Adaptive Enhancement of Swipe Manipulations on Touch Screens with Content-awareness

Yosuke Fukuchi, Yusuke Takimoto and Michita Imai
Keio University, Yokohama, Japan
{*fukuchi, takimoto, michita*}*@ailab.ics.keio.ac.jp*

Keywords: Swipe, Intelligent UI, Proactive UI, Touch Screen, Human-computer Interaction.

Abstract: Most user interfaces (UIs) do not consider user intentions behind their manipulations and show only fixed responses. UIs could contribute to more effective work if they are made so as to infer user intentions and to respond proactively to the operations. This paper focuses on users' swipe gestures on touch screens, and it proposes IntelliSwipe, which adaptively adjusts the scroll amount of swipe gestures based on the inference of user intentions, that is, what they want to see. The remarkable function of IntelliSwipe is to judge user intentions by considering visual features of the content that users are seeing while previous studies have only focused on the mapping from user operations to their intentions. We implemented IntelliSwipe on an Android tablet. A case study revealed that IntelliSwipe enabled users to scroll a page to the proper position just by swiping.

1 INTRODUCTION

This paper addresses the problem of "how a UI should respond to user input." Most UIs provide a *reactive* response; they are unaware of what users want to do and show only fixed responses towards their input. However, a *proactive* UI (PUI) infers the intentions behind users' manipulations and adaptively changes its behavior to help the user achieve her or his goals. A PUI has the potential to require fewer user operations and to enable users to execute their tasks in a more efficient way.

Few studies have addressed the problem of generating a UI's proactive responses. Although many intelligent UIs (IUIs) show adaptive behavior based on the inferences of what a user wants, most of these approaches focus on recommendations, that is, what a UI should show before a user does an operation. For example, methods to adapt menus or content so that users can quickly access what they want are actively being proposed (Soh et al., 2017; Gobert et al., 2019; Todi et al., 2018).

Previous studies suggest that UIs can proactively contribute to user operations by considering the user intentions and adapting the response. Fowler et al. proposed a touch-screen keyboard that infers what word a user intends to type using a language model personalized to the user (Fowler et al., 2015). The keyboard could correct users' mistypes due to imprecise tapping and could reduce the word error rate. Kwok et al. focused on mouse operations (Kwok et al., 2018). They proposed a model that predicts a user's next interaction using past mouse movements, and they applied the model to the detection of non-intentional clicks. Delphian desktop (Asano et al., 2005) more proactively intervenes user operations to enhance mouse-based interaction. It predicts the goal of a moving cursor based on its velocity and makes the cursor "jump" to the most probable icon that a user intends to point. The jumping cursor succesfully decreased users' time to point their targets.

Figure 1: A user intends to skip an advertisement on a website. When the user swipes, IntelliSwipe adjusts the amount of scrolling based on the inference of user intentions.

Previous studies have allowed users to reduce time and the number of user operations, but there is still room for further research. One problem is that the

Fukuchi, Y., Takimoto, Y. and Imai, M.
Adaptive Enhancement of Swipe Manipulations on Touch Screens with Content-awareness.
DOI: 10.5220/0008917804290436
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 2*, pages 429-436
ISBN: 978-989-758-395-7; ISSN: 2184-433X

inference of user intentions was based mainly on a user's ongoing operations and not able to be applied on the situation in which a user formulated an intention. Suppose that you find an advertisement on a website and intend to skip it. You can only see the head of the advertisement first, so you do not exactly know how much you should scroll to get past it. Then, you try scrolling a little, check the screen, find that the advertisement still continues, and scroll again. You repeat this loop until you find the tail of the advertisement. Here, your operations are not connected directly with the original goal to skip the advertisement but with the subgoal of checking succeeding content. Because even the users do not know how much they should scroll, a system cannot infer the final target of scrolling if it only focuses on user operations.

In this paper, we propose IntelliSwipe (Fig. 1), which enables touch-screen systems to respond adaptively towards users' swipe gestures considering the user intentions, or what they want to see (or skip). IntelliSwipe acquires users' manipulation history and learns to infer user intentions based on visual features of the content. When a user swipes, IntelliSwipe successively evaluates the content on the display and tries to adjust the scrolling to the desired position. Users just need to swipe once to scroll if IntelliSwipe properly infers their intentions.

We implemented IntelliSwipe on an Android tablet and conducted a case study. Participants were requested to work on a web-based task that we prepared. We trained IntelliSwipe with the manipulation history and examined the learned behavior. The results showed that IntelliSwipe enabled users to scroll a page to the proper position when a user swiped.

This paper is structured as follows. Section 2 presents related work and the background of IntelliSwipe. Section 3 proposes IntelliSwipe and describes how it enhances users' swipe operations. Section 4 explains a case study to evaluate IntelliSwipe and discusses the learned behavior. Section 5 describes the limitations and the future directions of the proactive touch-screen UI. Section 6 concludes the paper.

