ISSN (Online) : 2277-5420

The Trikey-board - Introduction to and analysis of a new design of touch-sensitive virtual keyboard

[1]Sunil Joshi

Abstract
This paper introduces a new design and layout of touch-sensitive virtual keyboard, namely the Trikey-board. Through quantitative comparative analysis with the traditional keyboard, this paper aims to establish that the trikey-board facilitates in reducing typographical errors.

Keywords: *Trickey Board.*

1. Introduction

In this paper, we are going to discuss a new layout for the touch-sensitive virtual keyboards that are generally seen in smartphones of today's generation. Henceforth, the use of term 'keyboard' in this paper refers to touch-sensitive virtual keyboards. Further, we will do a quantitative comparison of the two keyboards to demonstrate that the trikey-board offers better typing experience over the traditional keyboard layout by reducing the chances of typo (i.e. error in typing).

First, let us try to understand why users sometime press a wrong key. Figure 1 shows layout of keys in a traditional keyboard where the keys are rectangular in shape. When a user taps on one of the keys, the finger touches a certain area of the screen. Let's assume this area to be of circular shape, and the software reads the touch event to be at the center of this circle. The circle in Fig. 1 indicates the area taken by a finger when tapping on the key.

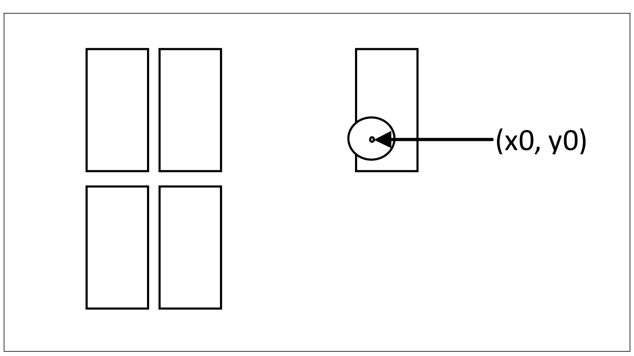

Fig. 1. Layout of keys in traditional keyboard and location of a key touch event represented by a circle centered at (x0, y0)

The co-ordinate (x0, y0) is the center of the circle, which will be read by the software as the point where user has tapped. If (x0, y0) lies within the boundary of the key, it is considered as a hit, while if (x0, y0) lies outside this boundary, the software will consider it as a miss and the key press will not be registered.

1.1 Introducing the Trikey

As is obvious, a trikey is nothing but a triangular key and trikey-board is a board made up of these trikeys. We can have two arrangements of trikeys as shown in Fig. 2. A variation of the latter of the two layouts in Fig. 2 can also be seen in the Crocodile Keyboard [1].

The Crocodile Keyboard has reduced the size of its keys so that user has more space between them to avoid mistyping. This however also results in less real-estate for each key. But so is the case with the trikey-board layouts of Fig. 2. In fact, on comparing with the traditional key the trikey has half the area. To understand this, let us fix the base of the trikey to be of same length as the

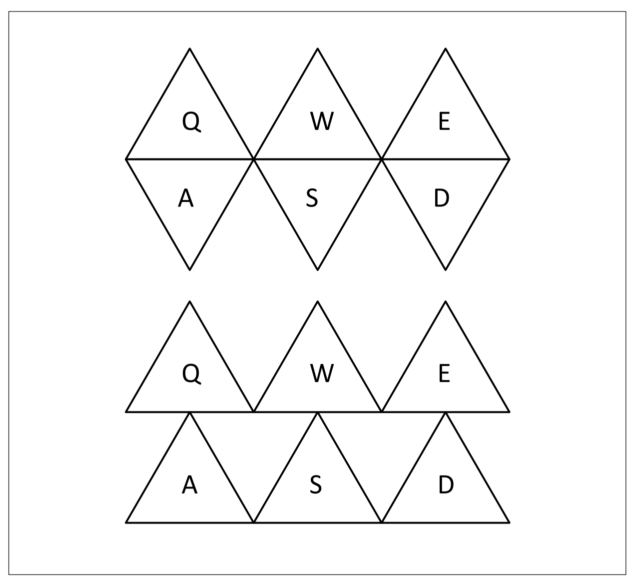

Fig. 2. Two possible layouts of trikeys

traditional key as shown in Fig. 3. Keeping the base of

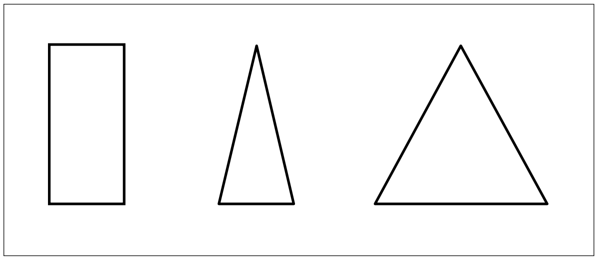

Fig. 3. A rectangular key, and two trikeys with base widths equal to and double of the rectangular key.

