
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Sarcar, Sayan; Jokinen, Jussi P.P.; Oulasvirta, Antti; Wang, Zhenxin; Silpasuwanchai, Chaklam; Ren, Xiangshi

Ability-based optimization of touchscreen interactions

Published in:
IEEE Pervasive Computing

DOI:
[10.1109/MPRV.2018.011591058](https://doi.org/10.1109/MPRV.2018.011591058)

Published: 01/01/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Sarcar, S., Jokinen, J. P. P., Oulasvirta, A., Wang, Z., Silpasuwanchai, C., & Ren, X. (2018). Ability-based optimization of touchscreen interactions. *IEEE Pervasive Computing*, 17(1), 15-26.
<https://doi.org/10.1109/MPRV.2018.011591058>

Ability-Based Optimization of Touchscreen Interactions

Sayan Sarcar, Jussi Jokinen, Antti Oulasvirta, Zhenxin Wang, Chaklam Silpasuwanchai,
and Xiangshi Ren, *Senior Member, IEEE*

Abstract—The paper examines a computational design approach for improving user interface designs for people with sensorimotor and cognitive impairments. In ability-based optimization, designs are created by an optimizer and evaluated against model of an individual performing tasks. Alternative designs can be explored and adapted to an individual's abilities. In this paper, we explore text entry on touchscreen devices as the case. Individual abilities are parametrically expressed as part of a task-specific cognitive model, and the model estimates how the individual might adapt her interaction to the task. Optimized designs can potentially improve speed and reduce error for users with tremor and dyslexia. Ability-based optimization does not necessitate extensive data-collection and could be applied both automatically and manually by users, designers, or caretakers.

Index Terms—Ability-based design; model-based UI optimization; tremor



1 INTRODUCTION

FOR reasons of social inclusion and cost-efficiency, people with disabilities should be able to adequately use off-the-shelf smartphones. The reality is that they face everything from mild to prohibitive challenges exacerbated by the properties of touchscreen devices. Users with motor impairments (e.g., tremor) have problems with accuracy, strength, and coordination. These are made worse by the small size and absence of physical buttons [1], [2], [3]. Visual impairments are similarly problematic, compromising reading, pointing, and inference on the small display [4]. Dyslexia, memory deficits, and other cognitive impairments hamper proofreading and the ability to orient and multi-task.

This paper investigates methodology for *ability-based design* [5], which aims at identifying personal abilities as the basis for design. Previous work has applied the approach, among others, to identify interaction styles [1], [2] and methods for input [6], error reduction [7], and contextual adaptation [8]. Duff et al. [1] conducted experiment with motor impaired users performing number entry task on touchscreen found they faced more problems in front orientation of the finger tapping than normal users. Trewin et al. [7] interviewed a group of people with dexterity impairment and found they prefer tablet devices and many of them felt that touch-screen devices physically easy to use but require significant visual effort.

In particular, we are interested in computational methods to support design. Design optimization uses algorithmic search to identify solutions that are optimal for some user goal [9]. In *ability-based optimization*, the goal is to find designs that better suit some particular individuals or groups. In principle, this would allow automating parts of design, assisting designers, and adapting interfaces based on sensor data. However, much work needs to be completed to extend this approach to touchscreen interactions.

In previous work, SUPPLE optimized widget layouts for users with vision and motor impairments [10]. Their approach used simple models of motor performance (Fitts'

law) combined with heuristics (rules-of-thumb) for visual impairments, such as “users with poor vision need a larger font”. However, with heuristics, the validity of predictions is called into question. Conflict resolution also poses a problem: It is impossible to say how much one design factor can be changed without compromising another. To extend this approach beyond widget selection, the complex interplay of design choices and user behavior must be addressed.

Our approach is to use a task-specific cognitive model similar to the familiar models in HCI research, such as KLM, GOMS, ACT-R, or EPIC. This improves validity and allows collapsing the optimization task into a single objective. Moreover, more complex tasks can be addressed. However, several challenges stand out in modeling users with disabilities.

In this paper, we discuss how to address impairments, such as in working memory capacity or visual acuity, via explicit model parameters adopted from the literature, clinical assessments, or user/carer estimates. We also address how these abilities affect user behavior in a task. We build on existing work on rational analysis and computational rationality to identify the optimal interactive behavior [11], [12]. This provides an estimate of upper bound to an individual's performance with a given design.

We apply the approach in the case of text entry on touchscreen devices, a complex interactive task. Few text entry interfaces were developed for both able-bodied users [13], [14] and people with motor disabilities [15], [16], where dynamic adaptation was employed to improve the speed and accuracy of text entry. But these papers did not employ model-based optimization approach to achieve adaptiveness. The goal is to increase the reportedly low typing performance of people with disabilities. Application scenarios are given in the Side Box.

