# An Automatic GUI Adjustment Method for Mobile Computing

HaiTao Song, HuanYu Liu, Dongyi Chen
School of Automation Engineering
University of Electronic Science and Technology of China
Chengdu, 611731, China
Email: saga.song@gmail.com

***Abstract*—A variety of Mobile devices such as small screen Netbook, Ultra-mobile PC (UMPC), smart phone, Tablet PC are becoming more prevalent in our daily use. These devices usually mounted with cameras in their front panels, and occupy small portions in user eyes' Field of View (FOV). Accordingly, the screen contents displayed on these screens are comparatively smaller than those on desktop LCDs. Sometimes, the small sized screen contents may affect the performance of user interaction. In this paper, we propose an automatic GUI (graphical user interface) adjustment method for preventing the performance decrease when interacting with mobile screens. Firstly, we propose an algorithm to find out how to scale the GUI (graphical user interface) content size for mobile screens. Secondly, we use a Fitts' pointing task to determine a critical size, which reflects in what condition the GUI content is too small and need to be adjusted. Thirdly, we discuss an implementation for this method, which based on human face tracking. We conclude with an evaluation experiment and the results show that our adjustment method could work for mobile screens under different display FOV conditions.**

***Keywords-Fitts' Law; Mobile Computing; GUI; face tracking***

## I. INTRODUCTION

Interacting with mobile, small screen devices has become an integral part of our daily lives. The small contents on those screens are sometimes appears even much smaller depending on how the user holding/watching it. E.g. watching the screen under a big Eye Screen Distance (ESD), viz. under a small display FOV. The relationship between ESD, display FOV and visual angular size (VAS) of target width/size ($W$) is shown in Equation1. In which, $dLength$ is the physical screen length units in millimeter, $dFOV$ is the screen FOV units in degree, $W_V$ is the VAS of $W$, $W_P$ is the pixel size of $W$, $dRes$ is the screen resolution. For the sake of computational convenience, we set $dLength$, $dFOV$ and $dRes$ as the horizontal measurement when holding the screen.

$$\begin{cases} dFOV = 2 \cdot arctan\left(\dfrac{dLength}{2 \cdot ESD}\right) \\ W_V = dFOV \cdot \dfrac{W_P}{dRes} \end{cases} \tag{1}$$

There is evidence that the decrease in content size may result in the decrease in user interaction performance. Prior work has shown that GUIs which have small targets are difficult to interact with. Sutter and Ziefle [1] found that $W$ had a more significant impact on movement time ($MT$) than target distance/amplitude ($D$) and that $MT$ was disproportionally longer for small targets. Wobbrock et al. [2] observed that the commonly held assumption that as index of difficulty ($ID$) decreases, error rates decrease did not hold for small $W$, $D$ and $W$ did not contribute proportionally to error rates in pointing tasks.

Thus, the user experience could be enhanced by adjusting (scaling) the GUI, when necessarily and appropriately. However, as to our knowledge, few work has been down on this.

In this paper we propose a method to supplement this requirement. First, for accommodating GUIs across various ESDs, screen resolutions and screen dimensions, we proposed an algorithm to scale the GUI components into some appropriate sizes which would balance them neither too small for interacting with, nor too big for layout consideration. Second, we conducted a Fitts' experiment under different ESDs to determine to what extent the component is too small to interact with and thus needed to be adjusted. Third, we proposed a method to automatically adjust the GUI by detecting the ESD through natural human face tracking. We concluded with the evaluation for this method and the results show that the method works for adjusting the GUI according to the mobile computing requirement.

Fitts' Law [3] is an extensively used technique to evaluate human motor performance in a variety of pointing tasks and as a method for input device or GUI evaluation. It describes $MT$ as a function of index of difficulty, according to the Equation 2. Where $ID_e$ is effective index of difficulty, and could be computed by effective $D$ ($D_e$) and effective $W$ ($W_e$). The $W_e = 4.133\sigma$ and $\sigma$ is the standard deviation of each recorded $W$, the $D_e$ is the mean value of each recorded $D$[4].

$$\begin{cases} MT = a + b \cdot ID_e \\ ID_e = log_2(1 + \dfrac{D_e}{W_e}) \end{cases} \tag{2}$$

## II. AN ALGORITHM FOR SCALING MOBILE GUI'S COMPONENTS

Assuming there is a special size for a GUI element (e.g. an Icon), any size which is smaller than it would let a user feel difficult to interact with, we then name it Critical Size (CS).