trikey to be of same length allows us to insert equal number of trikeys in a row as in the traditional keyboard. We know that area of triangle is half the area of a rectangle having the same base and height. So, to make the area of trikey to be equal to the rectangular key, let us double the length of the base, as shown by the second triangle in Fig. 3. This will now force us to have only half the number of trikeys in a row than earlier. To compensate for the loss in number of keys in a row, let us reverse alternate keys, which gives us the layouts as shown in Fig. 4.

2. Trikey-board vs. traditional keyboard

Now let us try to understand why we should choose trikey over the usual rectangular key. What advantages does trikey-board offer?

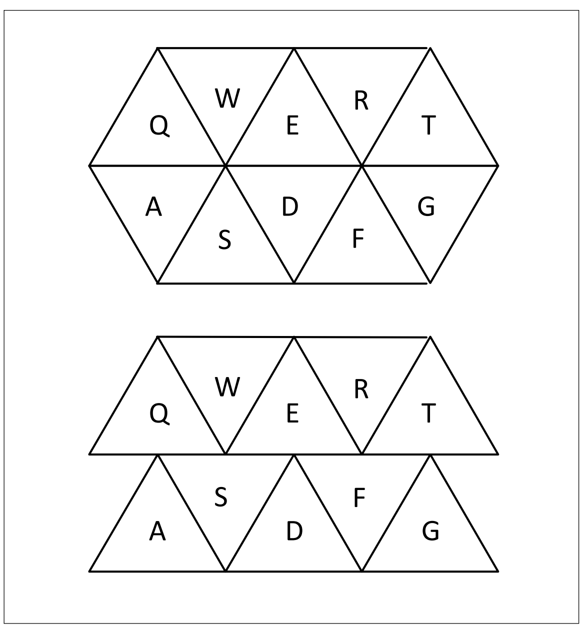

Fig. 4. Two possible layouts of a trikey-board

Before comparing the two keyboards, let us think of a single touch event. When user is about to press the key, generally he attempts to press at the center of the key. But since he is typing at a speed, his "hit" on the key may not fall exactly at its center. Let's define '**Ph**' as the probability of hitting within the boundary of the key and '**Pm**' as the probability of a miss when pressing a key.

Now we will try to deduce the affect of shape of the key on its **Ph** (or **Pm**). For this, let us draw a circle with its center coinciding with the center of the key and radius of maximum length such that the circle does not fall outside the key. This is called the incircle. Bigger the area of incircle, higher the value of **Ph**.

You may argue that even when user touches area outside the incircle but which falls inside the key, it will still be a hit. That's true, but here we are trying to analyze the "probability" of a hit, i.e. **Ph**. The incircle is certainly not the only area for a correct hit and **Ph** in itself will depend upon the shape and size of the key. But having a larger incircle area will definitely increase the probability of a hit.

For example, the value of **Ph** will be too low for an irregular shape like that shown in Fig. 5. But as we inflate it from its center, the area of it's incircle will also increase, which in turn will increase its **Ph**.

For any key in a virtual keyboard, to increase the **Ph**, the size of the key should be large. Assuming the height of the key to be of fixed length for a keyboard, we should try to increase the width of the key.
Now for the rectangular key, the **Ph** is decided by its width as it is less than the height. But it's not possible to increase the width of any key without compromising on another key. The trikey however achieves this easily. By arranging the trikeys in alternate reverse positions, we are able to give each key double the width.

There is also one more important difference between the rectangular key and trikey. In the former, the center lies

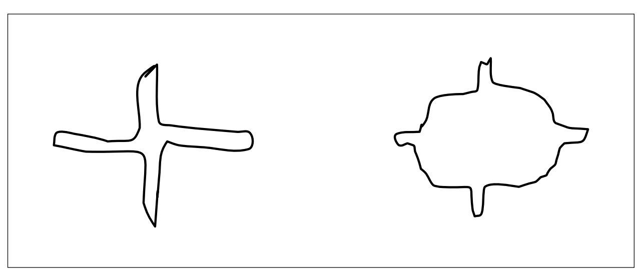

Fig. 5. Probability of hit (Ph) increases as we inflate the figure

in the middle of the key (mid-point of width and height).

