Aalto University

OPEN ACCESS

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

Li, Zhi; Ko, Yu-Jung; Putkonen, Aini; Feiz, Shirin; Ashok, Vikas; Ramakrishnan, I. V.; Oulasvirta, Antti; Bi, Xiaojun

Modeling Touch-based Menu Selection Performance of Blind Users via Reinforcement Learning

Published in:
Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)

DOI:
10.1145/3544548.3580640

Published: 19/04/2023

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

Please cite the original version:
Li, Z., Ko, Y.-J., Putkonen, A., Feiz, S., Ashok, V., Ramakrishnan, I. V., Oulasvirta, A., & Bi, X. (2023). Modeling Touch-based Menu Selection Performance of Blind Users via Reinforcement Learning. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23) Article 357 ACM. https://doi.org/10.1145/3544548.3580640

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Modeling Touch-based Menu Selection Performance of Blind Users via Reinforcement Learning

Zhi Li
zhili3@cs.stonybrook.edu
Department of Computer Science,
Stony Brook University
New York, USA

Yu-Jung Ko
yujko@cs.stonybrook.edu
Department of Computer Science,
Stony Brook University
New York, USA

Aini Putkonen
aini.putkonen@aalto.fi
Department of Information and
Communications Engineering,
Aalto University
Helsinki, Finland

Shirin Feiz
sfeizdisfani@cs.stonybrook.edu
Department of Computer Science,
Stony Brook University
New York, USA

Vikas Ashok
vganjigu@odu.edu
Department of Computer Science,
Old Dominion University
Virginia, USA

IV Ramakrishnan
ram@cs.stonybrook.edu
Department of Computer Science,
Stony Brook University
New York, USA

Antti Oulasvirta
antti.oulasvirta@aalto.fi
Department of Information and
Communications Engineering,
Aalto University
Helsinki, Finland

Xiaojun Bi
xiaojun@cs.stonybrook.edu
Department of Computer Science,
Stony Brook University
New York, USA

ABSTRACT

Although menu selection has been extensively studied in HCI, most existing studies have focused on sighted users, leaving blind users' menu selection under-studied. In this paper, we propose a computational model that can simulate blind users' menu selection performance and strategies, including the way they use techniques like swiping, gliding, and direct touch. We assume that selection behavior emerges as an adaptation to the user's memory of item positions based on experience and feedback from the screen reader. A key aspect of our model is a model of long-term memory, predicting how a user recalls and forgets item position based on previous menu selections. We compare simulation results predicted by our model against data obtained in an empirical study with ten blind users. The model correctly simulated the effect of the menu length and menu arrangement on selection time, the action composition, and the menu selection strategy of the users.

CCS CONCEPTS

• **Human-centered computing → Accessibility theory, concepts and paradigms**; **Empirical studies in accessibility**; **HCI theory, concepts and models**; **User models**.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI '23, April 23–28, 2023, Hamburg, Germany
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9421-5/23/04…$15.00
https://doi.org/10.1145/3544548.3580640

KEYWORDS

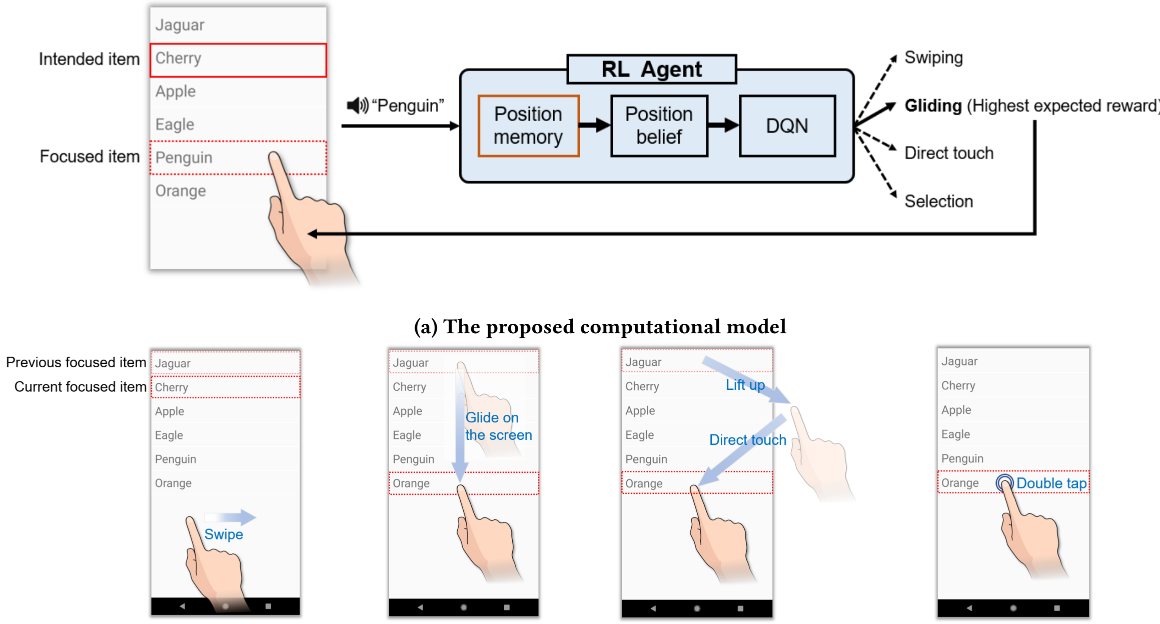
accessibility, menu selection, computational rationality, boundedly optimal control, deep reinforcement learning

ACM Reference Format:
Zhi Li, Yu-Jung Ko, Aini Putkonen, Shirin Feiz, Vikas Ashok, IV Ramakrishnan, Antti Oulasvirta, and Xiaojun Bi. 2023. Modeling Touch-based Menu Selection Performance of Blind Users via Reinforcement Learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3544548.3580640

1 INTRODUCTION

Smartphones have become the go-to device for many blind users to access information and communicate with others [1, 57]. Menu selection (i.e., selecting items on a menu) is one of the most basic and common tasks that blind users perform on smartphones [30, 40]. A computational model that can simulate blind users' menu selection behavior would be a valuable tool for designing, adapting, and evaluating accessible user interfaces (e.g. SUPPLE [20]). For example, by simulating user's menu selection behavior, a model can serve as a basis for testing a menu design even before releasing it to end users. Similarly, using a computational model in an inverse manner (i.e., fitting a model to empirical data) allows inferring user-specific characteristics that can be used to adapt a menu system to individual preferences [38]. It is critical that the needs of different groups, like those of visually impaired users, are taken into account when building menu systems.

Although menu selection may appear rather easy for sighted users, it involves rather a complex behavior for blind users. To select a menu item, a blind user first navigates to the item with the auditory feedback [63] from a screen reader (e.g., VoiceOver in



(a) The proposed computational model

(b) Swiping action (c) Gliding action (d) Direct touch action (e) Selection action

Figure 1: We develop a computational model for predicting both performance and strategies of non-sighted users using screen readers for menu selection. A reinforcement learning (RL) agent simulates how blind users perform menu selection task. The agent first partially observes the menu selection environment based on auditory feedback. It then utilizes a memory model, which considers long-term interaction history, to maintain the menu item position and updates belief of menu item position based on the memory. After that, the agent uses a Deep-Q Network to map the position belief to the optimal action (one of the swiping, gliding, direct touch and selection actions) that leads to the highest expected reward, i.e. shortest selection time.

iPhone [4] or TalkBack in Android [25]), and then selects the item by double tapping (Fig. 1e). The current smartphone (e.g., iPhone or Android) typically provides the following three navigation actions and each has its own strengths and weaknesses:

- Swiping (Fig. 1b): a quick finger flick on screen which will move the selection focus by one item. It is easy to perform but it only supports sequentially navigating the menu.
- Gliding (Fig. 1c): gliding the input finger on the screen to explore. It allows a user to check the item the finger passes through during exploration, but the finger has to stay on the screen during the entire process.
- Direct touch (Fig. 1d): directly moving the finger over the air to land on an item in the menu. It enables quick access to a particular item but controlling the landing position of the finger can be difficult without visual feedback.

Given a menu design, a blind user may choose any of the three navigation actions, or combination of them to search for and then select the menu item. Their selection strategy depends on a complex way on their familiarity with the menu design, memory strength for item positions, capability of executing actions, and their habits. For example, some menu layouts contain information that may help memorize and locate menu items such as the alphabetic arrangement where menu items are ordered alphabetically and grouped arrangement where similar items are grouped together [63]. A blind user may develop different memory strength of items in different layouts, which affect their interaction behaviors. Having a computational model that can simulate how a blind user selects menu items on different layouts would not only advance our understanding on user's interaction behaviors on this important interaction task but also serve as a useful tool in menu design.

In this paper, we research and develop a computational model that simulates how blind users select items in a linear menu with finger touch. Our model is based on the assumption that the blind user's menu selection behavior is *boundedly optimal* [22, 29, 43, 52, 54, 55]. Users try to pick the best interaction strategy but they are limited by their capabilities. More technically, users are assumed to try to pick the best action based on a reward estimate that is jointly shaped by their preferences and bounds, where these bounds arise from cognition, physiological capabilities, and the designed interaction environment [52]. More specifically, we assume that a blind user chooses the best action (e.g., swiping, gliding, or direct touch) in a menu selection task (control problem) to minimize selection time (preference) in light of their perceptual limits and memory (bounds) on a vertical menu (interaction environment). We also assume that blind users' observations are limited to auditory feedback only [63], and that they could learn the positions of the items with sufficient iterations of the task [26, 28].

In the rest of the paper, we develop and evaluate an RL (reinforcement learning) -based predictive model for blind users' menu selection behavior (Fig. 1). We formulate the menu selection as a *stochastic sequential decision problem* [8] where the agent (the user) performs one action (swiping, gliding, direct touch and selection) at each time step and finally selects the intended menu item. We particularly model the menu selection process as a Partially Observable Markov Decision Process (POMDP) [36, 60] as the agent (simulating a blind user) only has partial access to its environment via auditory, rather than visual feedback. Based on so-far experienced feedbacks, the agent holds a belief about the position of each menu item, which comes from the memory model. The memory model, rooted on prior HCI work, probabilistically describes how the blind user memorizes and recalls the menu item position, and updates upon each action taken. Based on the belief, the agent chooses from available actions that maximizes the total expected rewards: negative value of the time cost. Under this setup, we train an optimal policy (i.e., menu selection strategy) for selecting a menu item in a given layout via Deep Q-Network (DQN) [49]. The obtained optimal policy mimics blind users' menu selection behaviors. To sum up, the technical contribution of this paper is an extension of generative POMDP-based user modeling to the case of menu selection by blind users.

We evaluate the model by comparing the simulated selection time and action composition to empirical results in a user study. Evaluation results show that the proposed model can successfully simulate the blind users' menu selection behavior. Specifically, the model can simulate the selection time with a mean absolute error 0.39 s (mean absolute percentage error: 6.71%) as well as simulate the learning effect during the menu selection, i.e. the selection time decreases as the user study proceeds. It can simulate the effect of menu length on the selection time, i.e. 10-item menus have longer selection time than 6-item menus. For the 10-item menu, it can also simulate the effect of menu arrangement (alphabetic, grouped, random) on the selection time, i.e. random menu has the longest selection time and grouped menu has the shortest selection time. Besides the selection time, our model can simulate action composition that were used for the menu selection task at a small mean absolute error, which is 3.89%, and it can also predict the menu selection strategies used by the users. At the end of the paper, we discuss potential applications and needs for further development of the model.

