



**HAL**  
open science

# LayerStroke: a Layer based Selector for Small Target Acquisition

Siju Wu, Amine Chellali, Samir Otmane, Guillaume Moreau

► **To cite this version:**

Siju Wu, Amine Chellali, Samir Otmane, Guillaume Moreau. LayerStroke: a Layer based Selector for Small Target Acquisition. 27ème conférence francophone sur l'Interaction Homme-Machine., Oct 2015, Toulouse, France. pp.w17, 10.1145/2820619.2825019 . hal-01219114

**HAL Id: hal-01219114**

**<https://hal.science/hal-01219114>**

Submitted on 23 Oct 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# LayerStroke: a Layer based Selector for Small Target Acquisition

**Siju Wu**

IBISC, Evry University  
Evry, France  
siju.wu@ibisc.univ-evry.fr

**Amine Chellali**

IBISC, Evry University  
Evry, France  
amine.chellali@ibisc.univ-evry.fr

**Samir Otmane**

IBISC, Evry University  
Evry, France  
samir.otmane@ibisc.univ-evry.fr

**Guillaume Moreau**

CERMA, Ecole Centrale de  
Nantes  
Nantes, France  
guillaume.moreau@ec-nantes.fr

**Résumé**

In this paper, we introduce LayerStroke, a new touch-based selection technique for small targets located in a dense cluster. This technique aims to reduce the cluster density by dividing the targets space into different layers. To select a target, users can first select a layer by drawing a stroke on the screen. After that, targets inside the selected layer can be picked up by tapping the corresponding tiles. Our technique has two advantages. First, targets can be selected easily by using a simple combination of drag and tap gestures. Second, unlike some other acquisition techniques, no magnified view is necessary and the initial view can be used for selection.

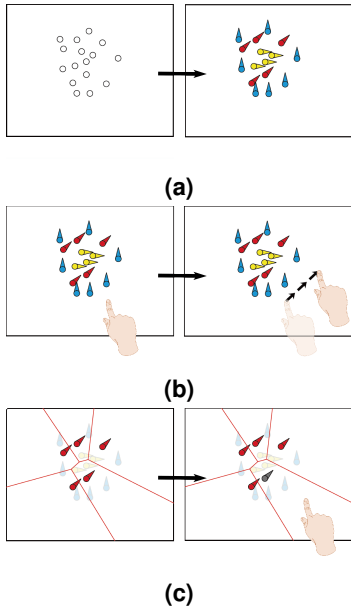
**Mots-clés**

target acquisition ; touch input ; dense target selection.

**Introduction**

Since the emergence of the Apple iPhone, touch input became quickly the main interaction method on smartphones and tablet computers. However, although a decade has passed since the first use of touchscreens on mobile devices, touch input still suffers from the “fat finger” problem. This issue refers to the tendency to make errors when elements on the display are too small to be selected using a fingertip.

© ACM, 2015. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Actes de la 27ème conférence francophone sur l'Interaction Homme-Machine, 2015.  
<http://dx.doi.org/10.1145/2820619.2825019>



**FIGURE 1** – The selection procedures of LayerStroke. (a) Divide targets into several layers. (b) Draw a stroke to select a layer. (c) Tap the finger in the tile of a target to select it.

This decreases the selection accuracy and causes the occlusion problem. Hence, it is still a challenge to accurately select a small object displayed on the screen. To overcome the occlusion problem, several techniques such as Shift [5], MagStick [4], Bubble Cursor [3], Area Cursor [6] or Starburst [2] have been proposed. However, these techniques have some limitations such as long cursor adjusting time, influence of high cluster density or visual distraction.

In addition to the occlusion problem, another challenge of selection on touchscreens is to reduce the finger motion precision requirement. Compared with controlling the cursor by a mouse, it is more difficult to tap the finger very accurately at one position on the screen. As a result, the user may fail to select a small target by a single tap. Several approaches such as Escape [7] and LinearDragger [1] have been proposed to reduce the influence of the target size. Similarly, these techniques also suffer from some issues such as the high cluster density and the limited size of the screen.

In this paper, we introduce a dual-step target selection technique named LayerStroke. This technique aims to improve the selection task by dividing the targets space in the display into several layers to reduce the cluster density. Precise selection of small targets can be made without magnifying the initial view.

## Design and implementation

We present in this section, the design and implementation of LayerStroke.