To this end, we present a novel predictive model called Touch-WLM and first designs optimized using it. The model makes predictions on how users with given abilities enter text. It takes into account how they may regulate speed

and shift attention between the keyboard and text display. We model dyslexia, tremor, and memory dysfunction. In the rest of the paper, we first discuss the general method, then the modeling approach, and finally turn to results and discussion of future work.

2 ABILITY-BASED OPTIMIZATION

Our goal is to develop ability-based optimization for increasingly realistic and important interactive tasks. Although we here focus on text entry as the case, the approach offers a more generally applicable procedure. Five steps are necessary:

- 1) Definition of the design space
- 2) Definition of the objective function
- 3) Constructing a parameterizable generative model of user behavior.
- 4) Obtaining parameters to describe a given user/group
- 5) Using a combinatorial optimization method to search for design solutions.

Modeling To go beyond the limits of previous work, our goal is to use models that are 1) generative (generate step-by-step task performance), 2) parameterizable (parameters describing functional-level consequences of disabilities), and 3) rational (adapt their behaviors to constraints of the UI to make the most of the capabilities). In the case of touchscreen typing, such models can predict both task-level and keystroke-level behaviors, such as inter-key intervals and typing errors.

The upper bound of their performance can be estimated by making the model choose actions or behaviors that optimize its behavior (rational analysis) [11], [12]. In typing, a user valuing fast typing can opt to type rapidly. Because fast typing increases error rates, the user must compensate with frequent proofreading of the typed text. Because of the human visual system’s constraints, this takes time and

uses attention resources that are needed elsewhere in the typing task. The model we describe below identifies the best interaction strategy for each design the optimizer has generated.

Parameter acquisition Model parameters that describe individual abilities can be acquired with four methods. First, although individual differences have not received extensive scrutiny in cognitive science, several sources address parameters related to abilities, such as vision [17] and working memory [18]. Second, standard practices exist for empirically measuring abilities like working memory capacity, visual acuity, tremor, etc. Third, one can infer parameters from unconstrained (natural) user behavior by using machine learning methods [19]. Fourth, end-users or carers could try to estimate and express them interactively for example via dialogue or settings panel.

Optimization For design optimization, several methods are available, from precise methods to black-box optimization. In the work described below, we use exhaustive search.

Most steps in the process depend on expert input, such as designers providing suitable designs, psychologists supplying models of disabilities, and medical professionals specifying the disabilities of a user as parameters. It is also imaginable that a system infers the user’s disabilities during the use and adapts dynamically. How good the optimized designs are depends on the match between the design problem and the individual. For instance, if the key size is not an adjustable parameter, it is impossible to optimize for a person with tremor. Moreover, any functional simulation of a disability is necessarily an abstraction. While cognitive modeling has been shown to work well for healthy adults with significant practice, care should be taken to validate predictions when working with users with disabilities.

3 APPLICATION TO TEXT ENTRY

Here we describe how ability-based optimization can be applied to text entry on touchscreens. We stress that we are not suggesting new design ideas but demonstrating how ability-based optimization can be used to identify designs suitable for users with disabilities.

We first describe the design space and then the predictive model. The outputs of Touch-WLM are predictions of task performance and step-by-step actions of an individual entering text on a touchscreen device. We then describe how impairments – especially tremor, dyslexia, and memory deficits — can be incorporated in the model. In the next section, we report optimized designs obtained with this approach.

3.1 Design Space

To cover the design space of text entry methods, we analyzed several keyboard applications available for common smartphones. The design space assumes a five-inch smartphone device ($14.3 \times 7.1 \text{ cm}^2$). The design parameters, presented in Figure 1, cover decisions ranging from space allocated to elements to advanced support like word-prediction lists.

The full screen area is occupied by the keyboard, word-prediction list (WPL) (if present), and the text display area.

SIDEBOX: SCENARIOS FOR TEXT ENTRY

This paper focuses on Scenarios 1 and 2.

1. Julia endures ataxic cerebral palsy, a movement disorder causing tremor and coordination problems. An optimizer automatically adapts her keyboard to use larger buttons, grouping three letters per button. After the first few letters have been selected, she completes the word by selecting from a word-prediction list.

2. Alec is a high school student with diagnosed dyslexia. He uses an optimized layout that lets him select words from a word-prediction list and hence be certain that the words selected are entered correctly. Further, the layout’s spacing and the typed text’s font size have been set for ease of reading.