2 RELATED WORK

This work is positioned at the intersection of three research areas: 1) accessibility for touchscreens, 2) models of menu selection, and 3) modeling human interactive behavior with reinforcement learning.

2.1 Accessibility for Touchscreens

Touch screens are widely used in different devices, including smart phones, tablets, laptops, GPS devices and smart watches. To make these devices accessible for users who are visually impaired, assistive technologies such as screen readers, voice control and captions are widely available across different operating systems [4, 24]. Previous research has addressed the need to develop these features by exploring various alternative interaction techniques. Following the increase in prevalence of touch screen devices in the 2000s, Kane et al. [37] presented the Slide Rule: a multi-touch interaction technique using audio output only and different gestures as inputs. Other similar techniques relying on different gestures as input modalities include the MessageEase [51] (combining sliding and tapping) and NavTouch [27] (using directional gestures).

Despite the improvements in the understanding of interaction techniques for visually impaired users, a problem remains that in order to access features on a device visually impaired users need to navigate through, sometimes, complex user interfaces. For instance, Khan and Khusro [39] addressed this issue by presenting a user-adaptive framework for generating accessible user interfaces, taking advantage of information related to the context, screen layout and interaction pattern of the user. To this end, user modeling can help improve accessibility of different layouts. To the best of our knowledge, user models capturing interaction behavior of users who are blind in realistic applications is less explored in the literature.

2.2 Modeling Menu Selection

Modeling target selection behavior on menus and estimation of selection time with sighted users have been well-studied [2, 5, 6, 13, 15]. Fitts' law [19, 48], one of the well-known pointing model, is suitable to describe the behavior of target selection with gliding. However, it assumes that the user is fully aware of the position of the target before gliding. It was not designed to model people with vision impairment. There were some existing works that tried to model the target selection time of blind users based on Fitts' Law [16, 17]. Ko et al. [41] presented a mixture model for gliding strategies blind users employ when interacting with touchscreens. However, the more generic problem of how blind users make decisions on menu selection strategies is still an open research problem.

Some works viewed the menu selection task as a stochastic sequential decision problem and used reinforcement learning to estimate user's behavior [13, 61]. Chen et al. [13] modeled the menu selection behavior of sighted people as a Markov Decision Process and trained the policy with a model-free algorithm. Todi et al. [61] proposed a model-free based reinforcement learning method to select the optimal adaptation of the layout given the specified design. However, as far as we know, little existing work modeled how blind users perform menu selection task given different navigation methods: gliding, swiping, and direct touch. We follow the reinforcement learning routine to design an adaptive model for menu selection behavior of blind users.

2.3 Modeling Interactive Behavior with Reinforcement Learning

During the last ten years, there has been a surge in research applying reinforcement learning (RL) to model human interactive behavior. This line of research is usually based on the theory of computational rationality [22, 29, 43, 52, 54, 55]. A core assumption is that the user makes their behavioral choices governed by an attempt to optimize performance of interaction which is optimally bounded by the user's ability and the environment, known as *bounded optimal* control [52]. The interaction problem is usually modeled as a Partially Observable Markov Decision Process (POMDP) [36, 60]. The RL is a solver to the POMDP estimating users' behavior with bounded optimality. This theory was successfully applied in recent research works, e.g. menu selection [13, 38], visual search [14, 35, 62], multitasking [10, 33, 34], typing [32, 56], pointing [7, 11], decision making [42, 53] and drawing [58].

In the only RL-based model of menu selection behavior so far, Chen et al. [13] modelled menu selection strategies of sighted people as optimal adaptation to perceptual bounds, assuming that visual primitives – like shape and word content – can be sampled at different rates and accuracy by the visual system depending on their eccentricity. They showed that the search strategy depends, in complex way, on menu length and its organization. This paper presents a model for menu selection behavior of blind users, which assumes that recognition (of auditory feedback) and recall (of item positions) play a crucial role in determining a search policy.

3 MODELING MENU SELECTION AS BOUNDEDLY OPTIMAL CONTROL

We first formally describe the problem of selecting one item in a vertical linear menu with touch input, using auditory feedback only, as follows:

> Given a menu with K items, denoted by $\mathbf{M} = \{m_1, m_2, \cdots, m_K\}$, and a target menu item m_i, select the item m_i from $\mathbf{M}$ as quickly and accurately as possible using swiping (Fig. 1b), gliding (Fig. 1c), direct touch (Fig. 1d) and selection (Fig. 1e) actions.

We assume that the blind user chooses actions from (1) swiping, (2) gliding, (3) direct touch, and (4) double tapping to confirm the selection, to optimize the menu selection time under the constraint of the memory about the menu layout and the uncertainty presents in actions, especially in direct touch.

We further assume that menu selection is a sequential decision problem [8] where the agent (blind user) performs one action at a time to select the intended menu item and earlier actions influence how quickly a user can find the target item. Since the blind user only use auditory feedback, they have an incomplete access to the item positions on the layout. Hence, we model the menu selection problem as a Partially Observable Markov Decision Process (POMDP) [36, 60]. The agent maintains a memory of item positions and updates a belief of the position, an estimation of the state, at every time step. It makes a decision of which action to use based on the belief. We use the Deep-Q Network (DQN) [49] to learn an optimal policy, to find an optimal sequence of actions to reach a menu item. We assume this optimal policy would mimic the blind user's menu selection behavior.

In the following sections, we first formulate the menu selection as a POMDP and then introduce models describing users' memory of item positions and uncertainty in actions (especially in direct touch).

3.1 Modeling Menu Selection as Partially Observable Markov Decision Process

We model how a blind user selects a menu item as a POMDP (Fig. 2). Below is the model setup:

State (S): At each time step t, the environment is in a state $s_t = \{s_{\text{target}}, s_{\text{finger}}, s_{\text{focus}}\}$, where s_{target} is the position of the target menu item, s_{finger} is the finger position in the menu, and s_{focus} is the focused menu item position in the menu.

Observation (O): At every time step, the agent cannot directly observe the target item position s_{target}. We assume that the agent (simulating the blind user) can estimate the s_{finger} and s_{focus} by estimating the relative position of finger on the phone-screen.The rationale is that as the user holds the phone with one hand, the user will likely be able to sense the phone frame with the holding hand and thus estimate the finger position relative to the phone.

Belief update. As the agent does not observe s_{target} directly, it maintains a position memory (introduced in Section 3.2). Instead of assuming a Kalman filter as a state estimator [12], we assume that the position memory should consider the time of visits and amount of visits to menu items. When the agent makes a new observation, it adds a visit of the menu item at s_{focus} to the position memory. The agent then updates the position memory following Section 3.2. It represents the belief of s_{target} by a probability vector $\mathbf{p_i}$, indicating the probability that the target menu item is at each position in the menu layout, based on the position memory.

Action (A): After making an observation, the agent guesses a target position based on the belief (by sampling a position in the menu based on $\mathbf{p_i}$). It then takes an action a_t to move toward the sampled target position or select the current focused item. The available actions are:

- Swiping (Fig. 1b): Moving the selection focus by one item towards the sampled target position.
- Gliding (Fig. 1c): Gliding the finger to the sampled position. In one gliding action, we assumed that the user would glide continuously to the intended position and would not pause in the middle of the gliding. If the user paused in the middle of gliding to perceive an item, and then resumed gliding after obtaining information about the item, it will be modeled as two gliding actions: one before, and one after the pause.
- Direct touch (Fig. 1d): Directly touching the sampled target position. Due to uncertainty [44], the real landing position is modeled by a Gaussian distribution (Section 3.3).
- Selection (Fig. 1e): Selecting the focused menu item with double tapping.

At the end of swiping, gliding and direct touch, if s_{focus} reaches the sampled target position, but the item is not the target item (based on the auditory feedback), the agent samples another different target position based on $\mathbf{p_i}$.

After taking an action, the environment enters a new state. For gliding and direct touch, we assume the s_{focus} is the same as s_{finger} because both actions will move the input focus to the item underneath the finger. For swiping and selection actions, we assume that s_{finger} stays unchanged. In other words we assume a user swiped and double tapped on the similar region of the screen. Therefore, after these two actions, the s_{focus} and s_{finger} could be different. For example, s_{finger} could be at the top of the screen (i.e., the user always swiped at the top of the screen), and s_{focus} could be at an item in the center of the screen.

Reward (R): The agent receives an immediate reward from the environment once it takes an action. We define the reward of an action as the negative value of its time cost. We estimate the time cost MT based on empirical findings in a menu selection user study (see Section 5.1 and Table 2). In addition, for the selection action,

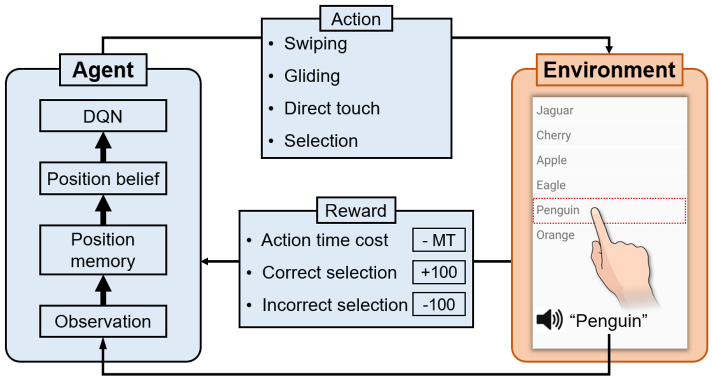


Figure 2: The reinforcement learning framework for the menu selection problem. We model the time cost MT for each action in Section 5.1 based on empirical results from a menu selection user study.

if the agent selects the correct menu item, there is an extra +100 reward, otherwise -100 reward. The ±100 reward values provide enough incentives to guide the agent for successful selections and to prevent it from attempting the selection action all the time.

Discount Rate $0 \leq \lambda < 1$***:*** The agent receives a scalar reward after taking each action. By maximizing the expected total reward $E[\sum_{t=0}^{\infty} \lambda^{\tau} r(s_t, a_t)]$, the optimal policy can be found.

Based on the above formulation, at each time step, we represent the belief of the agent by a vector $[sim, s_{\text{finger}}, s_{\text{focus}}, \mathbf{p_i}]$, where sim is the similarity between the current focused item and the target item, i.e. $sim = 1$ if they are the same, otherwise $sim = 0$. We normalize the s_{finger} and the s_{focus} to [0, 1] range for better training performance based on the menu length. As the state space is continuous and the action is discrete, we use a model-free algorithm DQN [49] to learn the optimal policy. We choose DQN because it can handle well the reinforcement learning problems where the state space is continuous and the action space is discrete [46].

3.2 Memory Model

As the blind users' menu selection is guided by the memory of menu item positions [26, 28], it is important to develop a model describing the memory strength of item position based on visiting history. To this end, we propose a Gaussian position memory model.

Given the long-term memory of interaction, we assume that the recalled position of an item m_i follows a Gaussian distribution [5], i.e. $y_i \sim \mathcal{N}(x_i, \sigma_i)$, where x_i is the actual position of the item and σ_i is a standard deviation related to the total memory $B(m_i)$ that the user has about the item. We calculate σ_i as follows

$$\sigma_i = \frac{\sigma_{PM}}{1 + B(m_i)}, \tag{1}$$

where σ_{PM} is a parameter representing the initial standard deviation. We use the learning component from [61] to compute $B(m_i)$, which considers all the visiting history to m_i:

$$B(m_i) = \sum_{j=1}^{n} (t - t_i^j)^{-\rho} \tag{2}$$

where t is the current time, t_i^j is the time of the j-th visit to the item, and ρ is a decay parameter and we also set it as 0.5. With this formulation, the recalled item position becomes more accurate with every visit as $B(m_i)$ increases. That is, the more a menu item has been visited, and the visit is more recent, it is more likely for the user to accurately recall the position of the item.