2 BACKGROUND

2.1 Designing Manipulation on Touch Screen

Touch screens are widely accepted in our daily life because of their usability. Many designers have tried to define the correspondence between a user's operation and their effects for more natural and intuitive manipulations (Malik et al., 2005; Wu et al., 2006).

One measure of well-defined gestures is the similarity of the effects on the operation of physical objects. When you slide your finger on a touch screen, the content on the screen follows the movement of your finger as if you are touching a paper document. Such realistic behavior improves the predictability of the system's responses to user's manipulations and reduces the gulf of execution, or the gap between a user's goal and the allowable actions (Norman, 2013).

However, realistic behavior can also be a restriction. If you want a large amount of scrolling or movement, you need to repeat the gesture many times. Proactive UI has the potential to address this problem. Though it may also increase the unpredictability of the system (Paymans et al., 2004), we can expect that a UI can reduce the number of operations and cognitive costs by inferring the intentions and proactively supporting them.

2.2 Intelligent User Interface

IUIs attempt to improve human-computer interactions using adaptive behavior based on the models of users and the environment (Wahlster and Maybury, 1998). IUIs have addressed problems such as personalization, information filtering, and splitting the user's job (Alvarez-Cortes et al., 2007).

Especially with introducing machine learning, data-driven IUIs should be able to deal autonomously with countless possible contexts based on actual usage history data and to exceed a human designer's imagination. Utilization of a user's usage history has been attempted for enabling UIs to adapt such as by self-adapting menus on a website (Gobert et al., 2019) and by designing application forms (Rahman and Nandi, 2019).

Some studies have approached the problem of a UI's proactive responses when user input is provided. They enabled UIs to correct user misoperations (Kwok et al., 2018; Fowler et al., 2015) or to provide shortcuts to their targets (Asano et al., 2005) by inferring user intentions behind their manipulations and changing the responses adaptively.

We argue that previous studies cannot be applied to users' complex operations. For explanation, we introduce Norman's model of how users interact with systems (Norman, 2013). The model is broadly divided into two processes: execution and evaluation (Fig. 2). In the execution process, users are considered to choose a plan to achieve their goals, specify an action sequence for the plan, and perform it. Through these actions, the status of the system changes. Then, users perceive the responses from the system, interpret them, and evaluate the current situation. Users may reformulate their goals based on the results of their operation and proceed to the execution process again until they achieve their goals.

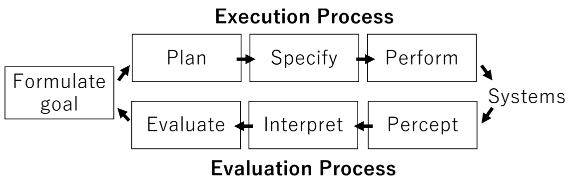


Figure 2: Norman's seven-stage model of action while users are operating a system.

Proactive responses in previous studies were completed in the execution process. They mapped user operations being performed to their goals and intervened with the operations for the inferred goals in the performance stage. However, they were not capable of recognizing or evaluating the situation. Although this approach works when we focus on low-level goals for which users do not have to repeat the execution-evaluation loop, it is difficult to support users' higher-level goals that require them to repeat the loop and to set subgoals depending on the situation. Achieving higher-level goals requires longer time and more cognitive loads, so systems will contribute to reducing time and the number of operations more if they can deal with users' high-level goals. Moreover, user intentions may differ even if they show the same operations. We can expect that systems can interpret an operation more precisely by considering the situation in which the operation was performed.

In addition, previous studies have focused on the problem of predicting users' targets from limited number of candidates. Thanks to the spread of touch screens, we perform more continuous operations such as scrolling, swiping and pinching every day, but little attention has been paid to adapting them.

2.3 Shared Autonomy

Shared autonomy is a concept in the field of robotics, one that aims to enable effective robot teleoperation by combining user inputs with autonomous assistance (Ferrell and Sheridan, 1967; Michelman and Allen, 1994; Seno et al., 2018). Users sometimes feel manipulating robots is difficult due to the limitations of manipulation interfaces and the complexity of robots, especially a high degree of freedom. Shared autonomy enables users to operate robots in such situations by inferring the intention of operators from their input and adaptively assisting the accomplishment of their task.

Research on shared autonomy has shown the potential of proactive responses in human-computer interactions. However, the application of shared autonomy is limited to embodied agents, and applicability to UI remains an open question.

2.4 Swipes

Swipes involve a gesture to move one's finger quickly across a touch screen. In common swipe implementations, a page is scrolled along as the finger moves while one's finger is on the screen, and the scrolling is gradually decelerated after the finger has left. It is called inertia scrolling or momentum scrolling. The amount of scrolling using swipe operations depends only on the speed of the finger.