There are two method for determining the $W_{CS}$ value for a given target on a mobile screen. First, we could conduct the

**978-1-4244-8728-8/11/$26.00 ©2011 IEEE**

user Fitts' study on a small screen device under a normal operation distance (ESD) and a recommended minimal $W$ condition. Then the $W_{CS}$ value could be resulted from the Fitts' performance analysis. Second, the $W_{CS}$ value could be chosen by user themselves according to their own visual conditions.

The CS should be a perceived size, viz. $W_{CS}$, and be measured in the unit of visual arc minute ($'$). As a target would appear in different size depending on how a user observing it (e.g. at different ESD). Ideally, we may set the $W_{CS}$ as the preferred minimal element size for the GUI as it would both benefit for interacting with and for maximizing the usability of GUI layout space. Then this $W_{CS}$ value could be used as a benchmark and should be managed to kept unchanged before and after switching display FOV (ESD).

Assuming an optimal sized GUI which has the target size of $W_{CS}$ has been adopted. The adjustment procedure could be achieved following the steps in Figure1.

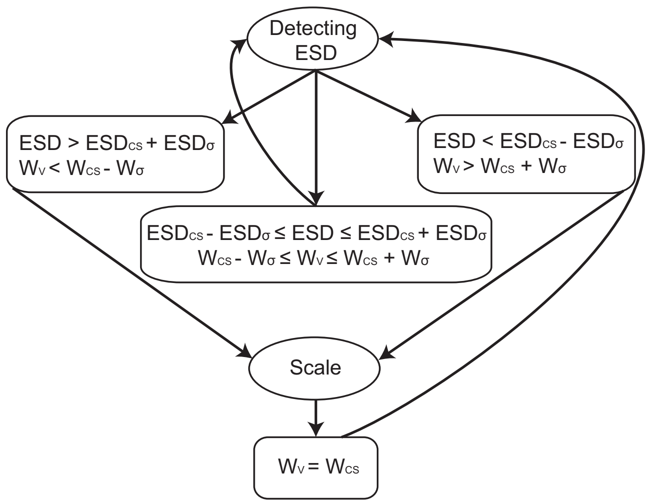


Figure 1. The adjustment procedure

As in the figure, the $W_V$ should be scaled to $W_{CS}$ if it is detected too small ($ESD > ESD_{CS} + ESD_\sigma$, where $W_V < W_{CS} - W_\sigma$), or too big ($ESD < ESD_{CS} - ESD_\sigma$, where $W_V > W_{CS} + W_\sigma$). Otherwise, it should be kept if it is in an appropriate size ($ESD_{CS} - ESD_\sigma \leq ESD \leq ESD_{CS} + ESD_\sigma$, where $W_{CS} - W_\sigma \leq W_V \leq W_{CS} + W_\sigma$). In which, $W_\sigma$ is the out of boundary $W$ tolerance the user would like to set (e.g. 20% of the $W_{CS}$).

When the ESD is keeping change in a sequence, the whole procedure could be regarded as a Markov process, and could be presented in Equation 3, 4 and 5. Where, $ESD_{CS}$ is the ESD under the condition of $W_V = W_{CS}$, $F_{SCALE}$ is the scaling factor, which would be used to adjust the GUI property of pixels per element (PPE), or GUI scaling/zoom in function through the operation system.

$$W_{CS} = 2 \cdot \frac{W_P}{dRes} \cdot arctan\Big(\frac{dLength}{2 \cdot ESD_{CS}}\Big) \tag{3}$$

$$W_V = 2 \cdot \frac{W_P}{dRes} \cdot arctan\Big(\frac{dLength}{2 \cdot ESD}\Big) \tag{4}$$

$$F_{SCALE} = \frac{W_{CS}}{W_V} = \frac{arctan(\frac{dLength}{2 \cdot ESD_{CS}})}{arctan(\frac{dLength}{2 \cdot ESD})} \tag{5}$$

After providing the algorithm for the GUI adjustment, we now turn to determine the $W_{CS}$ value and discuss the ESD detecting method which should be convenient for most of the existing mobile devices.