We use the position memory in the proposed model as follows. We represent the probability of the recalled position of m_i at each position in the menu as $\mathbf{p_i} = \{p_i^1, p_i^2, \cdots, p_i^K\}$, where K is the menu length, p_i^k means the normalized probability that m_i is the k-th item in the menu layout based on the Gaussian distribution and $\sum_k p_i^k = 1$.

Menu Arrangement. We also assume that some menu arrangements could help the blind user memorize item positions [59]. For example, for alphabetic arrangement, if the current item starts with letter "C" and the target menu item starts with letter "K", the user may determine the target menu item is below the current item. Similarly, for the grouped arrangement, the user may determine in which group the menu item is. Based on the menu arrangement, we apply a mask vector $mask = \{mk_1, mk_2, \cdots, mk_K\}$ to the $\mathbf{p_i}$, where the mask mk_i is set to 1 if it is a possible position, otherwise 0. Then the updated vector will be $\mathbf{p_i} = \{Z \times mk_1 \times p_i^1, Z \times mk_2 \times p_i^2, \cdots, Z \times mk_K \times p_i^K\}$, where Z is a normalization term.

3.3 Direct Touch Uncertainty Model

The direct touch action – directly moving the finger over the air to land on an item in the menu, is an action that could result in uncertainty due to the lack of visual feedback and uncertainty in finger touch [44]. We propose an action uncertainty model to describe the uncertainty of the menu item that the finger lands on. Specifically, we assume the real landing item can be modeled by a Gaussian distribution as it is usually a good candidate as the least-informative default [31]. The mean of the distribution is the position of the intended item, and the standard deviation is controlled by a parameter σ_{DTU}.

4 MENU SELECTION USER STUDY

We conducted a user study to understand how blind users perform menu selection in linear menus. The purpose of the study is two-fold. First, we aimed to empirically determine the parameter values for the proposed model (e.g., duration for a swiping action). These parameters are crucial for the model to generate interaction behaviors via reinforcement learning. Second, we can evaluate the performance of our model by comparing the generated with the observed interaction behaviors, and by examining whether the model can predict the menu selection time and actions from real users.

4.1 Participants and Apparatus

We recruited 10 legally blind participants (4 females, 6 males) who age between 34 and 60 and are unable to perceive any visual feedback useful to the menu selection task. Table 1 shows the demographic information of the participants in the user study. The study was IRB-approved and all the participants participated under informed consent. They were recruited through a non-profit organization that provides rehabilitation services for people with vision impairments. All the recruited participants are daily mobile phone and screen reader users. The participants were asked to perform the menu selection tasks on a Google Pixel phone with 1080×1920 pixels and 441 PPI. The experiment APP ran on Android API 29. The modified TalkBack was based on version 8.1. The navigation scaling (the option to navigate via heading, links or other level of components) was turned off.

ID	Age	Gender	Legally Blind?	Diagnosis
1	38	Female	Yes	Congenital glaucoma
2	41	Male	Yes	Congenital glaucoma
3	55	Male	Yes	Optic atrophy
4	36	Female	Yes	Congenital glaucoma
5	46	Male	Yes	NA
6	34	Male	Yes	Born with no vision
7	48	Female	Yes	Leber congenital amaurosis
8	59	Female	Yes	Congenital cataract
9	60	Male	Yes	Detached retina
10	56	Male	Yes	Retinal degeneration

Table 1: Demographic information of our participants. All of them are legally blind and daily phone users. ("NA" means the participant was not willing to share the information.)

4.2 Experiment Design

We used a [2×3] within-subject design. The two independent variables were: (1) menu length with 2 levels (6-item and 10-item linear menu), and (2) menu arrangement with 3 levels (Alphabetic, Grouped and Random). The grouped arrangement is based on the menu attribute (i.e. animals and fruits). In each group, the items are in alphabetic order. We adopted the alphabetic and grouped arrangements because they are commonly used [63].

We picked two different types of item labels for the participants to select: animals and fruits, each made up half of the menu. For 6-item menu, the item labels were *apple, cherry, eagle, jaguar, orange, penguin*. For 10-item menu, the item labels were *apple, bear, cherry, eagle, jaguar, lemon, orange, peach, penguin, zebra*. Instead of adopting practical menu items such as "Personal Hotspot" in phone settings, the names of animals and fruits are easier for the participants to recognize and distinguish from other items. Following previous works in menu selection [15, 47], smartphone APP launching [50], and target selection [3, 18, 45, 64], we generated number of occurrences of each item from the Zipf distribution:

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N}(1/n^s)} \tag{3}$$

where N is the number of menu items, $k \in \{1, 2, \cdots, N\}$ is the rank of each menu item, and s is the value of the exponent characterizing the distribution and is set s to 1 in this paper. Among 60 trials within a condition, the number of occurrences for each item in a 6-item menu would be 24, 12, 8, 6, 5, 5 and the occurrences for each item in a 10-item menu would be 20, 10, 7, 5, 4, 3, 3, 3, 3, 2. The frequencies were randomly assigned to each menu item, and the order of the 60 trials were randomized. We counter-balanced the user study by using random orders of the 6 conditions for each user (2 menu lengths × 3 menu arrangements). In the user study, we set the height of each menu item as 10 mm based on the material design guideline [23].

In total, there were 3,600 trials (2 menu lengths × 3 menu arrangements × 60 trials in each condition × 10 participants). During the user study, the swiping, gliding, direct touch, and double-tapping actions could be recognized as accessibility events[1]. Every accessibility event and its timestamp were recorded for further analysis.

4.3 Procedure

At the beginning of the study, we let the participant sit in a chair near a table in a comfortable position, and hold the phone with one hand and use the other to interact with the phone. We verified if the participant knew the actions that the system supported – swiping, gliding, direct touch, and selection (illustrated in Fig. 1), by asking the participant to perform several trials in a warm-up session. The item labels in the warm-up session were popular cities around the world.

After the warm-up session, the participants were asked to perform 6 different conditions (2 menu lengths × 3 menu arrangements). At the beginning of each trial, the screen reader would read out "Next Target", followed by three time of the target item to inform the participant which menu item she should select. For example, if the target item is "Apple", the screen reader would read out "Next target, apple, apple, apple".

After learning the label name of the target, the trail began. The participant could use gliding, swiping, or touch touch to navigate to the target item, relying the audio feedback from TalkBack in Android. After the target item was reached, the participant could double-tap the screen to select the target. If it was a correct selection, the screen reader would hint "Next target", followed by reading out what the next target was repetitively. If it was an incorrect selection, the screen reader would speak "Wrong selection", followed by indicating what the correct target should be. Besides the auditory feedback, haptic feedback was also enabled to indicated the number of the tapping (for selection action). After all the required trials were completed successfully within a condition, the screen reader would say "Task finished, thank you". Fig. 3 shows a participant during the study.

[1]https://developer.android.com/reference/android/view/accessibility/AccessibilityEvent

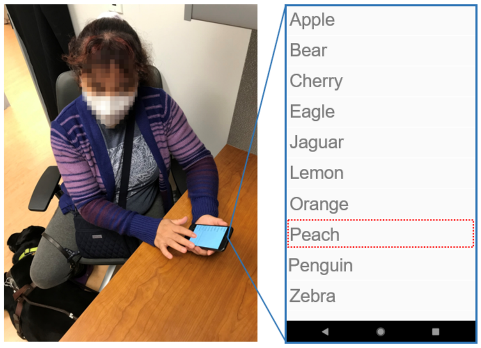


Figure 3: One participant was selecting a menu item.

4.4 Results

We analyzed the selection time and the action used in the user study. The selection time is defined as the duration starting from the beginning of a trial to the moment that the target menu item is selected by the participant.

4.4.1 Data Cleaning. We first cleaned the data by removing outliers, which happened when users forgot the target name or interrupted by unexpected events, e.g. resting in the middle of a trial. Specifically, for each condition, we removed the trials whose selection time is outside of 3 standard deviations of the mean value. In total, we removed 77 trials (out of 3,600 trials), which is 2.14% of the data collected. We used the cleaned dataset for further analysis.

4.4.2 Selection Time. Fig. 7a shows the selection time (standard deviation in parentheses) for each condition in the experiment. The selection time for 10-item menus was longer than that of 6-item menus. For 6-item menus, the selection time for different menu arrangements were similar. While for 10-item menus, the grouped layout had the shortest selection time, and the selection time for the random menu layout was higher than that of the others.

We fit a linear mixed effects model [21] with the menu length and menu arrangement as fixed effects, and the participants' ID and trials as random effects. The type III ANOVA results showed significant main effect for menu length ($F_{1,\ 3450} = 91.51,\ p < 0.001$), menu arrangement ($F_{2,\ 3450} = 7.27,\ p < 0.001$), and one significant interaction effect: menu length×menu arrangement ($F_{2,\ 3450} = 13.60,\ p < 0.001$) on selection time (The degree of freedom is large as all trials were included in the analysis, which is typical for mixed effect model.). Post-hoc tests with Holm adjustment showed that for 10-item menus, the difference was statistically significant for alphabetic vs. random ($P < 0.001$) and grouped vs. random ($p < 0.001$), and no significant difference between alphabetic vs. grouped ($p = 0.25$). For 6-item menus, there was no significant difference for alphabetic vs. grouped ($p = 0.42$), alphabetic vs. random ($p = 0.81$), and grouped vs. random ($p = 0.43$).

To analyze how the search time varies as the experiments proceed, we manually divided the 60 trials in each condition into 6 blocks, i.e. 10 trials per block. Fig. 8a shows the selection time in 6 blocks for the 6 conditions. We can observe that for almost all conditions, the selection time had a decreasing trend when the experiments proceed. As users were becoming more experienced, they could use fewer actions to select the menu item, e.g. the gliding action in all 6 conditions dropped by 7.25% in block 6 (537 times) compared with block 1 (579 times), resulting in this decreasing trend. We fit a linear mixed effects model with the menu length, menu arrangement and block as fixed effects, and the participants' ID and trials as random effects. The type III ANOVA results showed main significant effect for the block ($F_{5,\ 54} = 15.29,\ p < 0.001$) on selection time.

4.4.3 Selection Accuracy. Fig. 4 shows the accuracy of menu selection across conditions. The overall selection action accuracy was 96.02% (3523 out of 3669 selections were successful). As shown in the Fig. 4, the selection accuracy across conditions was close to each other. A repeated measures ANOVA did not show a significant main effect of the menu arrangement ($F_{2,\ 18} = 0.29,\ p = 0.75$) or menu length ($F_{1,\ 9} = 0.26,\ p = 0.62$) on selection accuracy. It did not show a significant interaction effect between menu length and menu arrangement either ($F_{2,\ 18} = 1.45,\ p = 0.26$).