3 INTELLISWIPE

3.1 Approach

IntelliSwipe enhances users' page scrolling by adjusting the scroll amount when they swipe. IntelliSwipe aims to skip the content that users are not interested in and stop at the desired position for the users, while typical UIs just show inertia scrolling for swipes regardless of the content or the user intentions.

The key idea of IntelliSwipe is that users' scroll amount reflects their intentions to see or skip what is being displayed. That is, we assume that users show long scrolling when they want to skip the content and scroll finely when there is something interesting for them. IntelliSwipe acquires the usage history of how much users scroll in various situations and learns the relationship between their scroll amount and the content displayed at that time.

Figure 3 shows the overview of the scroll controlling process in IntelliSwipe. While a user is operating, IntelliSwipe continuously captures the screen image shown to the user. Its prediction model predicts the amount of scrolling based on the captured images. IntelliSwipe adjusts the destination of scrolling to the position in which the user is predicted to stop scrolling.

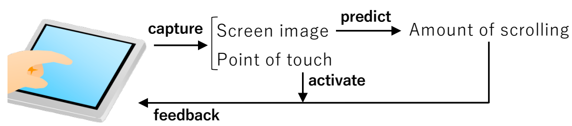


Figure 3: The information flow of IntelliSwipe.

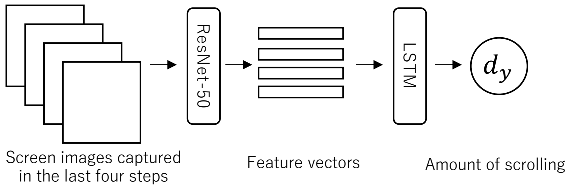


Figure 4: The network structure of the prediction model.

Algorithm 1: IntelliSwipe.

h: a queue for captured screen images.
m_1: threshold of d_y to stop scrolling.
m_2: the upper limit of d_y.
while in drawing loop **do**
  Capture the current screen image and add to h.
  if detect swipe gesture **then**
    SWIPING $\leftarrow$ true.
  end if
  if SWIPING **then**
    Extract the screen images in the last four steps from h.
    Predict the amount of scrolling d_y.
    if $d_y > m_2$ **then**
      $d_y \leftarrow m_2$
    end if
    if $d_y > m_1$ **then**
      Scroll the page by $k \cdot d_y$.
    else
      SWIPING $\leftarrow$ false.
    end if
  end if
end while

3.2 Implementation

Algorithm 1 shows how IntelliSwipe works in our implementation. In this paper, we only present dealing with one-dimensional vertical scrolling (y-axis), but this can be applied to other dimensions in principle. When a user moves their finger and leaves it at a certain speed, IntelliSwipe detects it as a swipe gesture and starts controlling the UI's behavior. The page keeps scrolling while the prediction model predicts that users will scroll further, and it stops when the predicted amount of scrolling becomes less than a certain threshold m_1. m_2 is the upper limit of d_y. k is a discount rate to avoid rapid movements and achieve gradual deceleration similar to inertia scrolling.

Figure 4 shows the structure of the prediction model. The network is composed of ResNet-50 (He et al., 2016), which is trained to extract the features of the input images, and LSTM (Hochreiter and Schmidhuber, 1997), a deep-learning model for time series information. The input of the prediction model is the screen images captured in the last four steps. The model is expected to predict the scroll amount based on the visual features of the content and the speed and acceleration of the user's past operations.

4 CASE STUDY

4.1 Overview

A case study was conducted to investigate the behavior of IntelliSwipe. We first collected users' usage history in a task that we prepared. The collected data were used to train the prediction model. Then, we applied IntelliSwipe with the model to three situations and analyzed the results.

4.2 Apparatus

We deployed an Android tablet (Huawai MediaPad M5), which had an 8.4-inch touch screen (1600 x 2560 pixels). It was used in portrait orientation. The inertia scrolling was disabled when we collected the training data so as to clarify what participants wanted to see or skip. The participants needed to repeat scrolling when they wanted to skip content.

Because the calculation cost of the prediction model was heavy for the tablet, we prepared a server for the calculation. The tablet continuously sent screen images, and the server renewed d_y at 36 Hz on average. This design requires a certain amout of communication traffic but will be realistic with the spread of 5G networks (Gupta and Jha, 2015).

4.3 Task

We prepared a task in which participants were asked questions in the Japanese language (Tsutsui et al., 2010a; Tsutsui et al., 2010b). The participants worked on the task on web-based applications.