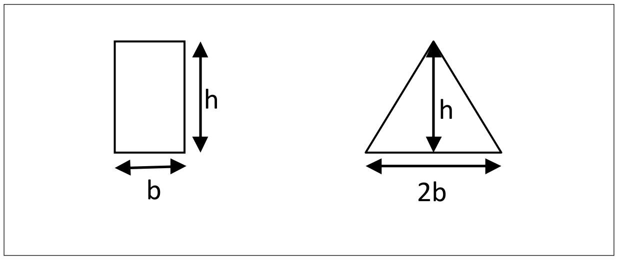

Fig. 6. Dimensions of rectangular key and trikey considered` here for comparison

In the trikey however, the center of the key is more towards the base than the top. Mathematically, the top vertex is twice the distance away from the incenter than the base for an isosceles triangle. This has an important consequence.

Remember that when a user is about to press a key, he tries to hit at its center. In case of the trikey, the center is more towards the base. Also, the width of the key is more towards the base than at the top. Because of this, the area of the incircle also increases for the trikey when compared to the rectangular key.

2.1 Comparison of areas of the incircles

Let's take the height and width of the two types of keys to be as shown in Fig. 6. Measurements of the traditional rectangular key:

Height = h
Width = b
Inradius = b/2
Incircle area = Sr = Pi * $b^2/4$

Measurements of the trikey:

Height = h
Base = 2b
Hypotenuse = $\sqrt{(b^2 + h^2)}$
Semi-perimeter = s = b + $\sqrt{(b^2 + h^2)}$
Inradius = r = b/h ($\sqrt{(b^2+h^2)}$ – b) … [2]
Incircle area = St = Pi * r^2

Let us calculate the ratio of the incircle areas of the two keys.

St/Sr = (Pi * r^2) / (Pi * $b^2/4$)
= 4 [$\sqrt{(1 + b^2/h^2)}$ – b/h] ^{2} (on simplifying)

To estimate the range of **St/Sr**, let's consider four cases 1) **b/h = 1** and 2) **b/h = ¾** 3) **b/h = ½** and 4) **b/h = ¼**.

St/Sr = 0.686 … case 1

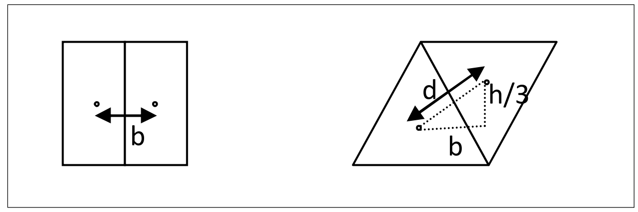

Fig. 7. Comparison of distance between center of adjacent keys in a row of traditional keyboard and trikey-board

St/Sr = 1 … case 2
St/Sr = 1.528 … case 3
St/Sr = 2.438 … case 4

As we can see, for **b/h > ¾**, the ratio increases above **1**. Generally, keyboards of smartphones have rectangular keys in portrait mode and will likely fall in the range of case 2 and case 3.

2.2 Further analysis

An error in typing may also happen when user mistakenly presses a key adjacent to the actual one in the same row. The distance between the centers of the two rectangular keys is **b**. But as can be seen in Fig. 7, for trikeys this distance is greater than **b**, which reduces the chances of wrong key presses.

Let us now compare the spatial arrangements of the center of the trikeys in the two possible versions of the trikey-board. As can be seen in Fig. 8, the first layout has the centers in the arrangement of vertices of hexagon. In this layout, the distance between the centers of the trikeys in adjacent rows is equal to **2r**.

The second layout in the Fig. 8 however has a more spread-out arrangement of the centers. The distance between two vertical trikeys in the two rows is same as that in the traditional keyboard, i.e. equal to **h** which is an improvement over the first layout.

Let us try to further analyze the second trikey-board layout. If we draw the incircle of all the trikeys, it will look like the layout in Fig. 9 (assuming the trikeys are all equilateral triangles here for simplicity). As we can see, this layout actually has the incircles placed alternately in two levels in each row. These circles are also at a distance from any of the circles in the adjacent rows.