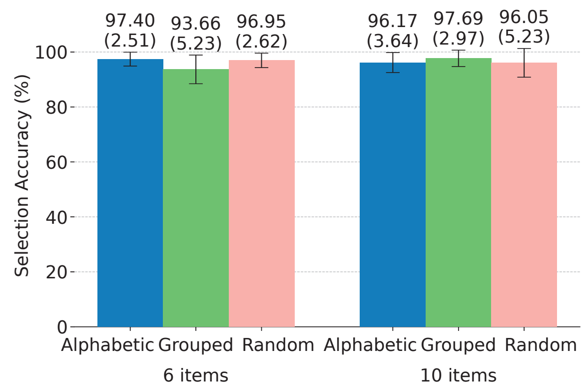


Figure 4: The mean selection accuracy (95% confidence interval) by condition. The overall selection accuracy was 96.02% (3523 out of 3669 selections were successful).

4.4.4 Actions. In this part, we analyzed the action used by the 10 participants. The actions were extracted based on the accessibility events in the cleaned dataset. In total, there were 13,856 actions in the cleaned dataset: 4,627 swiping actions, 3,250 gliding actions, 2,310 direct touch actions and 3,669 selection actions.

We first plotted the matrix of consecutive occurrence of actions in Fig. 5 to see the relative order of actions. In the figure, the action on the Y axis happens before that on the X axis, the numbers mean occurrences, and the BoT means beginning of a trial. We can see that the direct touch action always happened before the gliding action (as shown by cell 2310). That means the participant first used direct touch action to quickly move to an item that is close to the target and then used gliding to search for the target. We reflected this finding when building the model in Section 5.

We then plotted the overall compositions of the four actions in different conditions in Fig. 9a to show the relative amount of each action was used in the study. We can see that the action composition was stable across different conditions. However, we noticed in the user study that each individual participant had different preference over the available actions. We plotted the action composition

Figure 5: Matrix of consecutive occurrence of actions, where the action on the Y axis happened first, the numbers means the occurrences, and BoT means "Beginning of Trial". We can see that direct touch action always happened before gliding action.

of each participant across the 6 conditions in Fig. 6. We can see that participant 2, 6, 7 almost never used the swiping action, and the other participants used all the four actions available but with different amount (which might be caused by different search strategy). Based on the preference of the participants, we could divide the participants into two subgroups: (A) Participant 1, 3, 4, 5, 8, 9, 10: They used all available actions, (B) Participant 2, 6, 7: They did not use swiping action. We reflected this finding during model evaluation by considering two training strategies (Section 5.1).

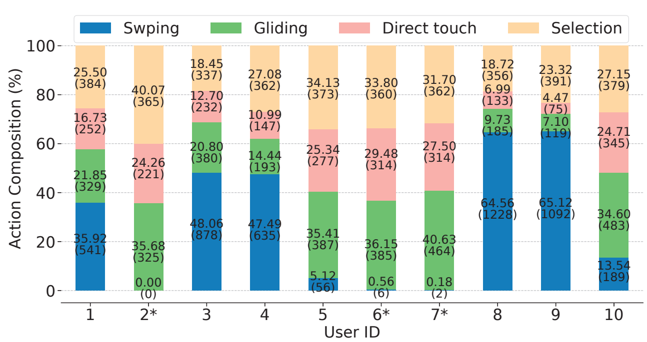


Figure 6: Action composition of each of the 10 participants. The results indicated that different participant has different menu selection strategy. That is, participant 2, 6, 7 (marked by *) did not use the swiping action, and others used all the four actions.

5 MODEL IMPLEMENTATION

In this section, we describe how we obtained the parameters of the model. The process consisted of two phases. First, we obtained the parameters of the input action and the memory models, including both μ and σ of six models in Table 2, and σ_{DTU} and σ_{PM} introduced in Section 3.2. This was done by fitting their empirical parameters to the user study data, which is similar to the fitting of empirical parameters of Fitts' law. Second, we trained the reinforcement learning model's policy, which predicts what action to take given an observation, in a simulated menu environment (see Fig. 2). This environment mimics a menu and allows the model to learn via trial and error, and it was built based on the input action and memory models learned in the first phase. To sum up, in no part of this parameter fitting process did our model have access to users' action sequences.

5.1 Parameters of Input Action and Memory Models

In this section, we describe how to obtain the parameters of input action and memory models from the data collected in the user study. We assumed the time cost for executing a swiping, gliding, direct touch, and selection action followed a Gaussian distribution. Recall that the direct touch uncertainty model and the position memory model were also modeled by a Gaussian distribution (Section 3.2 and Section 3.3). The means and standard deviations were obtained based on the data from the user study using the following method:

- Swiping action. We estimated the mean and standard deviation of its time cost from the user study data.
- Selection action. A user performed a double tap to commit the selection of a menu item. Similar to swiping action, we estimated its mean and standard deviation of time cost from the data collected in the user study.
- Gliding action. A user moves the finger from one to another location on the screen while keeping the finger in contact with the screen in the entire course of gliding. We adopted the gliding-based pointing model for blind users proposed by Ko et al. [41] to estimate the mean of time cost for a gliding action. The gliding-based pointing model (Equation 4) [41] states that the mean selection time (MT) for gliding is linearly related to both finger travel distance A and index of difficulty of the task $ID = \log_2(A/W + 1)$ where W is the width of a target:

$$MT = a + bA + c\log_2(\frac{A}{W} + 1). \tag{4}$$

  We empirically determine the parameters a, b, and c from gliding data collected in the user study. We used the observed standard deviation for a particular ID as the standard deviation for MT for the same ID.
- Direct touch action. Direct touch is an action where the user takes the finger off the screen, travels in the air, and lands it on a location on the screen. We adopted the Fitts' law (Equation 5) [19, 48] to estimate the mean of time cost for the direct touch action.

$$MT = a + b\log_2(\frac{A}{W} + 1). \tag{5}$$

  We empirically determine the parameters a and b from the direct touch action data collected in the user study. Similar to the gliding action, we used the observed standard deviation for a particular ID as the standard deviation for time cost under the same ID.

- Direct touch uncertainty model. The mean of the direct touch uncertainty model μ_{DTU} was set to the center of the intended menu item. We used grid search to determine standard deviations: (1) σ_{DTU} in the direct touch uncertainty model (Section 3.3) which was searched in the range [0.25, 2] with a step size 0.25.
- Position memory model. The mean of the position memory μ_{PM} was set to the center of the target menu item. We also used grid search to determine the standard deviation σ_{PM} (Equation 1) which was searched in the range [0.5, 5] with a step size 0.5.

We optimized the σ_{DTU} and σ_{PM} together based on the selection time (Fig. 7a) and action composition (Fig. 9a) in the user study. The optimal parameters were determined by minimizing the sum of the mean absolute percentage error (MAPE) for selection time across conditions and the MAPE for action composition across action × condition.

Two Training Strategies. We adopted two strategies to obtain the parameters:

- One-group parameters: We assumed all the participants in the study were from the same pool and obtained one set of parameters for all the users. In other words, there was only one set of parameters for all users.
- Two-subgroup parameters: To reflect the observation that some users used all four types of actions (i.e., swiping, selection, gliding, and direct touch) for menu selection, and some users never used the swiping action. We obtained two sets of parameters for two subgroups of users introduced in Section 4.4.4 based on their corresponding user study data separately. More specifically, one set of parameters was for subgroup A with participants 1, 3, 4, 5, 8, 9, 10 who used all 4 types of actions, and the other set of parameters was for subgroup B with participants 2, 6, and 7 who never used the swiping action. We expected the model created from these two sets of parameters (called two-subgroup model) would better capture the difference between these two types of users.

The procedures of obtaining both one-group and two-subgroup parameters were identical.

We summarized the optimized parameters learned from all user data for the input actions and memory model in Table 2, and the standard deviations of the gliding action and direct touch action were summarized in Table 4 in the Appendix.

5.2 Implementing and Training Menu Selection Models

After obtaining parameters for input action and memory models, we could implement a simulated menu selection environment. We trained our reinforcement learning based menu selection model based on simulated interactions following the reinforcement framework described in Section 3 (Fig. 2). As we adopted two strategies for obtaining the parameters, we created two models for the two strategies accordingly. The first one is called one-group model, which was created based on the one-group parameters, and the second one is called two-subgroup model, which was created based on the two-subgroup parameters. The two-subgroup model consists of two models, for Subgroup A and B, respectively. The procedures of creating each of these models were identical, as described below.

We implemented a menu selection environment simulating real-world menu selection scenario using OpenAI Gym [9]. The environment simulates interactions between the agent and the environment. In the environment, each menu item has the same height as in the real study, i.e. 10mm. The occurrences of the menu items were controlled by a Zipf distribution (Equation 3) with $s = 1$ same as the user study. Because in the user study, the participants completed 60 trials on each menu layout, the agent did the same in the training. After every 60 trials, we randomly reset the menu layout to one in the user study, and reset the position memory, and reset the finger and focus position to the top of the menu.

We then used the DQN to learn the optimal policy for the agent, and used the implementation of the DQN algorithm in the Stable-Baselines3 library[2]. We trained one policy network for the 6 conditions (2 menu length × 3 menu arrangements) for all users. For the 6-item menu length condition, we padded 4 zeros to the end of the feature vector to make the feature vector the same for different menu length. The DQN was trained with the following hyper-parameters: MlpPolicy (3-layer neural network, hidden size: 64, activation function: ReLU) as the policy network, learning_rate = 0.001, learning_starts=0, batch_size=256. The other hyperparameters were the default ones: exploration_fraction=0.1, optimizer: Adam, gamma=0.99. For reproducibility, we used 20 different random seeds to train 20 models and average their results at trial level, while the optimal σ_{PM} and σ_{DTU} were found using a specific random seed.

6 MODEL EVALUATION

To understand whether the reinforcement learning framework described in Section 3 can truthfully simulate menu selection behavior of blind users, based on the basic action and memory models, we compared the simulated menu selection behaviors and predicted menu selection performance (i.e., menu selection time) from the menu selection model with the observed data in the user study. To further evaluate the generability of the menu selection models, we performed more strict tests: we split the user study data into training and testing datasets, and perform leave-trial-out and leave-user-out validations in which the data in the testing dataset were not used in parameter estimation and completely unseen by the models. Such validations further test the generability of the menu selection models.

6.1 Simulating Menu Selection Behaviors

We first simulated menu selection behaviors using both one-group and two-subgroup models, separately. For the one-group model, we simulated menu selection for 10 users and 6 conditions × 60 trials for each user, to match the user study setting where 10 users participated in the study and each performed 6 conditions × 60 trials. The menu layouts in the simulation also matched the layouts in the user study. For the two-subgroup model, we used the subgroup A model to simulate trials for 7 users and the subgroup B model to simulate trials for 3 users, each user with 6 conditions × 60 trials. This setup matched the user study in which 7 users belonged to

[2] https://github.com/DLR-RM/stable-baselines3

Action and Memory	Model	One-group model		Two-subgroup model (Subgroup A)		Two-subgroup model (Subgroup B)	
		μ	σ	μ	σ	μ	σ
Swiping	MT$\sim \mathcal{N}(\mu_{SP}, \sigma_{SP})$	1.11	0.99	1.11	0.99	/	/
Selection	MT$\sim \mathcal{N}(\mu_{ST}, \sigma_{ST})$	1.13	0.93	1.19	0.89	0.98	0.99
Gliding	MT$\sim \mathcal{N}(\mu_{G}, \sigma_{G})$	$0.9 + 0.13A + 0.97ID$, $R^2 : 0.98$	AP	$0.86 + 0.02A + 1.30ID$, $R^2 : 0.91$	AP	$0.88 + 0.29A + 0.57ID$, $R^2 : 0.94$	AP
Direct touch	MT$\sim \mathcal{N}(\mu_{D}, \sigma_{D})$	$2.09 + 0.21ID$, $R^2 : 0.49$	AP	$2.09 + 0.18ID$, $R^2 : 0.22$	AP	$1.88 + 0.39ID$, $R^2 : 0.65$	AP
Direct touch uncertainty	$m_i \sim \mathcal{N}(\mu_{DTU}, \sigma_{DTU})$	Center of intended menu item	0.5	Center of intended menu item	0.25	Center of intended menu item	0.5
Position memory	$m_i \sim \mathcal{N}(\mu_{PM}, \sigma_{PM})$	Center of target menu item	4.5	Center of target menu item	4.5	Center of target menu item	1.5

Table 2: Parameters of input action, direct touch uncertainty, and position memory models. MT is the movement time of each action. $ID = \log_2(A/W + 1)$ where A is the absolute moving distance and W=1 cm is the menu item height. R^2 is the coefficient of determination measuring the fitness of the model. AP means the standard deviation which is an empirically determined value from the user study and is further explained in Table 4 in the Appendix. m_i refers to the finger landing position for the direct touch uncertainty model, and the recalled position of the target menu item in the position memory model.

subgroup A and 3 users belonged to subgroup B. We then aggregated simulated trials from subgroups A and B for analysis.