3. Roger is an elderly person experiencing the onset of Alzheimer’s disease. The typing application, optimized for memory disease, splits the typing task into smaller sections and allows Roger to focus on one subtask at a time.

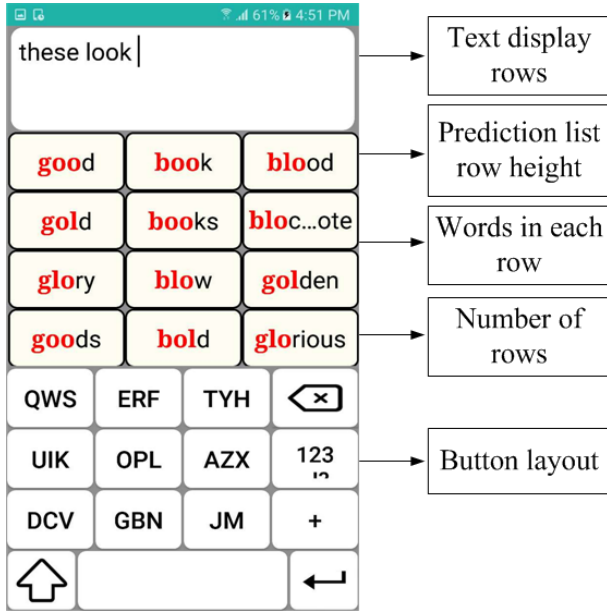


Fig. 1. Design parameters optimized for in the case study.

Text display rows controls the number of text rows. *Button layout* indicates whether the keyboard is ordinary Qwerty, with one letter per button, or a grouped layout, wherein multiple letters are assigned to a button. The grouped layout needs a WPL due to the ambiguity of a keypress. The WPL shows the most probable words, given the key-group buttons pressed and word frequencies (calculated using the CMU-SLM toolkit). The number of predictions depends on two parameters: *number of rows* and *words per row*. *Height of the prediction list* is an additional parameter.

3.2 Modeling Text Entry Performance

Touch-WLM is a model of text entry performance on touch-screen devices. It takes as input a set of parameter values describing the user, an interface design, and a set of sentences to be typed. It is a word-level model, where its task is to type words from the Enron Mobile corpus [20] as quickly possible with as few typing errors as possible. It outputs all keystroke-level actions, the number of typing errors and typing speed in WPM. To achieve this, it combines low-level sensorimotor actions with high-level strategic decisions.

An overview of the model structure is given in the Side Box. The low-level simulation predicts finger movements between individual keys. Using a model of visual search [21] (equations 1, 2, and 3), *Touch-WLM* predicts how long the user needs to search for a key, how long finger movement takes, and how accurate the movement is.

The high-level or strategic choices are 1) finger endpoint spread y , controlling the speed and accuracy of finger movement, and 2) proofreading frequency, given as number of letters l typed between error-checkings. Faster finger movements produce faster typing but also increase error probability. Errors need correcting, which takes time. Here, if the simulated user has tremor, a higher minimum finger endpoint spread value makes typing more error-prone, which suggests a strategy with frequent checking. While proofreading is time away from typing, overly long

intervals between checks increase the likelihood of having to correct a long string of letters. Determining the optimal strategy involves finding a typing-finger distribution y and proofreading frequency l such that words are typed as quickly as possible yet without errors in the final text. In addition, if a WPL is present, the model can search it for the word being typed. If the simulated user has dyslexia, both proofreading and reading the WPL will take longer. For this reason, a strategy wherein pointing is slower but more accurate decreases the frequency of checking.

SIDEBOX: TOUCH-WLM

Technically, *Touch-WLM* is stated as a control sequence:

- 1) Typing l letters by
 - a) attending and encoding the target key with eyes,
 - b) moving finger to key position and touching the key, calculating movement time from (1) with $y \in [y_{min}, y_{max}]$.
- 2) Proofreading typed text, and if there is a typing error,
 - a) moving finger to backspace button and touch it, calculating movement time from (1) with $y = y_{min}$,
 - b) repeating backspace press as many times as necessary,
 - c) repeating step 1, i.e., type the letters again.
- 3) Reading through the WPL by
 - a) attending words in the prediction list one at a time,
 - b) if the target is present, selecting it,
 - c) if not, typing until finishing the word.

3.3 Modeling Impairments

To account for individual abilities with *Touch-WLM*, it needs to be parametrized. We here describe parametrization for tremor, dyslexia, and memory dysfunction. They are necessarily functional simplifications of the complex underlying biological and cognitive phenomena. They try to capture some essential aspects as they affect text entry in particular.

