Alfred Schmitt & Peter Oel (1999). Calculation of Totally Optimized Button Configurations Using Fitts' Law. *Proceedings of the HCI International '99 Conference on Human-Computer Interaction*. Munich, Germany, August 22-26, 1999. Volume 1, 392-396. LEA, Mahwah, New Jersey, 1999.

Calculation of Totally Optimized Button Configurations Using Fitts' Law

Alfred Schmitt and Peter Oel
University of Karlsruhe
Institute for Operating and Dialogue Systems
Kaiserstr. 12, 76128 Karlsruhe, Germany
Phone: +49 (0)721 608-3965, Fax: +49 (0)721 696989
Email: {aschmitt,oel}@ira.uka.de

1 Introduction

This paper addresses the problem of optimal placement and configuration for a set of buttons on a graphical user interface (GUI). We present an optimization algorithm that reduces the average time necessary for pointing device motions. Fitts' law (Fitts 1954) is used to estimate the transfer time for cursor movements between buttons. Numerous studies in the literature focus on the verification and application of Fitts' law in a rather direct way (MacKenzie 1992). Our work differs in its goal: The problem of finding totally optimized button configurations is a non-trivial optimization task and is still an open issue.

Our problem can be stated as follows: Given is the number n of buttons and corresponding absolute probabilities $w(i,j)$ for user induced cursor movements from button i to button j. We are looking for a button configuration under the following conditions and constraints:

- the width of each square button may vary,
- each button may be positioned freely within a rectangular area,
- no two buttons may overlap each other,
- the average time calculated by Fitts' law for lengthy button clicking sequences is minimal.

The suggested approach is rather theoretical but its results will provide some practical insight into the placement and relative size of buttons if a total optimization based exclusively on Fitts' law is aspired.

The rest of this paper will shortly describe our solution to this optimization problem, discuss the results, and draw some conclusions.

2 Button configurations and Fitts' Law

Each button B_i is described by the coordinates of its central point (x_i,y_i) and its width $2r_i$ (see figure 1). For a concise approach to the problem we only consider square buttons. A tuple of buttons is called a button configuration. The desired configuration $\boldsymbol{C}$ consists of n buttons whose x_i, y_i, r_i have to be determined. According to Fitts' law the time to move to and to select a target of width W at a distance A is $TM = a + b \cdot \log(2A/W)$ where a and b are empirically determined well-known constants. For two buttons B_i and B_j the time $T_{pos}(i,j) = \log(d_{ij}/r_j)$ is calculated using Fitts' Law with target width $W = 2r_j$ and distance $A = d_{ij}$ (figure 1). We only consider the term $\log(d_{ij}/r_j)$ in Fitts' law, since constant factors have no influence on the results of our optimization process. Variations of Fitts' law (Card et al. 1983) like the ones proposed by Welford (Welford 1968) and MacKenzie (MacKenzie 1989) produce only very small divergences.

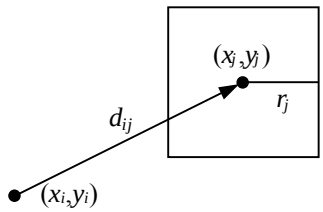

Figure 1: Movement of a user's hand to a target.

The user's behavior and his preferred traversal between the buttons induced by the graphical user interface depends on the application specific tasks to be solved. The absolute probabilities $w(i,j)$ for a movement from button B_i to Button B_j define the specific input data necessary to determine the average time $T_{avg}(\boldsymbol{C})$ for a given button configuration $\boldsymbol{C}$:

$$T_{avg}(\boldsymbol{C}) := \sum_{i=1}^{n} \sum_{j=1}^{n} w(i,j) \cdot T_{pos}(i,j)$$

3 Optimization

Finding a minimum of $T_{avg}(\boldsymbol{C})$ is a classical optimization problem. The search for optimal configurations is accomplished by simulated annealing (Kirkpatrick et al. 1983), a probabilistic approach, combined with a gradient method and a specific penalty function. The penalty function ensures that only non overlapping configurations are considered. The penalty for a configuration $\boldsymbol{C}$ is defined as

$$Pen(\boldsymbol{C}) := \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} ov(i, j)$$

where $ov(i,j) := max\{0,r_i+r_j-|x_i-x_j|\} \cdot max\{0,r_i+r_j-|y_i-y_j|\}$ is an approximation of the area of intersection of the buttons B_i and B_j. The term $ov(i,j) = 0$ holds, if the buttons do not overlap.

The gradient of T_{avg} and *Pen* is used to get better values for $T_{avg}(\boldsymbol{C})$ and to obtain $Pen(\boldsymbol{C}) = 0$ during the optimization. Furthermore buttons are swapped if T_{avg} is reduced by such a change of the configuration. This ensures not getting stuck in local minima.

Another condition is of a more practical nature: The total area of all n buttons is limited to a predefined value. If one button grows, others must shrink. Fixing the total area of all buttons does not influence the result of the optimization. $T_{avg}(\boldsymbol{C})$ is invariant to scaling and moving in 2D since scaling factors are lost in the term d_{ij}/r_j and translation offsets sum up to 0 in d_{ij}.