6.2 Simulated vs. Observed Selection Time

Fig. 7b and Fig. 7c shows the simulated selection time by conditions for the one-group model and two-subgroup model respectively. The mean absolute error (MAE) across condition was 0.61 s (mean absolute percentage error(MAPE): 10.44%) and 0.39 s (MAPE: 6.71%) respectively. The results showed that our model can simulate the effect of menu length on the selection time, i.e. the selection time for 10-item menu was longer than 6-item menu. We fit linear mixed-effect models on the results simulated by the one-group model and two-subgroup model separately with menu length and menu arrangement as fixed effects and the participants' ID and trials as random effects. The ANOVA results showed that the menu length has a significant main effect on menu selection time, for both the one-group model ($F_{1,\ 3526} = 265.80,\ p < 0.001$) and two-subgroup model ($F_{1,\ 3526} = 201.31,\ p < 0.001$), which reflects the main effect of menu length on selection time in observed user study data ($F_{1,\ 3450} = 91.51,\ p < 0.001$). Meanwhile, our model can also simulate the effect of different menu arrangements on selection time for 10-item menus. That is, the random arrangement has the longest selection time, and the grouped arrangement has the shortest selection time. For the 6-item menus, our model overestimated the effect of menu arrangements. Our model simulated that the random layout has the longest selection time and the grouped layout has the shortest selection time, which is similar to the 10-item case. While in the user study, for 6-item menu, the selection time of the three arrangements were close to each other. The ANOVA results showed that the menu arrangement has a significant main effect on menu selection time, for both the one-group model ($F_{2,\ 3526} = 25.05,\ p < 0.001$) and two-subgroup model ($F_{2,\ 3526} = 21.42,\ p < 0.001$), and that is the same as the user study data ($F_{2,\ 3450} = 7.27,\ p < 0.001$).

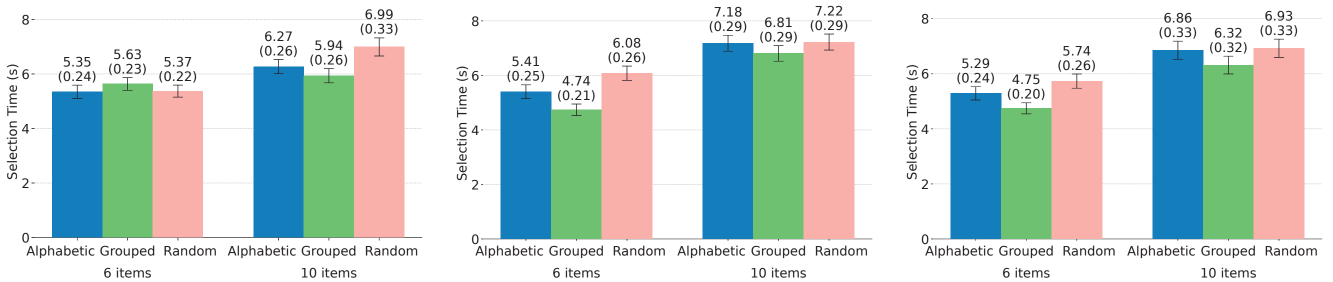
We also plotted the simulated selection time for the 6 conditions in each block in Fig. 8b and Fig. 8c for the one-group model and two-subgroup model respectively. Compared with that in the real user study (Fig. 8a), we can see that our model simulated the learning effect during the menu selection task. That is, the selection time decreases as the menu selection task proceeds. The reason behind this phenomenon is that with more interaction experience, the agent would have stronger memory of menu item positions, so that it can recall the item position more accurately which leads to less selection time (see the position memory model in Section 3.2). We fit a linear mixed effects model with the menu length, menu arrangement and block as fixed effects, and the participants' ID and trials as random effects. The ANOVA results showed main significant effect for the block on selection time, for both the one-group model ($F_{5,\ 54} = 5.92,\ p < 0.001$) and the two-subgroup model ($F_{5,\ 54} = 4.72,\ p < 0.01$), which is identical with the user study ($F_{5,\ 54} = 15.29,\ p < 0.001$).

6.3 Comparison with Input Action Model based Methods

A previous approach for modeling menu selection is to use the input action models (e.g., Fitts' law) to predict the item selection time (e.g., [15]). One advantage of our reinforcement learning based model is that it is able to predict the selection behaviours, while existing approaches cannot (e.g., a Fitts' law based model predicts the selection time only). Additionally, we expect our model would outperform the previous input action model based methods in prediction accuracy as our model more accurately reflects the menu selection behaviors. To evaluate our expectation, we compared our model with the following three input action model based methods in predicting the menu selection time:

- Fitts' law based method: We used the Fitts' law to model menu selection time. More specifically, we assumed that a user would travel the finger in the middle of air and land it on the screen and then double tap to select a menu item. We used Fitts' law to model each finger traveling action.
- Gliding model based method: We used the mixture gliding model for blind users [41] to predict the menu selection time. We assumed that the users would glide the finger on screen to reach the intended menu item, and double tap to confirm the selection.
- Swiping model based method: We assumed that a user would perform a sequence of swiping actions to reach the intended menu item, and double tap to confirm the selection.

We also made the following assumptions in order to apply the input action model based methods. In Fitts' law based method, we assumed the user knew the approximate location of the target so that he/she could land the finger to select the target, and in Gliding and Swiping model based methods, we assumed the user knew the gliding and swiping directions. Although these assumptions

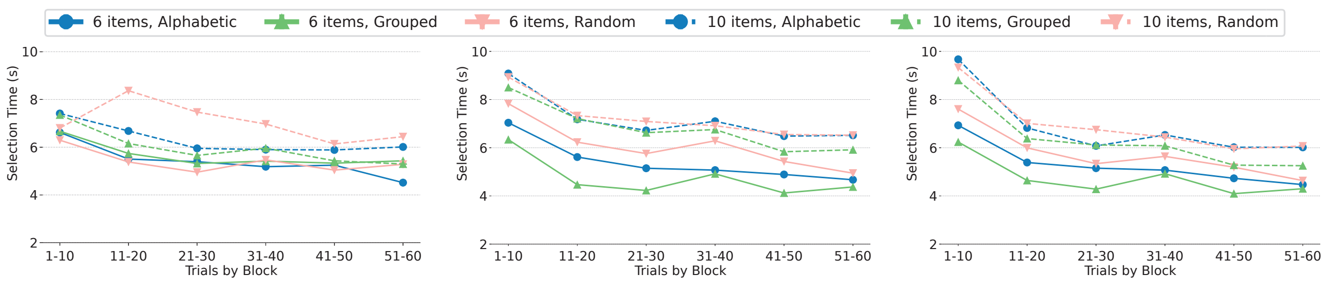


(a) Selection time in the user study

(b) Simulated selection time by the one-group model

(c) Simulated selection time by the two-subgroup model

Figure 7: The mean selection time (95% confidence interval) by condition. The MAE between the selection time simulated by the model and that in the user study is 0.61 s for one-group model, and 0.39 s for the two-subgroup model.



(a) Selection time in the user study

(b) Simulated selection time by the one-group model

(c) Simulated selection time by the two-subgroup model

Figure 8: The mean selection time by block. We manually divided the 60 trials into 6 blocks, i.e. each 10 trials a block.

reflected only the expert users' performance, they were necessary for applying the input action models. The parameters of the Fitts' law, gliding model [41], swiping model were the same with the direct touch action model, the gliding action model, and the swiping action model respectively in Table 2 for the one-group model.

The evaluation results (Table 3) showed that the reinforcement learning (RL) based model outperformed all the three input action model based methods in modeling menu selection times on all trials. Also, a closer look at the results (Fig. 16 in the Appendix) showed that the input action model based methods failed to reflect the effect of menu arrangement on selection time.

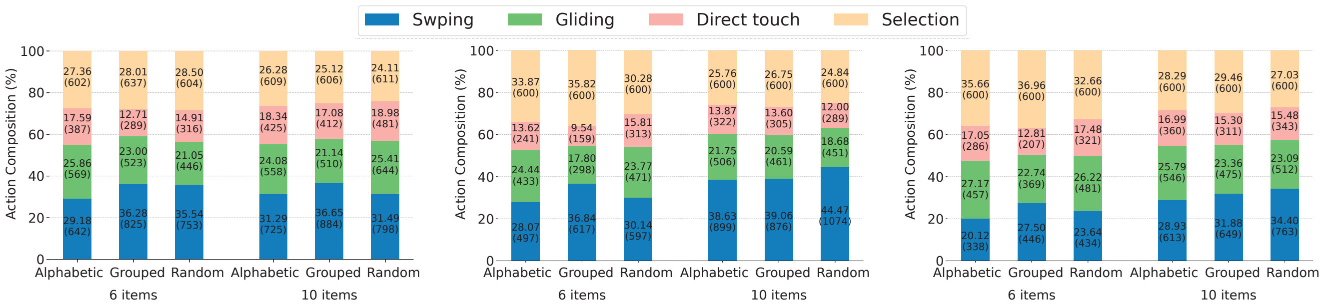
Method	MAE (MAPE) for Selection Time
RL based model (One-group)	0.61 s (10.44%)
RL based model (Two-subgroup)	**0.39 s (6.71%)**
Fitts' law based method	2.55 s (42.73%)
Gliding model based method	1.51 s (25.15%)
Swiping model based method	1.80 s (30.65%)

Table 3: The MAE (MAPE) of selection time for our reinforcement learning (RL) based model and the three input action model based methods.

6.4 Simulated vs. Observed Action Composition

We plotted the simulated action composition in Fig. 9b and Fig. 9c for the one-group model and two-subgroup model respectively. Compared to the results in the user study (Fig. 9a), the MAE (across action × condition) of action composition for the one-group model was 3.78% (MAPE: 15.67%), and 3.89% (MAPE: 14.47%) for the two-subgroup model. Considering only the swiping, gliding and direct touch actions, the MAE for the one-group model was 3.98% (MAPE: 17.04%), and the MAE for the two-subgroup model was 3.48% (MAPE: 12.97%). Our model accurately predicts the percentages of actions but underestimates the numbers of actions. For example, one-group and two-subgroup models predicted that the total numbers of swiping actions were 4560 and 3243, while the actual swiping actions was 4627. One reason is that users' input behavior could deviate from the prediction made by the model. For example, our model predicts that it would take 4 swiping actions from item #1 to #5 in a linear menu, while some users may perform 6 swiping actions because they overshoot by one item and then correct it. Our model reflects the optimal input behaviors of users, while some users' input actions may deviate from such prediction.