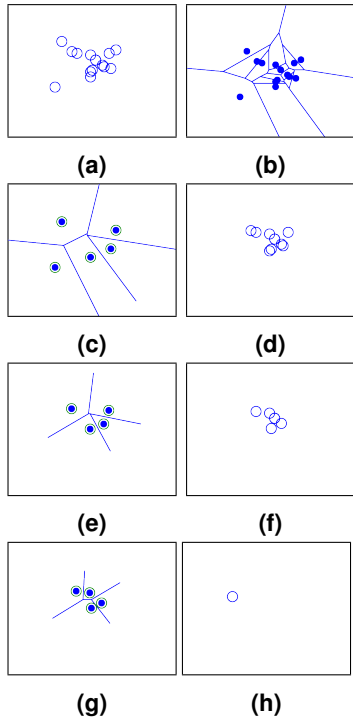
### *Design of selection techniques*

Our technique can be considered as a mutation of Bubble Cursor. The performance of Bubble Cursor is

influenced by the cluster distribution and selection becomes more difficult when the density increases. In our design, the objective was to reduce the influence of high distribution density. In some GIS (Geographic Information System) and design software such as Adobe Photoshop, layers are used to organize different contents. Specific tasks, such as data manipulation or drawing can only be made in the selected layer. Hence, we reused the same concept i.e. dividing the targets space into several layers, in order to simplify target selection.

When some selectable targets are displayed on the screen, the system first separates them into different layers (Figure 1(a)) using our algorithm. Targets inside the same layer are displayed in the same color. For example, in Figure 1, targets of three different layers are displayed in blue, red and yellow respectively. In addition, inspired by Escape [7], a directional layer indicator is also displayed upon each target. Targets having the same indicator are in the same layer. In the example of Figure 1, the direction of blue targets is Up and that of the yellow targets is Right.

After recognizing the direction of the desired layer, LayerStroke enables users to draw a stroke using one finger on the screen in the same direction to select the layer (Figure 1 (b)). The drag gesture can be started from an arbitrary position on the screen. After the layer is selected, all the targets located in the other layers become semi-transparent (Figure 1(c)). Furthermore, targets of the selected layer have the highest display order so that none of them are occluded by semi-transparent targets. The display order can be changed by modifying the z-buffer value of each target. Finally, tessellation tiles of this layer are generated and



**FIGURE 2** – Generation of layers. (a)Initial targets cluster. (b) Voronoi Tessellation of initial distribution. (c) The first layer. (d) The remaining targets after generating the first layer. (e) The second layer. (f) The remaining targets after generating the second layer. (g) The third layer. (h) The fourth layer.

visualized. After selecting a layer, the user can highlight a target by tapping one finger in the tile encircling it. Finally, the finger can be released directly to select the target or can be dragged to switch the target in focus.

To cancel the selection of a layer, a two-finger pinch enlarging gesture can be made on the screen. After that, tessellation tiles of the selected layer disappear and all the semi-transparent points turn back to opaque.

LayerStroke has several advantages :

- **Combination of two simple gestures** - They require only one drag gestures to select a layer and one tap gesture to select a target.
- **No spatial limit for layer selection** - The drag gesture can be started from an arbitrary point on the screen.
- **Reduced distribution density** - By separating targets into different layers, the distribution density is highly decreased. Moreover, to reduce the visual distraction of unwanted layers, their targets become semi-transparent and are displayed below those of the selected layer.
- **No fingertip occlusion** - Since tessellation tiles for each target are large enough, finger tapping can be made in an empty area of the tile to avoid occlusion.

#### *Grouping algorithm*

To generate layers, our grouping algorithm uses the following steps :

**1. Make Voronoi Tessellation for all the targets.** As shown in Figure 2 (b), given a dense targets cluster, the coordinates of targets are first normalized and then

a Voronoi diagram is generated to make screen partitioning for all the displayed targets.

#### **2. Grouping targets according to their tiles size.**

After making Voronoi tessellation, we calculate the size of each tile. All the targets whose tile is large enough for direct tap are added to group 1 and all the remaining points are put in group 2. After several tests, we decided to set the threshold value for the tile area to 0.005.

**3. Check the tile width.** Besides the tile area, we also have to check the width of the tile. If the tile is too narrow, it is still difficult to tap it even if its area is large enough. To check the width of the tile, we note the coordinates of all its corners and use PCA (Principal Component Analysis) to calculate the eigenvectors and eigenvalues. If one of the two largest eigenvalues is smaller than a threshold value, it means that the tile is too narrow along the direction of the eigenvector corresponding to the small eigenvalue. In this case, we remove the target of this tile from group 1 and put it in group 2. The threshold value for the eigenvalue was set to 0.05 after different pilot tests.