3.3.1 Tremor

Tremor, defined as involuntary movement of a limb, is present to some extent in all humans. Excessive tremor is linked to, for example, essential tremor and Parkinson's disease.

To address tremor in text entry, we use a pointing model, which parametrizes the individual ability to control the speed and accuracy of finger [22]:

$$(y - y_0)^{1-m_\alpha} (x - x_0)^{m_\alpha} = m_k, \quad (1)$$

It predicts finger movement time x , given the standard deviation y of finger landing points (endpoint spread). Very

precise movements (small y) require more movement time (large x), whereas fast movements (small x) entail less precision (large y). Everyone has a unique speed–accuracy curve, dictated by m_k and m_α . On this curve, the individual can choose a point matching how he or she wishes to balance speed and accuracy of pointing. Human physiology sets hard constraints to maximal accuracy y_0 and speed x_0 . There are also individual-level limits to speed and accuracy, such that $x < x_{max}$ (maximal speed) and $y > y_{min}$ (minimum pointing spread).

In the context of this model, we define tremor as a large minimum endpoint spread y_{min} . For a healthy adult with no noticeable tremor, finger endpoint spread under maximal-accuracy conditions is, on average, 0.01 cm [23]. A person with essential tremor has an average tremor amplitude of 4.7 cm, and the figure for someone with Parkinson’s disease is 10.6 cm [23].

The model presented here deals with only a subset of tremor-related pointing problems, mainly of speed and accuracy. Premature and multiple touching [7] are not covered by our model. However, extensions are possible.

3.3.2 Dyslexia

The time that it takes to inspect text is longer for dyslexics than non-dyslexics [24]. In text entry, visual attention is divided between proofreading and guiding the finger on the software keyboard. If proofreading takes the user a long time, this inevitably leads to poorer touchscreen typing performance.

A dyslexic user’s performance can be captured with a reading model that parameterizes time for reading a word, given its frequency [25]:

$$T_e = E_K \cdot [-\log(f)] \cdot e^{e_k \cdot \epsilon}, \quad (2)$$

where f is the frequency of the word and ϵ is the visual distance of the target. Higher values for the parameter E_K increase total letter- and word-inspection times so can be used in simulating dyslexia.

Additionally, higher e_k values can be used to simulate poorer visual acuity, because they make the visual distance of the object have a greater impact on reading speed. The non-dyslexic’s value for E_K is set to 0.006 [25]; for a hypothetical dyslexic, who needs twice as long as a non-dyslexic to read the word, the value should be 0.012.

3.3.3 Memory Dysfunction

Memory functioning has a significant role in complex tasks like text entry. We model the role of memory—and that of memory dysfunction—in typing, by implementing a memory and expertise model. The model utilized by Jokinen et al. [21] features parameters for long-term memory retrieval time and learning speed:

$$T_i = F e^{-f B_i}, \quad (3)$$

which gives the time T_i to retrieve a memory entry i , given its activation B_i (calculated from how often the entry is used). Increasing F increases retrieval times, to a point where retrieval from long-term memory is extremely unreliable. High f models a situation wherein the user would require numerous instances of exposure before the memory

entry can be reliably retrieved. Further, the modeler can specify a baseline activation parameter B , a value added to or subtracted from each B_i for simulating the effect of memory dysfunction [18].

4 DESIGNS OPTIMIZED FOR IMPAIRED USERS

The results presented in this section were obtained using exhaustive search of the design space, evaluating the designs using Touch-WLM.

Its parameters were set for dyslexia and essential tremor or Parkinson’s by reference to literature. For the tremor case, y_{min} was set to correspond to about 2 cm finger endpoint spread [23]. For dyslexia, scaling parameter E_K for reading time in Equation 2 was doubled from the default 0.006, to 0.012, and proofing time was doubled accordingly [24], [26].

4.1 Tremor

Our optimized design increases the predicted typing speed of a person with tremor by 16%. It permits very low error rates.

When no tremor and the baseline design is assumed, the model predicts 15.7 WPM. However, assuming a user with tremor (y_{min} at 2 cm resting tremor [23]), using the baseline design, WPM drops to 1.9 with a very large error rate of 60% (see Figure 2). Figure 3 illustrates the tremor model using the baseline design, making typing errors, and then having to spend time correcting them. In practice, this user would be unable to type with this design.

The optimizer suggests to fix this by using a layout that groups three letters per button, as shown in Figure 4b. With this layout, the simulated user achieves a 16% improvement (to 2.2 WPM) and the error rate falls to 5%. While there is an improvement in speed, it should be noted that the final typing speed is still fairly slow. However, the error rate has dropped from 60% to an acceptable level, which enables the individual to type.