Our model predicted 100% selection accuracy, and the selection accuracy in the user study was 96.02%. There is a small discrepancy between observed selection accuracy and model prediction. We observed that blind users occasionally made selection errors as they might forget the target name or mistakenly execute a double



(a) Action composition in the user study

(b) Simulated action composition by the one-group model

(c) Simulated action composition by the two-subgroup model

Figure 9: The action composition (occurrences) by action × condition. The MAE across action × condition between the action composition simulated by the model and that in the user study is 3.78% for the one-group model, and 3.89% for the two-subgroup model.

tapping. These types of mental errors/mistakes are not captured by the reinforcement learning model.

Overall, the small differences between the observed action composition and model prediction showed that our model can accurately simulate the relative amount of action used by the users, reflecting the menu selection behavior and preference. Additionally, the results showed that the two-subgroup model achieved higher performance than the one-group model. It confirms the effectiveness of considering the choice of actions in the model.

6.5 Simulated vs. Observed Menu Selection Strategy

In addition to the selection time and action composition, we also analyzed the selection strategies. We observed that there existed three strategies for menu selection: (1) Swiping only: the user swipes to a menu item and selects it, (2) Gliding only: the user lands the finger to a specific item (optional), glides to the target item, and selects it, and (3) Swiping and Gliding: the user uses both strategy (1) and (2) to select a target item. As shown in Fig. 10, our model was able to predict the composition of strategies observed in the user study. In the user study, there were 19.33% of trials for swiping only, 53.25% for gliding only, and 11.07% for swiping and gliding. The predictions by the two-subgroup model were 22.16% for swiping only, 41.86% for gliding only, and 15.5% for swiping and gliding, which were close to the observed data.

6.6 Train-Test Split Evaluation

To further evaluate the generability of the model, we estimated the parameters for action and memory models with 80% of data, and held 20% data for testing. This is a more strict validation as the 20% data were completely unseen by the model: the testing data were not even used to estimate the parameters for action or memory models. We performed two types validations following this scheme: (1) leave-trial-out validation, (2) leave-user-out validation. We performed train-test split evaluation on the one-group model only, as there was not sufficient data for such tests for the the two-subgroup model (e.g., there were only three users in total for the subgroup B). We performed this evaluation with 10 repetitions.

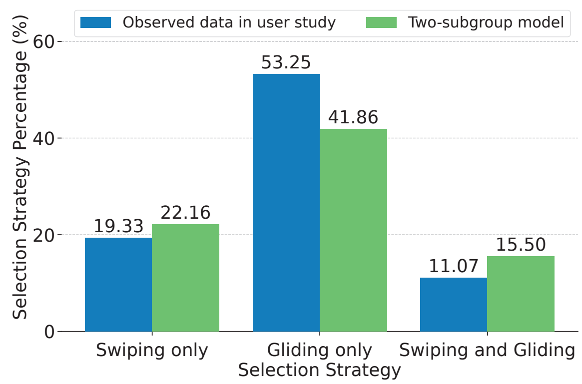
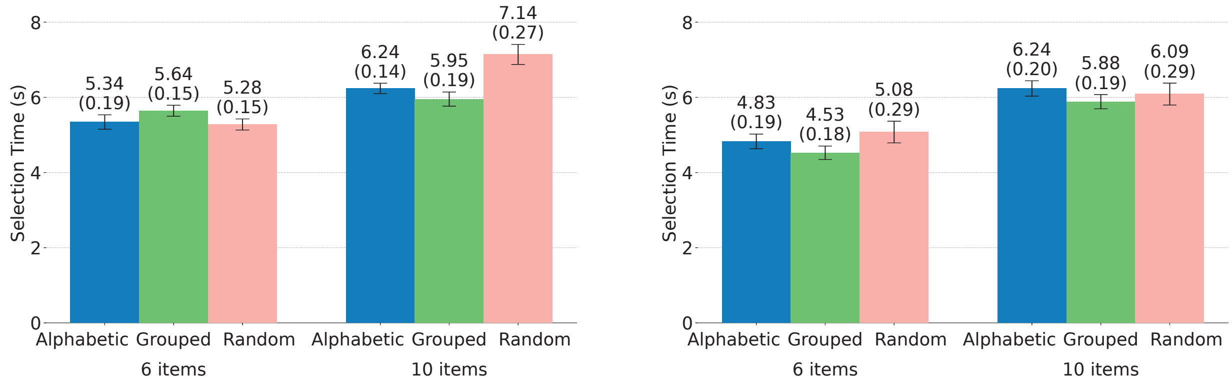


Figure 10: Observed selection strategies (blue) vs. predicted selection strategies by the two-subgroup model (green).

6.6.1 Leave-trial-out Validation. We randomly sampled 80% of trials as the training dataset which were used to estimate the parameters for action and memory models, following the procedure described in Section 5.1, and the remaining 20% data as the testing dataset. We then built a menu selection model based on the estimated parameters following the procedure described in Section 5.2, simulated the menu selection behaviors, and compared the simulated with testing data. We repeated the same procedure 10 times. In each repetition, we simulated 6 conditions × 60 trials for each of the 10 users. As we have only 20% of data withheld in the testing dataset, we identified the matching trials of these 20% of data in the simulated trials, and compared them against the testing data.

Fig. 11 shows the averaged mean selection time of the 10 repetitions in the testing dataset and predicted by the one-group model. The model was able to predict the effects of menu length on selection time: both simulated results and the testing data showed that the menu selection time on the 10-item menus were longer than that on the 6-item menus. Across all 10 repetitions, the mean (SD) of MAE for menu selection time was 0.58 s (±0.09 s), and the mean (SD) of MAPE was 9.71% (±1.54%).



(a) Selection time in the testing dataset

(b) Simulated selection time by the one-group model

Figure 11: The averaged mean selection time (95% confidence interval) by condition of the 10 repetitions of the leave-trial-out validation. The mean MAE (SD) across the 10 repetitions was 0.58 s (±0.09 s).

The analysis on the action composition (Fig. 12) also showed that our model can predict the action composition. The mean (SD) of MAE for the action composition prediction was 5.32% (±1.00%), and the mean (SD) of MAPE was 21.44% (±4.37%). Considering only the swiping, gliding and direct touch actions, the mean (SD) of MAE was 5.04% (±1.14%), and the mean (SD) of MAPE was 20.85% (±5.09%).

6.6.2 Leave-user-out Validation. We used 8 random user's data as the training dataset which were used to estimate the parameters for action and memory models, and held the remaining 2 user's data as testing dataset. Similar to the leave-trial-out validation, we then built a menu selection model based on the estimated parameters following the procedure described in Section 5, simulated the menu selection behaviors for the two users with 6 conditions $\times$ 60 trials, and compared the simulated with testing data. Similarly, we repeated the same procedure 10 times.

Fig. 13 shows the averaged mean selection time of the 10 repetitions in the testing dataset and predicted by the one-group model. The analysis on testing data showed that the mean of menu selection time on the 10-item menu was longer than that on the 6-item menu. Across all 10 repetitions, the mean (SD) of MAE for menu selection time was 0.93 s (±0.54 s), and mean (SD) of MAPE was 15.37% (±7.80%).

Regarding the action composition (Fig. 14), the mean (SD) of MAE for action composition was 9.04% (±2.38%), and the mean (SD) of MAPE was 63.91% (±31.29%). Considering only the swiping, gliding and direct touch actions, the mean (SD) of MAE was 10.17% (±3.27%), and the mean (SD) of MAPE was 77.64% (±40.48%).

In sum, the leave-trial-out validation showed that our model can predict the selection time and the effect of menu length, as well as the action composition. The MAEs were close to the the results presented in previous sections (Section 6.2 and Section 6.4). For the leave-user-out validations, our model also captured the effect of menu length. The prediction error was higher than that of the leave-trial-out validation. It was probably because the menu selection behavior of the user in the testing dataset was different with that of the user in the training dataset.

7 DISCUSSION AND FUTURE WORK

Even though menu selection for sighted users has received a lot of attention in the literature [2, 5, 6, 13, 15, 16, 41], understanding of blind users' behavior in the same task is limited. In this paper, we developed a computational model of how users who are blind perform menu selection. Our results suggest that modeling how users remember menu item positions is the key to understand menu selection in the absence of visual feedback. This paper also extends the existing literature on models of users who are blind [17, 41] to considering multiple actions (swiping, gliding, direct touch and selection), instead of just one.

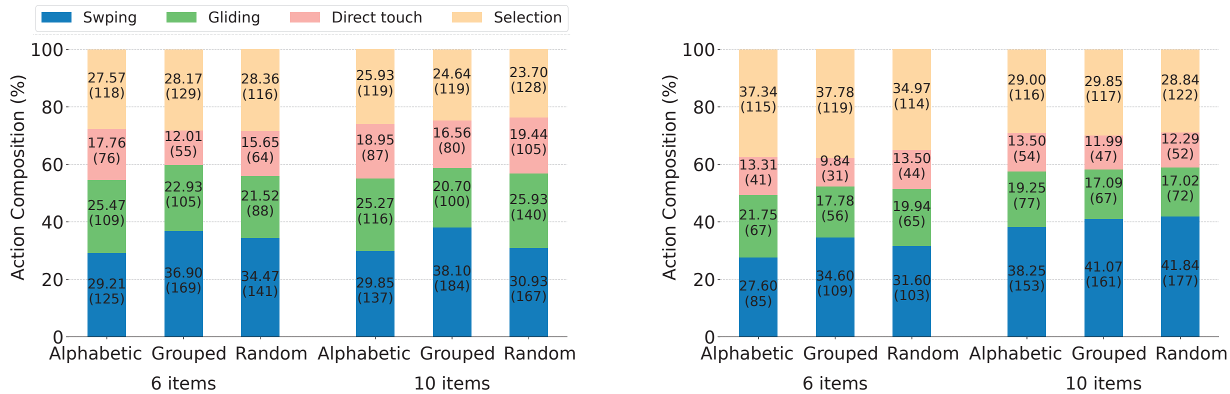
7.1 Effects Captured by the Model

The proposed model can successfully simulate the following effects observed in the user study:

- Longer menus have longer selection times.
- Selection time decreases after some iterations of the menu selection task have been performed.
- The action composition: the percentages of swiping, gliding, direct touch and selection actions.
- The menu selection strategies used by the users.

Compared with input action model based methods (Section 6.3), our reinforcement learning based model achieved lower errors on the selection time prediction. More importantly, our model is able to predict the menu selection strategies and action composition, while the input action model based methods (e.g., Fitts' law based method) predict the menu selection time only.