4 Optimization Results

Figure 2 shows two optimized configurations of a set of 30 buttons. The probabilities w(i,j) used in this scenario reflect the user's behavior during a typical dialogue. The buttons numbered by 2, 3, and 5 have the highest probabilities of being visited by the cursor, whereas button 1, 16, and 17 have the lowest. Optimization for this scenario leads to the following numerical results (note that values for T_{avg} do not represent time in the sense of Fitts' *TM*):

a) $T_{avg}(\boldsymbol{C}_a)$ = 1.6566 (equals 100%). No optimization: All buttons have the same shape and are arranged sequentially in rows of six buttons.

b) $T_{avg}(\boldsymbol{C}_b)$ = 1.2403 (equals 74.9%). Buttons are exchanged pairwise to reduce T_{avg}. The size and row-column arrangement of buttons remains fixed. The result of this standard optimization using Fitts' law is shown in figure 2 on the left side.

c) $T_{avg}(\boldsymbol{C}_c)$ = 1.1023 (equals 66.5%). Total optimization with the final configuration shown in figure 2 on the right side.

The relative improvement from b) to c) is 11.35%. These results show the superiority of totally optimized button configurations to standard optimizations with fixed button size.

Why do buttons at peripheral positions grow so significantly?

The totally optimized configuration looks quite confusing. We first thought of an error in the optimization program, but looking closer at the calculation of T_{pos}

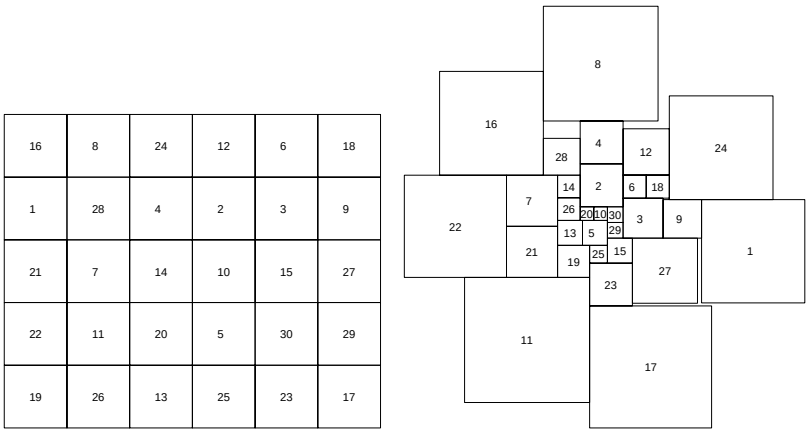

Figure 2: Two optimized button configurations, both consisting of 30 buttons. Each button is shown as a square containing the button number. In configuration C_b, on the left side, all buttons remain with same width. The optimization is done only by changing the position of the buttons. The configuration C_c, on the right, is totally optimized.

for buttons in the outer region will give the answer to the question. Let B_j be such a button and B_i a button where cursor movements to or from B_j will start or end. Terms of interest are $T_{pos}(i,j) = \log(d_{ij}/r_j)$ and $T_{pos}(j,i) = \log(d_{ij}/r_i)$, with distance $d_{ij} = r_j + d$. Finding a minimum for $T_{pos}(i,j) + T_{pos}(j,i)$ needs solving

$$\frac{d}{dr_j}\left(\log\left(\frac{d+r_j}{r_j}\right)+\log\left(\frac{d+r_j}{r_i}\right)\right)=0$$

which leads to $r_j = d$. The greater the distance between B_j and B_i, the larger the width of button B_j will be. Note that this also stops buttons in the interior region from getting bigger. This can be seen in figure 2 on the right side. Button 27 does not use the space on its right or beneath itself, because there are no possible traversals from or to the buttons 1, 11, 17 and 23.

5 Conclusions

In this paper we showed that non grid oriented placement of buttons and the variation of the button width depending on the probabilities for button traversals results in a better average time calculated using Fitts' law. The following list summarizes the main results derived from the structure of the totally optimized configuration:

- Buttons with higher probabilities $w(i,j)$ tend to move to the center of the configuration.
- Button size increases significantly from the center of the configuration to peripheral positions.
- The algorithm tries to position buttons in order to minimize the distances between probably subsequent buttons as indicated by $w(i,j)$ during a series of cursor movements.

It is quite difficult to derive practical implications, since the work is rather theoretical. At the moment we do not consider any space between the buttons. They are placed together as tight as possible. Adopting the optimization algorithm to consider free spaces between the buttons will give a more precise answer to special GUI design tasks, e.g. the design of control panels, where detailed characteristics for usage are known.

6 Acknowledgements

Finally we want to thank our students Jutta Klein, Robert Henßen, and Roland Heinemann for their implementation of the optimization algorithm.

7 References

Card, S., Moran, T. & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale (NJ): Erlbaum.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-391.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

MacKenzie, I. S. (1989). A note on the information-theoretic basis for Fitts' law. *Journal of Motor Behavior*, 21, 323-330.

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7, 91–139.

Welford, A. T. (1968). *Fundamentals of skill*. London: Methuen.