## III. Experiment

### A. Design

As aforementioned, the $W_{CS}$ could either be investigated by a Fitts' task performance, or determined by user's preferred choice. Here we discuss the first method.

A Sony UX17C with Windows XP SP2 and JDK 6.0 was used for the experiment, it has a screen resolution of $1024 \times 600$ with a dimension of $105 \times 55$ mm. The screen brightness were set to a fixed comfortable level in the room, which was confirmed by each participant. The optical mouse (Logic M505) was used as the input device, rather than Twiddler, trackball etc, which are usually used in mobile/wearable computing context but would involve with learning effects. A custom Java program (See Figure. 2) was used for the experiment. The program could record the experiment results in the form of MSExcel file, detect outlier, and compute the effective ID as well as error rate and $MT$. Our program used green as the color for the starting target (ST), red for the end target (ET) and light gray for the background [5].

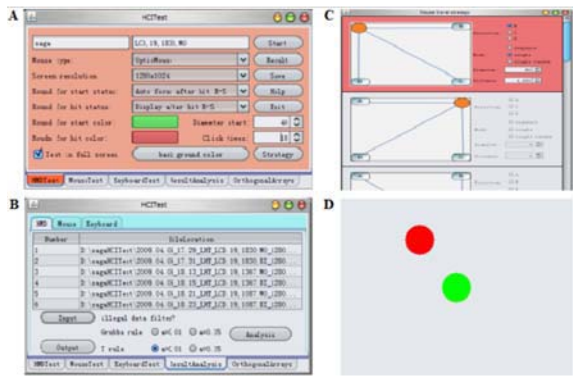


Figure 2. Our testing software: (a) Experiment settings, (b) Data processing, (c) Pointing Task Type, (d) Participant testing screen

We used the discrete Fitts' pointing task in our experiment. In this pointing task, two targets, a starting (ST) and an ending target (ET) are set to a round shape. Initially, only the ST is visible. Participants first click on the ST in the center of the screen to position the cursor at the exact center of the ST. Then participants click the target again to indicate they have ready for the trial. After a random amount of time between

| Factors | Levels |
|---|---|
| dFOV (degree) | 4, 5, 7, 10, 11.5, 15.7, 17.2, 25.6 |
| W (pixels) | 12 |
| D (pixels) | 294 |

TABLE I. Factors and levels for the experiment

0.1 to 1 seconds, the ET will appear somewhere along the circle determined by the $D$. After the participant has visually acquired the ET, they again click the ST and move to the ET and click it as soon as possible but also as accurate as possible. Clicking the ET completes one trial. Requiring users to click after visually acquiring the target was done to separate the visual acquisition time and reaction time from $MT$, as we believed that when $W$ was small the random target visual acquisition time and reaction time could pollute the $MT$[6].

We set the ST to 40 pixels diameter and ET as 12 pixels diameter. Then, the target appeared to the viewer as 4.7′ when under the display FOV of $4°$. This value is still bigger than the human eye visual acuity which is about 0.5'–1' [7]). The $D$ were chose as the largest possible value for the given $W$ (the ET would appear at the edge of the screen) for achieving the possible biggest $ID_e$ value.

### B. Procedure

Twelve college-students subjects (7 male and 5 female, mean age 27 years old) participated in the experiment over the course of two days in return for monetary compensation. All participants claimed to have normal or corrected to normal (glassed) visual acuity and gave their informed consent which was approved by the local institutional review board prior to test.