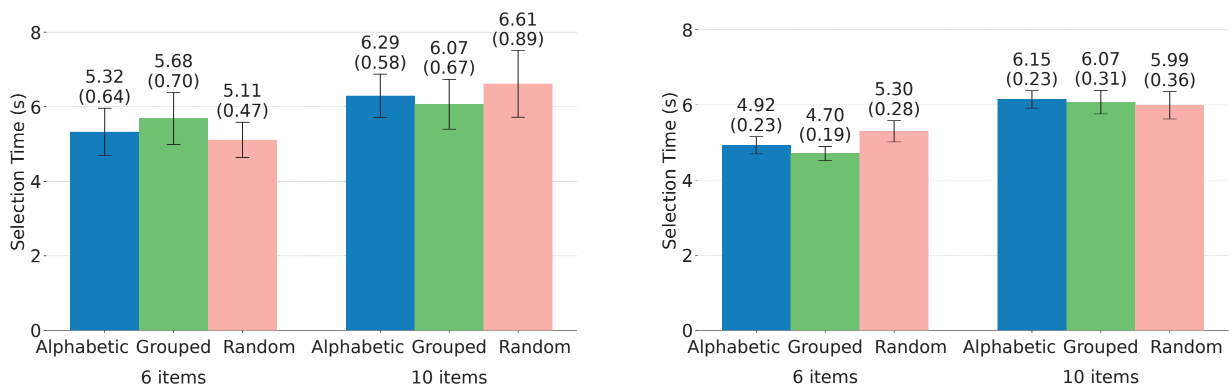
The key to modeling the blind user's menu selection behavior is the position memory model, which describes how the user remembers/recalls the menu item position. Introducing the memory model enables our model to account for the learning effects, and to predict how the selection strategies change as the user gets more familiar with the menu layout.



(a) Action composition in the testing dataset

(b) Simulated action composition by the one-group model

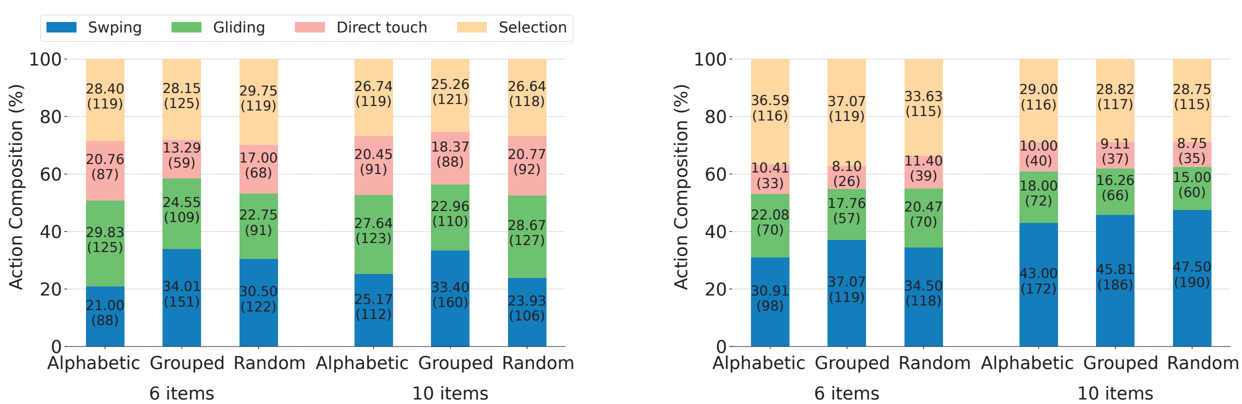
Figure 12: The averaged action composition (occurrences) by action × condition across the 10 repetitions in the leave-trial-out validation. The mean MAE (SD) across the 10 repetitions was 5.32% (±1.00%).



(a) Selection time in the testing dataset

(b) Simulated selection time by the one-group model

Figure 13: The averaged mean selection time (95% confidence interval) by condition of the 10 repetitions of the leave-user-out validation. The mean MAE (SD) across the 10 repetitions was 0.93 s (±0.54 s).



(a) Action composition in the testing dataset

(b) Simulated action composition by the one-group model

Figure 14: The averaged action composition (occurrences) by action × condition across 10 repetitions in the leave-user-out validation. The mean MAE (SD) across the 10 repetitions was 9.04% (±2.38%).

7.2 Applications of the Model

Understanding menu selection behavior of users who are blind is critical for designing functional user-adaptive systems. The proposed model could serve as a useful tool in interface design, optimization and evaluation, since it can be used to simulate user behavior in place of running an expensive user study.

For example, as the model can simulate the learning effect on a menu layout, it could provide a *quantitative* understanding of the performance of menus with different lengths for both novice and expert users. Fig. 15 shows the predicted menu selection time by menu length and practice block for an alphabetically ordered menu. It shows that short menus (e.g., 4-item menu) require little learning while long menus (e.g., 10-item menu) require a certain amount of learning to reach optimal performance. Fig. 15 further shows the time cost for menus with different lengths and practice trials. Such an understanding would guide a designer in choosing appropriate menu lengths when designing interfaces.

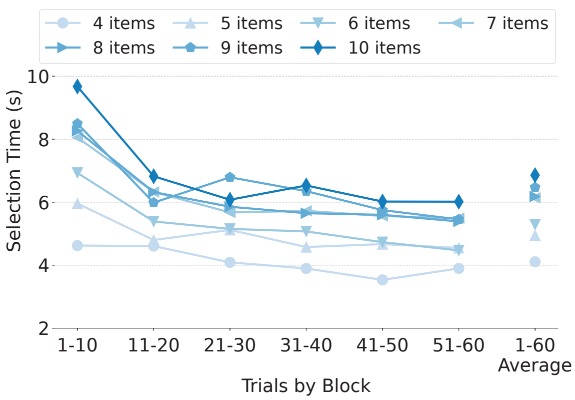


Figure 15: The predicted mean selection time by block of 4 to 10-item alphabetically ordered menus.

Additionally, we could use the model to decide the menu arrangement (Fig. 7). For example, both the study data and our model show that a grouped menu layout with 10 items resulted in a shorter completion time than both alphabetic and random layouts, indicating that a grouped layout is favorable for long menus (e.g., 10 items). We would also like to point out that the benefits of a grouped layout may not be generalizable to short menus (e.g., 6-item layout). The user study did not show a significant difference in selection time across different layouts for 6-item menus. The model could overestimate the benefits of grouped layouts for short menus (e.g., 6-item menus).

7.3 Limitations and Future Work

The presented computational model could be extended in future work. In the proposed model, we used the mixture pointing model [41] for the gliding action and the Fitts' Law [16] for direct touch action to estimate the mean time cost at different *ID*. Our results suggest that while the mixture pointing model fits the data well, Fitts' law does not. This could partly be explained by the fact that the mixture pointing model is designed for blind users, while Fitts' law is not. How to best estimate moving times of users who are blind remains an open research question.

In the future, it is interesting to investigate how the model would behave if certain assumptions are altered. For example, we assumed that the focused menu item position s_{focus} is fully observable to the agent. The s_{focus} is an important parameter for updating the position memory. However, due to the possible estimation noise from a user, the agent's knowledge on this parameter might come from partial observability. One interesting future direction is to investigate if introducing the probability distributions for parameters (e.g., s_{focus}) would improve the model performance.

8 CONCLUSION

We have proposed a computational model that simulates how blind users perform menu selection using swiping, gliding, direct touch and selection action in a linear menu using a screen reader. The model builds upon the theory of boundedly optimal control, which assumes the users' behavior emerge as an attempt to minimize the selection time in light of perceptual limits and memory of menu item positions. We formally modeled the menu selection problem as a stochastic sequential decision problem (Partially Observable Markov Decision Process). Partial observability stems from the reliance on limited memory and auditory feedback. The agent maintains a belief about the position of menu items, where the belief was computed based on a Gaussian position memory that we proposed, which considers the long-term interaction history. To then make predictions for a practical design-plus-user scenario, we trained a DQN in a simulated environment. We evaluated the model by comparing simulation results against empirical findings obtained in an IRB-approved user study with 10 legally blind users. The model correctly simulated the effect of menu length and menu arrangement on selection time as well as the percentage of action used by the users, and it also predicted the menu selection strategies of the users. This model advances our understanding of menu selection behavior for blind users, and could serve as a useful tool for accessible interface design, optimization, and evaluation.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments, and our user study participants. This work was supported by NIH award R01EY030085, NSF award 2113485, DoD award AL210015. AP and AO were supported by the Finnish Center for Artificial Intelligence (FCAI) and the Academy of Finland projects Human Automata (ID: 328813) and BAD (ID: 318559).

REFERENCES

[1] Carl Halladay Abraham, Bert Boadi-Kusi, Enyam Komla Amewuho Morny, and Prince Agyekum. 2022. Smartphone usage among people living with severe visual impairment and blindness. *Assistive Technology* 34, 5 (2022), 611–618.

[2] Robert St Amant, Thomas E Horton, and Frank E Ritter. 2007. Model-based evaluation of expert cell phone menu interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)* 14, 1 (2007), 1–es.

[3] Caroline Appert and Shumin Zhai. 2009. Using strokes as command shortcuts: cognitive benefits and toolkit support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Boston, MA, USA, 2289–2298.

[4] Apple. 2022. Vision Accessibility in iPhone. https://www.apple.com/accessibility/iphone/vision/

[5] Gilles Bailly, Antti Oulasvirta, Duncan P Brumby, and Andrew Howes. 2014. Model of visual search and selection time in linear menus. In *Proceedings of the sigchi conference on human factors in computing systems*. ACM, Toronto, Canada, 3865–3874.

[6] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. Menuoptimizer: Interactive optimization of menu systems. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, St. Andrews, UK, 331–342.

[7] Nikola Banovic, Tovi Grossman, and George Fitzmaurice. 2013. The effect of time-based cost of error in target-directed pointing tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris, France, 1373–1382.

[8] Richard E Bellman and Lotfi Asker Zadeh. 1970. Decision-making in a fuzzy environment. *Management science* 17, 4 (1970), B–141.

[9] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540* (2016).

[10] Duncan P Brumby, Samantha CE Davies, Christian P Janssen, and Justin J Grace. 2011. Fast or safe? How performance objectives determine modality output choices while interacting on the move. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, Vancouver, Canada, 473–482.

[11] Noshaba Cheema, Laura A Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämäläinen. 2020. Predicting mid-air interaction movements and fatigue using deep reinforcement learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu, HI, USA, 1–13.

[12] Xiuli Chen, Aditya Acharya, Antti Oulasvirta, and Andrew Howes. 2021. An adaptive model of gaze-based selection. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Online Virtual Conference, 1–11.

[13] Xiuli Chen, Gilles Bailly, Duncan P Brumby, Antti Oulasvirta, and Andrew Howes. 2015. 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, Seoul, Korea, 4217–4226.

[14] Xiuli Chen, Sandra Dorothee Starke, Chris Baber, and Andrew Howes. 2017. A cognitive model of how people make decisions through interaction with visual displays. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. ACM, Denver, CO, USA, 1205–1216.

[15] Andy Cockburn, Carl Gutwin, and Saul Greenberg. 2007. A predictive model of menu performance. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, San Jose, 627–636.

[16] Alistair DN Edwards. 1989. Modelling blind users' interactions with an auditory computer interface. *International journal of man-Machine studies* 30, 5 (1989), 575–589.

[17] Manahel El Lahib, Joe Tekli, and Youssef Bou Issa. 2018. Evaluating Fitts' law on vibrating touch-screen to improve visual data accessibility for blind users. *International Journal of Human-Computer Studies* 112 (2018), 16–27.

[18] Stephen R Ellis and Robert J Hitchcock. 1986. The emergence of Zipf's law: Spontaneous encoding optimization by users of a command language. *IEEE transactions on systems, man, and cybernetics* 16, 3 (1986), 423–427.

[19] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.