The optimized design allows a user with tremor hitting correct keys more often. This reduces the error rate

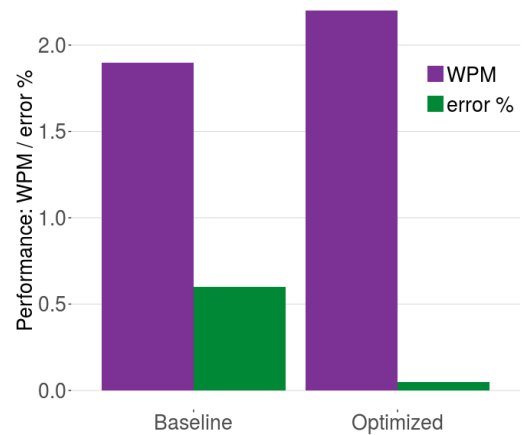


Fig. 2. Simulated typing speed (WPM) and error rate for a tremor model with a baseline Qwerty keyboard and an optimized keyboard using a grouped layout and word prediction (Fig. 4b). Typing performance is predicted to improve in terms of speed and in particular in terms of accuracy.

and thereby increases overall typing speed. Since most of the typing time is wasted on hitting the wrong key and backspacing, it is sensible to offer a grouped layout and a WPL with many options. This reduces the number of key presses needed.

4.1.1 Preliminary empirical evaluation

We conducted a preliminary empirical evaluation with two older adults with diagnosed essential tremor: a 69-year-old male (P1) and a female of age 67 (P2). They performed text-transcription tasks with two keyboards: baseline Qwerty (see Figure 4a) and the optimized layout (see Figure 4b). For the task, both were instructed to type a given sentence repeatedly for 10 minutes. This repetition ensured familiarity with both layouts, important since our target was to analyze the natural text-typing speed and accuracy [27].

The results, presented in Table 1, show clear decrease in error rate. However, in terms of typing speed, one participant was faster with the optimized layout than with the Qwerty baseline, while the other had trouble typing with the former.

For both participants, the typing speed observed was higher than the model predicted. The main explanation is that model parameters were not set to these individuals but based on literature. For example, users' actual finger motion was faster than assumed in the model. Adjusting the parameter for finger speed would allow matching the modeled typing speed for any given participant. Moreover, one could make the better model match the magnitude of the error rate by adjusting the parameter for how much tremor it simulates in the finger. Thus, the observed performance is in the capacity of the model to simulate. In future, inverse modeling methods could be used to infer parameters for an individual from behavioral observations [19].

The main result is that the optimized design did reduce error rates significantly, although for one user this happened at the expense of speed. We stress that the proposed design is not claimed to be the best solution for a person with

tremor. The benefit of the approach is that, as Figure 3 indicates, we can explore designs that account for a disability at its source.

TABLE 1

Results from a preliminary evaluation with two users with tremor: The optimized design reduced errors to an acceptable level (6% or below). For P1, speed was compromised, however.

Participant	Text Entry Rate (WPM)		Error Rate (%)	
	Baseline	Optimized	Baseline	Optimized
Model	1.9	2.2	60.0	0.1
P1	8.0	2.8	21.6	6.3
P2	6.6	7.5	7.2	3.2

4.2 Dyslexia

The optimized design is predicted to increase the typing speed of a person with dyslexia by 11%. The model of a healthy user using the baseline design achieves 15.7 WPM. Doubling the reading time parameter in the model of a dyslexic user decreased typing speed to 9.6 WPM. However, with the optimized layout, shown in Figure 4c, typing performance rises, by 11%, to 10.7 WPM.

The results can be investigated by examining the model-generated output in Figure 3. There, the yellow bars for proofing are longer—this particular subtask is costly for the user. Therefore, the optimal typing strategy is to locate the typed word in the WPL and thereby confirm that the word entered contains no errors and no checking is required. However, because the WPL does not necessarily contain the word needed (as occurs with infrequent words), the model predicts it to be faster to have a shorter WPL, for a lower read-through cost. The increase in reading-time parameter E_K influences the search for the correct word in the WPL, but, by relying on the list, the model can adopt a strategy wherein the even more costly proofreading is infrequent. However, too large a WPL would increase word-search times, so the optimizer suggests two rows of words as the best tradeoff.

5 DISCUSSION

It is important to study design methods that may allow users with disabilities to get more from their computers-disabilities. Without designs that better support individual abilities, people with even slight impairments may be hindered in their efforts to participate in an increasingly computerized society. While touchscreen devices have become the prime terminal for personal computing, their design has been particularly unfavorable for people with sensorimotor and cognitive impairments.