Participants were given written and verbal instruction on how to use the testing software, and performed 12 runs as a training course. There were 8 conditions in the experiment, and each conditions was required to successfully repeated 25 times to ascertain the central tendency of each subject [4]. The order of the conditions was counter-balanced for avoiding learning effect [4]. Participants were allowed to take a break between conditions if they wished. The factors and levels are shown in Table I.

### C. Results

We used Dixon's Q Test ($Q = 99\%$) method [8] to investigate and remove outliers from the 25 repeated trials of each condition, and the results are shown in Fig. 3. In which, $TP$ is the bandwidth which been extensively used to evaluate the over-all Fitts' task performance. The definition of $TP$ is shown in Equation 6, where, $y$ represents the number of subjects, $x$ is the number of experiment conditions, the units of $TP$ is bits per second (bps) [4].

$$TP = \frac{1}{x}\sum_{i=1}^{x}\left(\frac{1}{y}\sum_{j=1}^{y}\frac{IDe_{ij}}{MT_{ij}}\right) \tag{6}$$

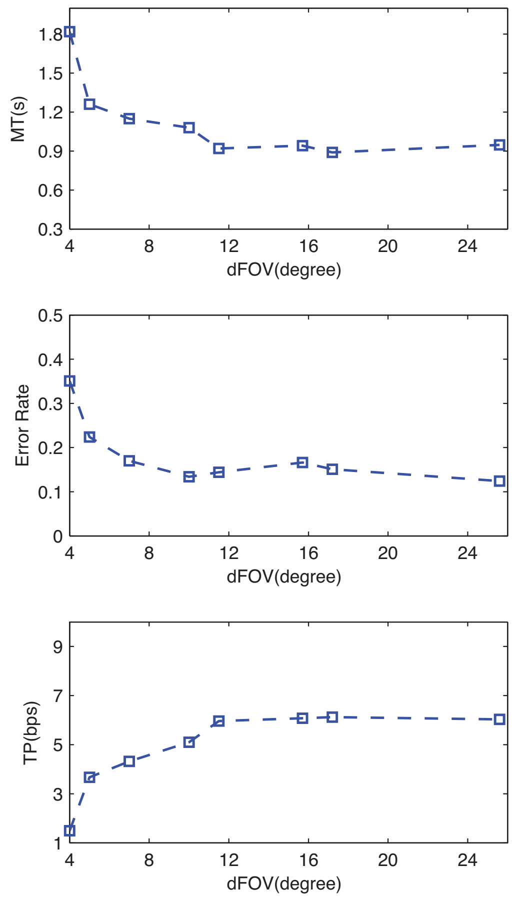


Figure 3. Results of MT, ER and $TP_i$ under different $dFOV$ without GUI adjustment

As can be seen in each of the subfigure, the curves are strongly bended when $dFOV$ decreases beyond an inflection point. This indicates the critical size for target which we mentioned in Section II. And not surprisingly, the participants repeatedly reported that the pointing task was much more difficult when executing tasks in a condition with a $dFOV$ smaller than that inflection point.

We then again used the Dixon's test to determine at which point along the $dFOV$ is the inflection point, and where the $W$ is the CS value . This was done by testing from the third points (due to the sample quantity requirement) starting with our biggest $dFOV$ and towards our smallest $dFOV$ until the Dixon's method showed that there was an outlier. Our analysis showed that the CS occurs when $dFOV$ were $10°$ thus resulting the visual angular size of $W$ as $12'$ at that point ($W_{CS}$).

## IV. Implementation

Given the result of $W_{CS}$, we now turn to detect the ESD to implement our adjustment algorithm. Since the frontal camera is becoming more common on mobile devices (such as iphone4, most of the netbook and UMPC etc.), we choose to obtain the ESD through human natural face tracking. The natural feature we used in the tracking is the horizontal length of our eyes-pair.

We used the Haar classifier in OpenCV [9], which based on AdaBoost (adaptive boosting) to track the eye pairs on human face. AdaBoost classifier method was proposed by Y. Freund and R. Shapire [10], it is a supervised classifier and uses edge features, line features and center-surround features [11], [12] as the Haar-like basis features.