[20] Krzysztof Gajos and Daniel S Weld. 2004. SUPPLE: automatically generating user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces*. ACM, Funchal, Portugal, 93–100.

[21] Andrew Gelman and Jennifer Hill. 2006. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

[22] Samuel J Gershman, Eric J Horvitz, and Joshua B Tenenbaum. 2015. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science* 349, 6245 (2015), 273–278.

[23] Google. 2022. Accessibility – Touch and point targets. https://material.io/design/usability/accessibility.html#layout-and-typography

[24] Google. 2022. Android accessibility overview. https://support.google.com/accessibility/android/answer/6006564?hl=en

[25] Google. 2022. Talkback: Hear your screen read out loud. https://support.google.com/accessibility/android/topic/3529932

[26] William Grussenmeyer and Eelke Folmer. 2017. Accessible touchscreen technology for people with visual impairments: a survey. *ACM Transactions on Accessible Computing (TACCESS)* 9, 2 (2017), 1–31.

[27] Tiago Guerreiro, Paulo Lagoa, Hugo Nicolau, Daniel Gonalves, and Joaquim A. Jorge. 2008. From Tapping to Touching: Making Touch Screens Accessible to Blind Users. *IEEE MultiMedia* 15, 4 (2008), 48–50.

[28] Sean G Gustafson, Bernhard Rabe, and Patrick M Baudisch. 2013. Understanding palm-based imaginary interfaces: the role of visual and tactile cues when browsing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris, France, 889–898.

[29] Andrew Howes, Richard L Lewis, and Alonso Vera. 2009. Rational adaptation under task and processing constraints: implications for testing theories of cognition and action. *Psychological review* 116, 4 (2009), 717.

[30] Mohit Jain, Nirmalendu Diwakar, and Manohar Swaminathan. 2021. Smartphone usage by expert blind users. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Online Virtual Conference, 1–15.

[31] Edwin T Jaynes. 1957. Information theory and statistical mechanics. *Physical review* 106, 4 (1957), 620.

[32] Jussi Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. Touchscreen typing as optimal supervisory control. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Online Virtual Conference, 1–14.

[33] Jussi PP Jokinen and Tuomo Kujala. 2021. Modelling Drivers' Adaptation to Assistance Systems. In *13th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, Leeds, UK, 12–19.

[34] Jussi PP Jokinen, Tuomo Kujala, and Antti Oulasvirta. 2021. Multitasking in driving as optimal adaptation under uncertainty. *Human factors* 63, 8 (2021), 1324–1341.

[35] Jussi PP Jokinen, Zhenxin Wang, Sayan Sarcar, Antti Oulasvirta, and Xiangshi Ren. 2020. Adaptive feature guidance: Modelling visual search with graphical layouts. *International Journal of Human-Computer Studies* 136 (2020), 102376.

[36] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101, 1-2 (1998), 99–134.

[37] Shaun K. Kane, Jeffrey P. Bigham, and Jacob O. Wobbrock. 2008. Slide Rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-Touch Interaction Techniques. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility* (Halifax, Nova Scotia, Canada) *(Assets '08)*. Association for Computing Machinery, New York, NY, USA, 73–80.

[38] Antti Kangasrääsiö, Kumaripaba Athukorala, Andrew Howes, Jukka Corander, Samuel Kaski, and Antti Oulasvirta. 2017. Inferring cognitive models from data using approximate Bayesian computation. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. ACM, Denver, CO, USA, 1295–1306.

[39] Akif Khan and Shah Khusro. 2019. Blind-friendly user interfaces – a pilot study on improving the accessibility of touchscreen interfaces. *Multimedia Tools and Applications* 78 (07 2019), 17495–17519.

[40] Akif Khan and Shah Khusro. 2021. An insight into smartphone-based assistive solutions for visually impaired and blind people: issues, challenges and opportunities. *Universal Access in the Information Society* 20, 2 (2021), 265–298.

[41] Yu-Jung Ko, Aini Putkonen, Ali Selman Aydin, Shirin Feiz, Yuheng Wang, Vikas Ashok, IV Ramakrishnan, Antti Oulasvirta, and Xiaojun Bi. 2021. Modeling Gliding-based Target Selection for Blind Touchscreen Users. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*. ACM, Online Virtual Conference, 1–14.

[42] Stelios Lelis and Andrew Howes. 2011. Informing decisions: how people use online rating information to make choices. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, Vancouver, Canada, 2285–2294.

[43] Richard L Lewis, Andrew Howes, and Satinder Singh. 2014. Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in cognitive science* 6, 2 (2014), 279–311.

[44] Frank Chun Yat Li, David Dearman, and Khai N Truong. 2010. Leveraging proprioception to make mobile phones more accessible to users with visual impairments. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*. ACM, Orlando, FL, USA, 187–194.

[45] Zhi Li, Maozheng Zhao, Dibyendu Das, Hang Zhao, Yan Ma, Wanyu Liu, Michel Beaudouin-Lafon, Fusheng Wang, I.V. Ramakrishnan, and Xiaojun Bi. 2022. Select or Suggest? Reinforcement Learning-based Method for High-Accuracy Target Selection on Touchscreens. In *CHI '22: ACM CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans, LA, USA, 1–15.

[46] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[47] Wanyu Liu, Gilles Bailly, and Andrew Howes. 2017. Effects of frequency distribution on linear menu performance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, Denver, CO, USA, 1307–1312.

[48] I Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1 (1992), 91–139.

[49] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[50] Alistair Morrison, Xiaoyu Xiong, Matthew Higgs, Marek Bell, and Matthew Chalmers. 2018. A Large-Scale Study of iPhone App Launch Behaviour. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montréal, Canada, 1–13.

[51] Saied B. Nesbat. 2003. A System for Fast, Full-Text Entry for Small Electronic Devices. In *Proceedings of the 5th International Conference on Multimodal Interfaces* (Vancouver, Canada) *(ICMI '03)*. Association for Computing Machinery, New York, NY, USA, 4–11.

[52] Antti Oulasvirta, Jussi P.P. Jokinen, and Andrew Howes. 2022. Computational Rationality as a Theory of Interaction. In *CHI '22: ACM CHI Conference on Human Factors in Computing Systems,*. ACM, New Orleans, LA, USA, 1–14.

[53] Tomi Peltola, Jussi Jokinen, and Samuel Kaski. 2019. Probabilistic Formulation of the Take The Best Heuristic. *arXiv preprint arXiv:1911.00572* (2019).

[54] Rajesh PN Rao. 2010. Decision making under uncertainty: a neural model based on partially observable markov decision processes. *Frontiers in computational neuroscience* 4 (2010), 146.

[55] Stuart J Russell and Devika Subramanian. 1994. Provably bounded-optimal agents. *Journal of Artificial Intelligence Research* 2 (1994), 575–609.

[56] Sayan Sarcar, Jussi PP Jokinen, Antti Oulasvirta, Zhenxin Wang, Chaklam Silpasuwanchai, and Xiangshi Ren. 2018. Ability-based optimization of touchscreen interactions. *IEEE Pervasive Computing* 17, 1 (2018), 15–26.

[57] Suraj Singh Senjam, Souvik Manna, and Covadonga Bascaran. 2021. Smartphones-Based Assistive Technology: Accessibility Features and Apps for People with Visual Impairment, and its Usage, Challenges, and Usability Testing. *Clinical Optometry* 13 (2021), 311.

[58] Siti Rohkmah Mohd Shukri and Andrew Howes. 2019. Children adapt drawing actions to their own motor variability and to the motivational context of action. *International Journal of Human-Computer Studies* 130 (2019), 152–165.

[59] Benjamin L Somberg. 1986. A comparison of rule-based and positionally constant arrangements of computer menu items. *ACM SIGCHI Bulletin* 17, SI (1986), 255–260.

[60] Matthijs TJ Spaan. 2012. Partially observable Markov decision processes. In *Reinforcement Learning*. Springer, 387–414.

[61] Kashyap Todi, Gilles Bailly, Luis Leiva, and Antti Oulasvirta. 2021. *Adapting user interfaces with model-based reinforcement learning*. ACM, Online Virtual Conference, 1–13.

[62] Yuan-Chi Tseng and Andrew Howes. 2015. The adaptation of visual search to utility, ecology and design. *International Journal of Human-Computer Studies* 80 (2015), 45–55.

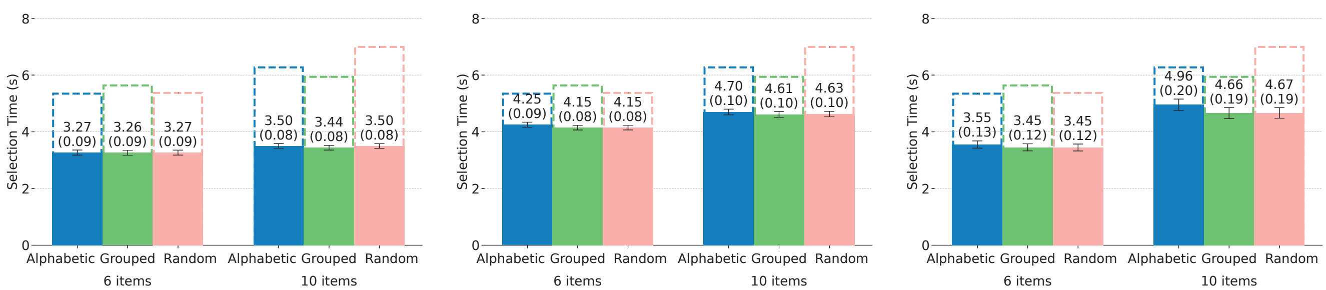
[63] Pavani Yalla and Bruce N Walker. 2008. Advanced auditory menus: Design and evaluation of auditory scroll bars. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*. ACM, Halifax, Canada, 105–112.

[64] Suwen Zhu, Yoonsang Kim, Jingjie Zheng, Jennifer Yi Luo, Ryan Qin, Liuping Wang, Xiangmin Fan, Feng Tian, and Xiaojun Bi. 2020. Using Bayes' Theorem for Command Input: Principle, Models, and Applications. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu, HI, USA, 1–15.

A APPENDIX

Action	Parameter	Training strategy	Absolute moving distance (cm)									
			0	1	2	3	4	5	6	7	8	9
Gliding	σ_G	One-group model	1.19	1.78	2.13	2.33	2.38	2.31	2.25	2.31	2.06	1.98
		Two-subgroup model (Subgroup A)	1.32	1.94	2.26	2.54	2.10	2.14	2.23	2.57	2.16	1.74
		Two-subgroup model (Subgroup B)	0.88	1.50	1.83	1.90	2.02	2.49	2.35	1.90	1.79	1.79
Direct touch	σ_D	One-group model	NA	1.16	1.21	1.07	1.13	1.06	1.08	1.03	1.26	1.12
		Two-subgroup model (Subgroup A)	NA	1.23	1.35	1.03	1.17	1.16	1.15	1.10	1.25	1.15
		Two-subgroup model (Subgroup B)	NA	1.02	0.85	1.08	0.98	0.88	0.90	0.60	1.22	0.70

Table 4: The observed standard deviation of the time cost (MT) of the gliding action and direct touch action at different absolute moving distance. The absolute moving distance 0 means that the user started the action and ended it at the same menu item. The "NA" means no user direct touched the current focused item.



(a) Fitts' law based method **(b) Gliding model based method** **(c) Swiping model based method**

Figure 16: The mean selection time (95% confidence interval) by condition predicted by the input action model based methods. The dashed bars are the means of the observed selection time in the user study.