The benefit of ability-based optimization is that designs can be obtained with very little input. Only the parameters describing the abilities are needed after the design task and the objective function are defined. In the future, ability-based optimization should be extended beyond the examples presented here. For instance, in the Alzheimer's case, described above, the system could utilize models of working memory and long-term memory in addition to vision and motor control. Optimizers could also be calibrated online

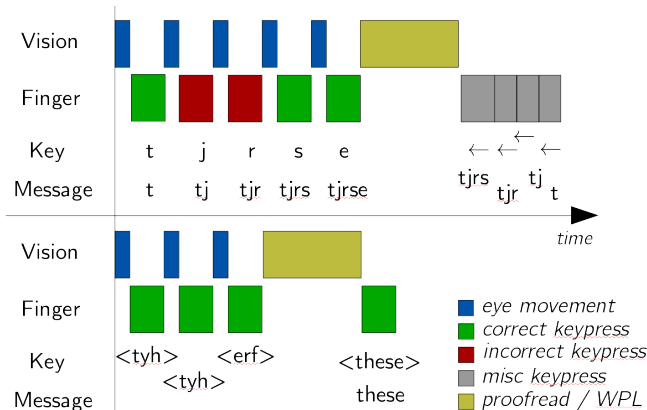


Fig. 3. Simulated eye and finger movements in typing of a word. **Top:** A tremor-simulating model using the baseline keyboard. It makes many typing errors. The model detects them after proofing, and they must be corrected. **Bottom:** The same model using an optimized design with large grouped keys and a word-prediction list (WPL) (Fig. 4b). In the time the model has typed the word in the lower condition, it types only one correct letter in the upper one. The bar widths are approximate for visualization purposes.