The training procedure aims to achieve a strong classifier out of a set of weak classifiers. This procedure could be presented in the following steps[10], [13]:

- Collect the training set $(x_1, y_1), \cdots, (x_m, y_m)$. Where $x_i$ is the training front picture of human eyes (labeled by $y_i = 1$) or background pictures without human eyes (labeled by $y_i = -1$);
- Set the training in $t = 1, \cdots, T$ rounds, and the weight distcibution of $x_i$ as $D_t(i)$. Set the initial value ($t = 1$) of $D_1(i)$ as $1/m$;
- For $t = 1, \cdots, T$, find a $h_t = \underset{h_t \in H}{argmin} \varepsilon_t$, where $\varepsilon_t = \sum_{i=1}^{m} D_t(i)[y_i \neq h_t(x_i)]$;
- If $\varepsilon_t \geq 0.5$ then stop and set $\alpha_t = \frac{1}{2} log(\frac{1-\varepsilon_t}{\varepsilon_t})$;
- Update $D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$, where $Z_t$ is a normalisation factor ;
- The final strong classifier: $H(x) = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big)$

After we finished the training and got the classifier file (the xml file), the eye pair was easily be detected by the class of *cvHaarDetectObjects* in OpenCV. The information that we could directly get from the tracking is the eye-pair's length (EPL) which measured in pixels. We run our program on the UMPC as shown in Figure 4.(a). In addition, the detection which running on an iphone would looks like in Figure 4.(b).


(a) (b)

Figure 4. The tracking procedure running on mobile devices (a) UMPC, (b) iphone simulator

Assuming the field of view of the frontal camera is $cFOV$, it could either be obtained from the producer or measured by the user according to Equation 7. Where, $cLength$, which units in millimeter, is the horizontal length of a given article (e.g. a marker or a ruler). $TSD_{full}$ is the distance between that article and the screen, when the horizontal dimension of the article just fulfill the horizontal direction of the screen. As in our case, the $cFOV = 30°$.

$$cFOV = 2 \cdot arctan\left(\frac{cLength}{2 \cdot TSD_{full}}\right) \tag{7}$$

The ESD detection was functional all the time in our method. It records every EPL values which measured in pixels ($EPLp$). Then the relationship between $cFOV$, $EPLmm$ (EPL measured in millimeter) and ESD etc. is shown in Equation 8. Where, $cFOV_{EPL}$ (units in degree) represents the proportion that EPL taken in the $cFOV$, subscribe $_{CS}$ means the variable is under the state of $W_V = W_{CS}$.

$$\begin{cases} \frac{cFOV_{EPL}}{cFOV} = \frac{EPLp}{dRes} = \\ \frac{2}{cFOV} \cdot arctan\left(\frac{EPLmm}{2 \cdot ESD}\right) \\ \frac{cFOV_{EPL\_CS}}{cFOV} = \frac{EPLp_{CS}}{dRes} = \\ \frac{2}{cFOV} \cdot arctan\left(\frac{EPLmm}{2 \cdot ESD_{CS}}\right) \end{cases} \tag{8}$$

As the $EPLp$ ($EPLp_{CS}$), $dRes$ and $EPLmm$ are all given, the "real-time" $ESD$ and the $ESD_{CS}$ can all be calculated out. In which, $ESD_{CS}$ should be calculated according to Equation 8 when under the condition of choosing the $W_{CS}$ by user. However, if the $W_{CS}$ is determined by the Fitts' task performance (or by our results of $12'$), the Equation 1 should be used instead. After this, the scaling factor $F_{SCALE}$ as in Equation5 can be calculated out, and then the PPE of the operation system could be adjusted according to the $F_{SCALE}$.

## V. Evaluation

We then conducted another experiment to evaluate the performance under different PPE which been determined by the $F_{SCALE}$. We used the same experimental setup as in the above experiment. We used the $W_{CS}$ value of $12'$ (where $ESD_{CS}$ = 600 mm) and the mean $EPLmm$ = 126 mm. The result is shown in Figure 5.

As can be seen in the figure, there is no such big bend on the curves, and thus indicates that the influence of the small $dFOV$ (big ESD) is generally eliminated by our adjustment method.