**4. Generate a new layer.** After extracting targets whose tiles are too narrow to tap from group 1, we generate the first layer and put all group 1 targets inside it (Figure 2 (c)).

**5. Repeat the same procedure for the remaining targets.** We repeat procedures from step 1 to step 4 for the remaining targets (Figure 2 (d)) until all the targets are added to a new generated layer. Figure 2 (e, f, g, h) show the generation of other layers for the remaining points. Through an informal test, we found that the target direction has no significant influence on

the selection efficiency. Therefore the target direction of each layer is selected arbitrarily.

### **Design choices**

To investigate some design choices of our selection technique, we have conducted two preliminary studies.

#### *Display of tessellation tiles*

After a layer is selected, visualization of tiles can help the users find out more easily the effective width of targets. However, displaying additional content can also cause some visual distraction. Thus we conducted a preliminary study to determine the effect of displaying tessellation tiles on selection time and errors rate.

#### *Apparatus and participants*

The experiment was conducted on an Apple iPhone 5s, with a 4.065" touchscreen. The screen is of  $58.6 \times 123.8$  mm and has a resolution of  $640 \times 1136$  pixels. Content was displayed in the Portrait mode. The experimental environment was developed using Unity3D with C#. Eight subjects participated in this study.

#### *Design and procedure*

In this experiment, the participants were asked to use LayerStroke to select a goal target from a cluster of 12 circular targets of 12 pixels in diameter distributed randomly within a circular region of 100 pixels in diameter. Targets did not overlay each other. Twenty distracting targets of the same size were generated randomly in the remaining empty space of the screen. All the targets were displayed in a white color except the goal target, which was displayed in blue. All the targets have already been added to a layer and the layer indicator was displayed upon each target. Each participant was asked to perform two groups of

selection trials. In the first group, tessellation tiles were displayed after a layer is selected while in the second group, tessellation tiles only appeared when a finger is tapped on the screen. For each group, 30 selection trials had to be completed.

The study was a within subjects design. The within subjects factor was group with two levels (group 1 and group 2). To counterbalance the order of selection tasks, four participants started with the first group while the other four participants started with the second group. A total of  $8 \times 2 \times 30 = 480$  selection trials were performed in this experiment, with each participant completing 60 trials. Both selection time and errors rate were recorded.

#### *Results and design implications*

The mean selection time was 1936 ms when tessellation tiles were displayed all the time and 3022 ms when tessellation tiles were displayed only when the finger taps the screen. The paired-samples t-test showed that a significant difference exists between the two groups regarding the selection time ( $t(7) = 18.66, p < 0.001$ ). The errors rate was 0.8% and 5.8% respectively for group 1 and group 2. No significant difference was found between them.

Results indicate that displaying tessellation tiles improves the selection time and does not influence the errors rate. Participants were more decisive and almost no dragging motion was necessary. However, when tiles were displayed only after tapping a finger on the screen, users hesitated for a moment before deciding for the tapping position. When the tapping position was not located in the tile of the goal target, participants had to adjust the contact position on the screen to switch the highlighted target. In general, the decrease

in selection speed is mainly caused by hesitation before tapping and adjustment of finger position after tapping. All the participants preferred the configuration of group 1 and they all commented that displaying tessellation tiles was helpful. This led to the design decision to display tessellation tiles once a layer is selected and this setting is kept for the next experiment.

#### *Layer colors*

In our first prototypes, targets were displayed in the same color. In this case, before selecting a layer, the desired target could be occluded by other targets so that the direction of the layer indicator is hard to recognize. To give users more cues for recognizing the peak direction, we decided to display targets using various colors. When the indicator of the desired target is occluded, users can still find out the peak direction by observing targets of the same color as long as the color of the desired target can be recognized. Thus, a second preliminary study has been conducted to examine whether displaying various colors for different layers can help reducing the selection time and errors rate.

#### *Apparatus, participants and measurements*

The same apparatus as those in the preliminary study 1 were used. Eight subjects participated in this study.