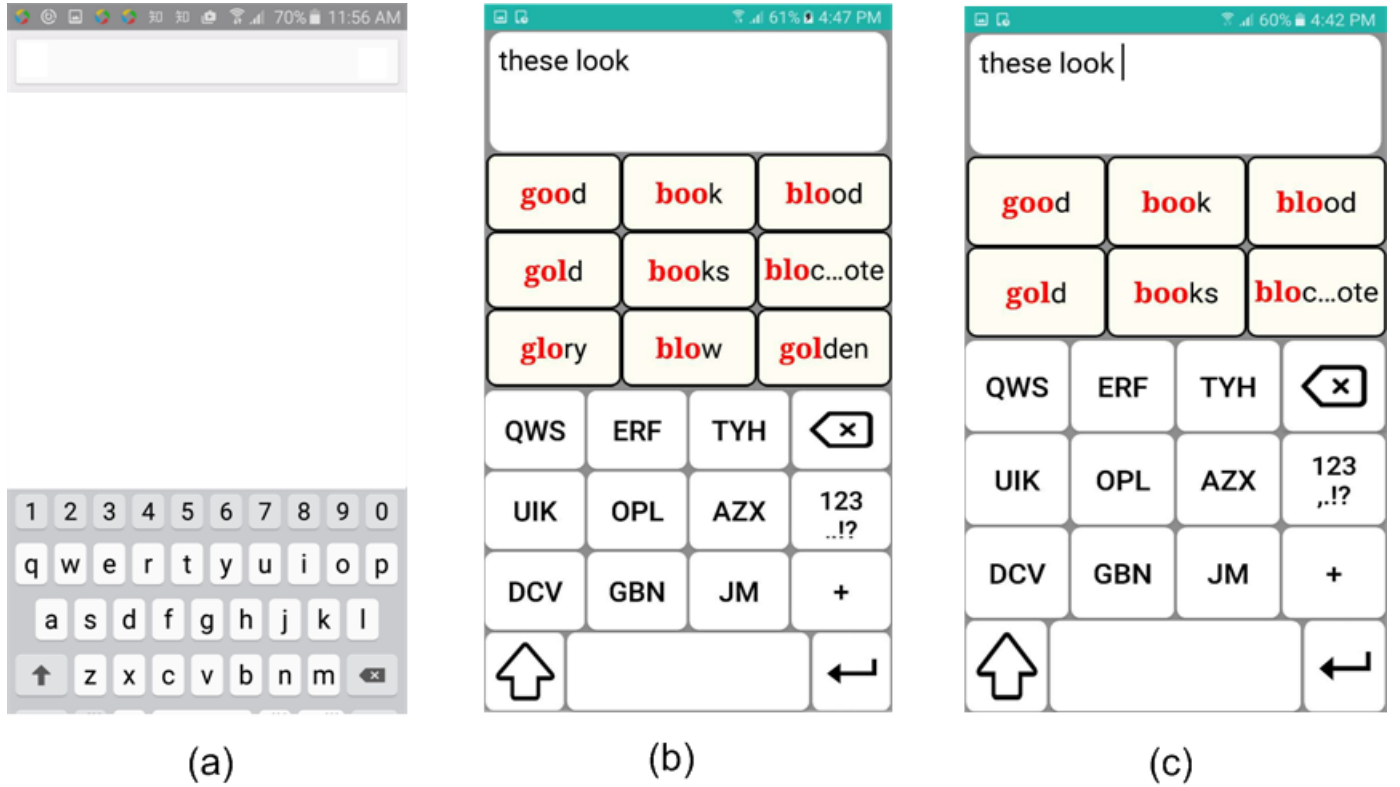


Fig. 4. (a) Baseline and optimized designs for (b) people with essential tremor or Parkinson's and (c) users with reading disabilities.

based on behavioral data. Such a system could be useful when disabilities worsen over time or change abruptly during use.

Previous work on ability-based optimization has been limited to motor performance and addressed other abilities via heuristics, if at all. Realistic models of individual capabilities must be embraced if we are to address increasingly important and complex user tasks. This paper has shown that individual-specific capabilities can be described in a theoretically plausible manner for predictive models familiar in HCI research. While more empirical work is needed to evaluate the results, the first evidence acquired in this paper is promising. While first evidence was found for the design targeting tremor, more work is needed to empirically test the design for dyslexics.

Perhaps the most critical challenge for the future is to formally understand disabilities. We must define optimization approaches that tackle the toughest challenges disabled people face in interaction. Their existing aids, peripherals, and prostheses should be characterized and included in the design spaces, for making the most of known-good solutions. At the same time, we need to work with clinicians and neuroscientists to produce increasingly plausible models of their disabilities. Optimized designs should be subjected to rigorous empirical testing to avoid mischaracterizing them.

6 ACKNOWLEDGMENTS

This work has received funding from the joint JST–AoF project “User Interface Design for the Ageing Population” (AoF grant 291556) as an activity of FY2014 Strategic International Collaborative Research Program (SICORP), and

from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement 637991). Olli Savisaari, Dr. Hiroshi Miyamoto, Ayumu Ono and Jingxin Liu helped with the studies.

REFERENCES

- [1] S. N. Duff, C. B. Irwin, J. L. Skye, M. E. Sesto, and D. A. Wiegmann, “The effect of disability and approach on touch screen performance during a number entry task,” in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 54, no. 6. SAGE Publications Sage CA: Los Angeles, CA, 2010, pp. 566–570.
- [2] T. J. V. Guerreiro, H. Nicolau, J. Jorge, and D. Gonçalves, “Assessing mobile touch interfaces for tetraplegics,” in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*. ACM, 2010, pp. 31–34.
- [3] Y. Zhong, A. Weber, C. Burkhardt, P. Weaver, and J. P. Bigham, “Enhancing Android accessibility for users with hand tremor by reducing fine pointing and steady tapping,” in *Proceedings of the 12th Web for All Conference*. ACM, 2015, p. 29.
- [4] S. K. Kane, J. P. Bigham, and J. Wobbrock, “Fully accessible touch screens for the blind and visually impaired,” *University of Washington*, 2011.
- [5] J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich, “Ability-based design: Concept, principles and examples,” *ACM Transactions on Accessible Computing (TACCESS)*, vol. 3, no. 3, p. 9, 2011.
- [6] C. Wacharamanotham, J. Hurtmanns, A. Mertens, M. Kronenbuerger, C. Schlick, and J. Borchers, “Evaluating swabbing: a touchscreen input method for elderly users with tremor,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 623–626.
- [7] S. Trewin, C. Swart, and D. Pettick, “Physical accessibility of touchscreen smartphones,” in *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 2013, p. 19.