## VI. Conclusions

In summary, a method for automatically adjusting the GUI contents for mobile devices is presented in the current paper. The evaluation results show that our adjustment method could work under different display field of view (or eye to screen

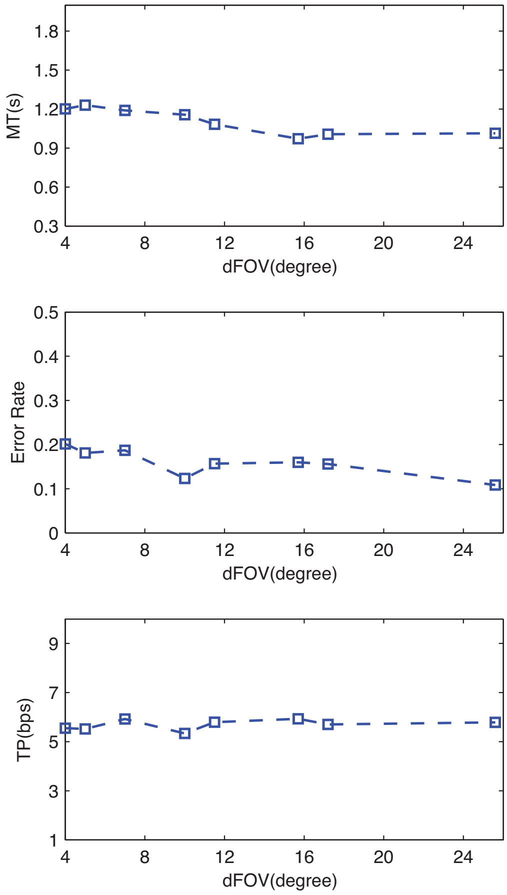


Figure 5. Results of MT, ER and $TP_i$ under different $dFOV$ with GUI adjustment

distance) conditions. However, the real-time performance of the current implementation should be enhanced.

## ACKNOWLEDGEMENT

This work is founded by the National High Technology Development 863 Program of China under the grant NO.2009AA01Z310, and International S&T Cooperation Program from Ministry of Science and Technology of China under the grant NO. 2009DFA12100. The authors would like to thank Xinyu Li, Shiji XiaHou, Zhiqi Huang for their help in data collection.

## REFERENCES

[1] C. Sutter and M. Ziefle, "Psychomotor efficiency in users of notebook input devices: Confirmation and restrictions of fitts' law as an evaluative tool for user-friendly design," in *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 2004, pp. 773–777.

[2] J. Wobbrock, E. Cutrell, S. Harada, and I. MacKenzie, "An error model for pointing based on fitts' law," in *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 2008, pp. 1613–1622.

[3] P. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement." *J Exp Psychol*, vol. 47, no. 6, pp. 381–391, 1954.

[4] R. Soukoreff and I. MacKenzie, "Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci," *Int J Hum Comput Stud*, vol. 61, pp. 751–789, 2004.

[5] L. MacDonald, "Using color effectively in computer graphics," *IEEE Comput. Graph.*, vol. 19, pp. 20–35, 1999.

[6] H. Munro, M. Plumb, A. Wilson, J. Williams, and M. Mon-Williams, "The effect of distance on reaction time in aiming movements," *Exp Brain Res*, vol. 183, no. 2, pp. 249–257, 2007.

[7] C. Cavonius and A. Schumacher, "Human visual acuity measured with colored test objects." *Science*, vol. 152, no. 726, pp. 1276–1280, 1966.

[8] R. Dean and W. Dixon, "Simplified statistics for small numbers of observations," *Anal Chem*, vol. 23, no. 4, pp. 636–638, 1951.

[9] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly,CA, 2008.

[10] Y. Freund and R. Shapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, 1995, pp. 23–37.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 511—518.

[12] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," *IEEE ICIP*, pp. 900—903, 2002.

[13] J. F. T. Hastie, R. Tibshirani, *The Elements of Statistical Learning*. Springer-Verlag, New York, 2001.