precise capacitive touch tracking via 3d orientation estimation. In: SIGCHI Conference on Human Factors in Computing Systems, pp. 2575-2584, (2011).
11. Wang, F., Cao, X., Ren, X., and Irani, P.: Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In: 22nd annual ACM symposium on User interface software and technology, pp. 23-32, (2009).
12. Boring, S., Ledo, D., Chen, X.A., Marquardt, N., Tang, A., and Greenberg, S.: The fat thumb: using the thumb's contact size for single-handed mobile interaction. In: 14th International Conference on Human-computer interaction with mobile devices and services, pp. 39-48, (2012).