Let us now try to identify those areas in the keyboard that when touched do not produce any key-touch event. Let's call these areas as 'dead-zones'. Ideally, the dead-zone should be an area that is either not falling under any

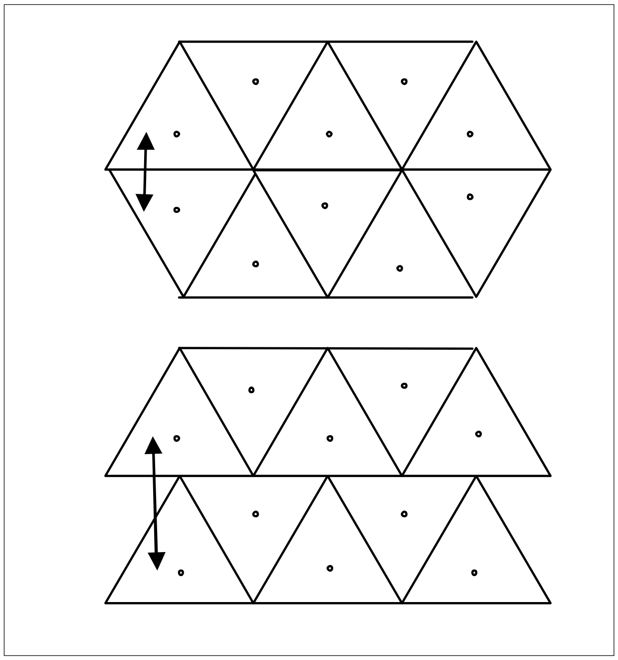

Fig. 8. Spatial arrangements of the center of trikeys in the two possible layouts

triangle, or is at an equal distance from more than one incenter. This would generally include all the edges of a trikey as it shares all its edges with other keys. However, in Fig. 9 we have marked those areas as dead-zones that fall at the point of convergence of the corners of three trikeys.

On comparing the dead-zones in the trikey-board with those in the traditional rectangular key-board, we can see that the number of dead-zones will be more in the latter. This is because the number of dead-zones depends upon number of corners of a key that are converging with those of other adjacent keys. Thus, this also contributes to improvement in trikey-board over the traditional one.

In the virtual keyboard, the dead-zone areas should form part of the border and be visually indicated so that user is aware that tapping on those areas will not register any key event. This will also improve the speed of typing because when user taps on such an area, he will not have to press backspace again to delete the wrong character.

4. Conclusion

In this paper, we introduced the trikey-board and compared it with the traditional keyboard. With the trikey having area equal to that of the traditional key (**= bh**), the

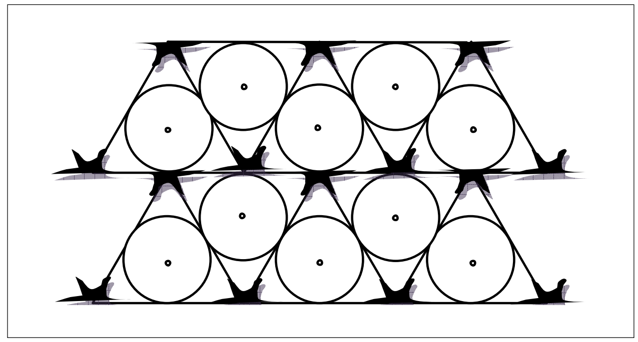

Fig. 9. Final layout of the trikey-board depicting incricles, incenters and dead-zones.

trikey-board offers the following advantages over the traditional keyboard:

1. Bigger incircle area for **b/h < ¾** which will result in reduced **Pm**, i.e., probability of the point of actual impact falling outside the incircle.
2. More distance between the centers of keys placed adjacent in a row.
3. Lesser number of dead-zones around a key.

The advantages listed above should result in lesser typing errors and give user a better typing experience with the trikey-board.

References

[1] http://phandroid.com/2009/12/03/crocodile-keyboard-coming-december-7th/ and http://crocodilekeyboards.com/
[2] http://en.wikipedia.org/wiki/Incircle_and_excircles_of_a_triangle

Sunil Joshi received a degree of B.E. in Electronics and Communications from N.S.I.T., Delhi in 2004. He works at Adroit Business Solutions Pvt. Ltd., NOIDA as a Sr. Team Lead and is involved in development of multimedia based mobile applications on Android.