- [8] M. Naftali and L. Findlater, "Accessibility in context: understanding the truly mobile experience of smartphone users with motor impairments," in *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*. ACM, 2014, pp. 209–216.
- [9] A. Oulasvirta, "User interface design with combinatorial optimization," *IEEE Computer*, 2017.
- [10] K. Z. Gajos, J. O. Wobbrock, and D. S. Weld, "Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 2008, pp. 1257–1266.
- [11] X. Chen, G. Bailly, D. P. Brumby, A. Oulasvirta, and A. Howes, "The emergence of interactive behavior: A model of rational menu search," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 4217–4226.
- [12] S. J. Payne and A. Howes, "Adaptive interaction: A utility maximization approach to understanding human interaction with technology," *Synthesis Lectures on Human-Centered Informatics*, vol. 6, no. 1, pp. 1–111, 2013.
- [13] L. Findlater and J. Wobbrock, "Personalized input: improving ten-finger touchscreen typing through automatic adaptation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 815–824.
- [14] Y.-L. Lin, M.-C. Chen, Y.-P. Wu, Y.-M. Yeh, and H.-P. Wang, "A flexible on-screen keyboard: Dynamically adapting for individuals needs," in *International Conference on Universal Access in Human-Computer Interaction*. Springer, 2007, pp. 371–379.
- [15] S. Trewin, "Automating accessibility: the dynamic keyboard," in *ACM SIGACCESS Accessibility and Computing*, no. 77-78. ACM, 2004, pp. 71–78.
- [16] S. M. Lafi and S. K. B. Hock, "An adaptive text entry method based on single-key minimal scan matrix for people with severe motor disabilities," *International Journal of Scientific & Engineering Research*, vol. 3, no. 8, pp. 1–5, 2012.
- [17] T. S. Jastrzebski and N. Charness, "The Model Human Processor and the older adult: parameter estimation and validation within a mobile phone task," *Journal of experimental psychology: Applied*, vol. 13, no. 4, pp. 224–248, 2007.
- [18] A. Serna, H. Pigot, and V. Rialle, "Modeling the progression of Alzheimer's disease for cognitive assistance in smart homes," *User Modelling and User-Adapted Interaction*, vol. 17, no. 4, pp. 415–438, 2007.
- [19] A. Kangasrääsiö, K. Athukorala, A. Howes, J. Corander, S. Kaski, and A. Oulasvirta, "Inverse modeling of complex interactive behavior with ABC," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2017.
- [20] P. O. Kristensson and K. Vertanen, "Performance comparisons of phrase sets and presentation styles for text entry evaluations," in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. ACM, 2012, pp. 29–32.
- [21] J. Jokinen, S. Sarcara, A. Oulasvirta, C. Silpasuwanchai, Z. Wang, and X. Ren, "Learning visual search on new keyboard layout," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2017.
- [22] Y. Guiard and O. Rioul, "A mathematical description of the speed/accuracy trade-off of aimed movement," in *Proceedings of the 2015 British HCI Conference*. ACM, 2015, pp. 91–100.
- [23] M. M. Sturman, D. E. Vaillancourt, and D. M. Corcos, "Effects of aging on the regularity of physiological tremor," *Journal of neurophysiology*, vol. 93, no. 6, pp. 3064–3074, 2005.
- [24] J. Hyona and R. K. Olson, "Eye fixation patterns among dyslexic and normal readers: Effects of word length and word frequency," *Journal of Experimental Psychology. Learning, Memory & Cognition*, no. 6, p. 1430.
- [25] D. D. Salvucci, "An integrated model of eye movements and visual encoding," *Cognitive Systems Research*, vol. 1, no. 4, pp. 201–220, 2001.
- [26] S. Sarcara, J. Jokinen, A. Oulasvirta, C. Silpasuwanchai, Z. Wang, and X. Ren, "Towards ability-based optimization for aging users," in *Proc. IxAP 2016*. ACM, 2016, pp. 77–86.
- [27] X. Bi and S. Zhai, "Ijqwerty: What difference does one key change make? gesture typing keyboard optimization bounded by one key position change from qwerty," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 49–58.



SAYAN SARCAR is a research associate at Kochi University of Technology. His research interests include ageing and accessibility, input and interaction techniques on smartphones, and multimodal text input systems. He can be contacted at sayan.sarcara@kochi-tech.ac.jp.



JUSSI JOKINEN is a postdoctoral fellow at Aalto University. Among his research interests are cognitive science, HCI, and modeling. He can be contacted at jussi.jokinen@aalto.fi.



ANTTI OULASVIRTA is an associate professor at Aalto University. His research interests include modeling and computational design of human-computer interaction. Contact him at antti.oulasvirta@aalto.fi.



ZHENXIN WANG is an assistant professor at Kochi University of Technology. His research interests include machine learning and HCI. He can be reached at wang.zhenxin@kochi-tech.ac.jp.



CHAKLAM SILPASUWANCHAI is an assistant professor at Asian Institute of Technology. Among his research interests are digital games, physiological computing, brain-computer interfaces. Contact him at chaklam.silpasuwanchai@gmail.com.



XIANGSHI REN is Professor and director of the Center for Human-Engaged Computing at Kochi University of Technology. His research interests encompass all aspects of human-computer interaction. Contact him at xsren@acm.org.