#### *Procedure and design*

Similar to preliminary study 1, participants were asked to select a goal target from a dense targets cluster. However, some modifications have been made in this experiment. In fact, the goal target was generated in the center of the cluster for all the selection trials and targets could overlap each other. Target size was the same as in preliminary study 1. Sixteen targets were generated in the circular region in this experiment to

make the cluster denser. The study was a  $2 \times 2$  within subjects design with two within subjects' factors : Display order and Display color. Display order had two levels : Top and Bottom. Top means that the goal target is displayed on the top of the cluster so that there is no occlusion on the goal target. Bottom means that the goal target is displayed in the bottom of the cluster so that other targets can occlude it. There were also two levels for displaying color : MonoColor and MultiColor. In MonoColor condition, all the targets other than the goal target were displayed in white color while the goal target was in blue. In MultiColor condition, targets inside the same layer had the same color and their color was different from targets in other layers. To indicate the goal target, a black contour was displayed around it. Participants were asked to use LayerStroke to accomplish four groups of selection trials. The configuration of Display order and Display color of each group was different. For each group, 30 selection trials should be performed. In total there were  $8 \times 2 \times 2 \times 30 = 960$  selection trials. Both selection time and errors rate were recorded.

#### *Results and design implications*

A 2-way repeated measure ANOVA was conducted and a significant main effect was found for Display order ( $F_{1,7} = 12.69, p = 0.009$ ), and for Display color ( $F_{1,7} = 5.91, p = 0.045$ ). A significant interaction effect was also found for DisplayOrder $\times$ DisplayColor ( $F_{1,7} = 5.76, p = 0.047$ ). The mean selection times were 2222 ms and 2794 ms for MultiColor and MonoColor respectively. No significant effects were found for errors rate.

It is reasonable to find out that occlusion of the goal target slowed down the selection time. It took more

time for participants to recognize the direction of the layer indicator. The results also show that when occlusion occurs, displaying various colors can help improving the selection efficiency so our hypothesis is validated. All the participants thought that MultiColor was more helpful and they could find the direction of the goal target by observing other targets of the same color. On the other hand, they could only guess a direction when all the targets were displayed in white color. In this case, there was a lack of information to infer the indicator direction. This led to the design decision to use MultiColor for our technique.

### **Conclusion and future work**

In this paper, LayerStroke, a touch-based target acquisition technique was presented. By using our proposed algorithm, this technique can divide targets on the screen into several overlapping layers. Users can select a layer and make a tap gesture to select a target inside it. Our algorithm ensures that tessellation tiles are large enough for simple tapping without fingertip occlusion. The advantage of our technique is that accurate selection of small targets can be done without controlling very precisely the finger movement. It is also less influenced by the small screen size of the smartphone. In the future, we will compare our technique with some state-of-the-art techniques such as TapTap, Magstick and LinearDragger. Our objective is to determine whether our technique can help to further reduce the selection time and improve the selection precision. In addition, because in some applications such as Google Maps the background is complex and the color of the target has a meaning, changing its initial color to the color of its

corresponding layer may cause distraction problem. Thus we also want to examine the influence of the configuration of MultiColor in a more realistic scenario.

### **Références**

- [1] Au, O. K.-C., Su, X., and Lau, R. W. Lineardragger : a linear selector for target acquisition on touch screens. Dans *Proc. CHI '14*, ACM (2014), 2607–2616.
- [2] Baudisch, P., Zotov, A., Cutrell, E., and Hinckley, K. Starburst : a target expansion algorithm for non-uniform target distributions. Dans *Proc. AVI '08*, ACM (2008), 129–137.
- [3] Grossman, T., and Balakrishnan, R. The bubble cursor : enhancing target acquisition by dynamic resizing of the cursor's activation area. Dans *Proc. CHI '05*, ACM (2005), 281–290.
- [4] Roudaut, A., Huot, S., and Lecolinet, E. Taptap and magstick : improving one-handed target acquisition on small touch-screens. Dans *Proceedings of the working conference on Advanced visual interfaces*, ACM (2008), 146–153.
- [5] Vogel, D., and Baudisch, P. Shift : a technique for operating pen-based interfaces using touch. Dans *Proc. CHI '07*, ACM (2007), 657–666.
- [6] Worden, A., Walker, N., Bharat, K., and Hudson, S. Making computers easier for older adults to use : area cursors and sticky icons. Dans *Proc. CHI '97*, ACM (1997), 266–271.
- [7] Yatani, K., Partridge, K., Bern, M., and Newman, M. W. Escape : a target selection technique using visually-cued gestures. Dans *Proc. CHI '08*, ACM (2008), 285–294.