The task was composed of two parts: exercises in grammar and reading comprehension. They were asked in this order. In the grammar exercise page (Fig. 5), grammatical instruction sections and question sections were repeated six times. The mean length of the instruction sections was 8,059 pixels (SD = 865 pixels), so the participants needed to repeat scrolling to reach the question sections. There were eight questions for each instruction section. The participants were requested to answer questions from four choices. The reading comprehension page (Fig. 6) had the same structure as the grammar page, but instruction sections were replaced with

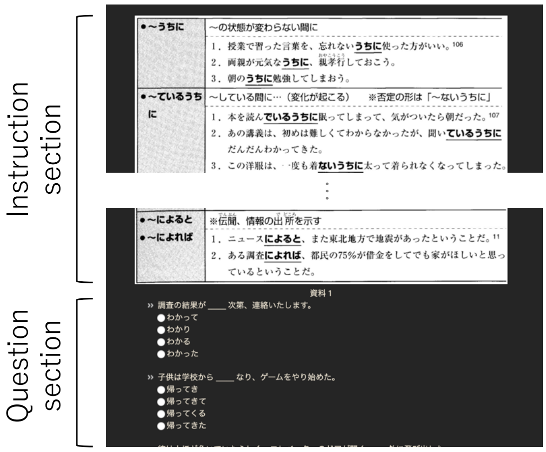


Figure 5: A part of the grammar exercise page. Instruction sections gave example sentences for Japanese words and idioms. A question section has eight questions. The participants gave answers by selecting radio buttons.

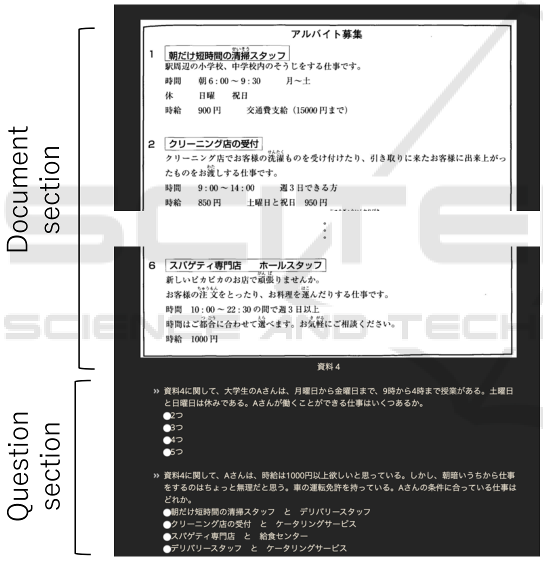


Figure 6: A part of the reading comprehension page. One or two questions were provided for each document. The participants needed to refer to the document to give the correct answer.

documents (short stories or flyers). The participants were asked to answer questions about the documents. The participants needed to read the documents to give the correct answer contrary to the grammatical instructions which participants did not necessarily have to read if they had sufficient knowledge of Japanese grammar. The mean length of document sections was 2,017 pixels (SD = 362 pixels). The number of questions was one or two for each document.

4.4 Data Acquisition

Participants were four male undergraduate or graduate students majoring in computer science aged 21 to 28 years old (Mean = 24, SD = 2.5). All of them were native Japanese speakers who could solve the grammar exercises without instructions.

During the task, all the participants put the tablet on a desk to manipulate because it was large to hold by hand. Participants read the grammatical instructions in the beginning of the task but gradually began to skip them. In the reading comprehension page, they went back and forth between a document and questions to give right answers.

The data were captured at 64 Hz on average. The total number of the samples was 230,984.

4.5 Training

We trained the prediction model with the users' usage history data acquired in subsection 4.4. The prediction model was trained to predict the amount of scrolling based on the screen images in the last four steps. The amount of scrolling at time t was defined as the difference in the y coordinates between the current touch point y_t and that of the final point where the user left her or his finger y_{t^*}.

$$d_{y,t} = y_{t^*} - y_t, \tag{1}$$

From the data, we removed the samples that were acquired when the participants were not scrolling, and 42,809 samples remained for the training. Figure 7 shows the loss on both the training and validation datasets. After 50 epochs, the loss decreased to 0.00045, which is equivalent to 2.0 % of the height of the screen.

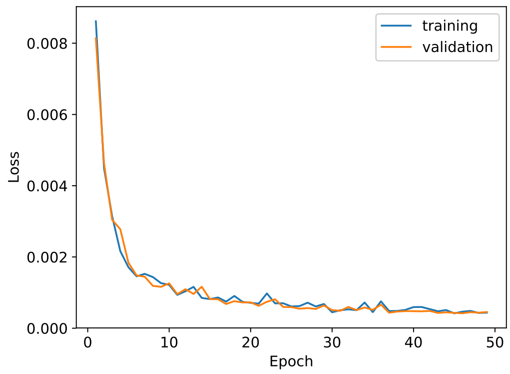
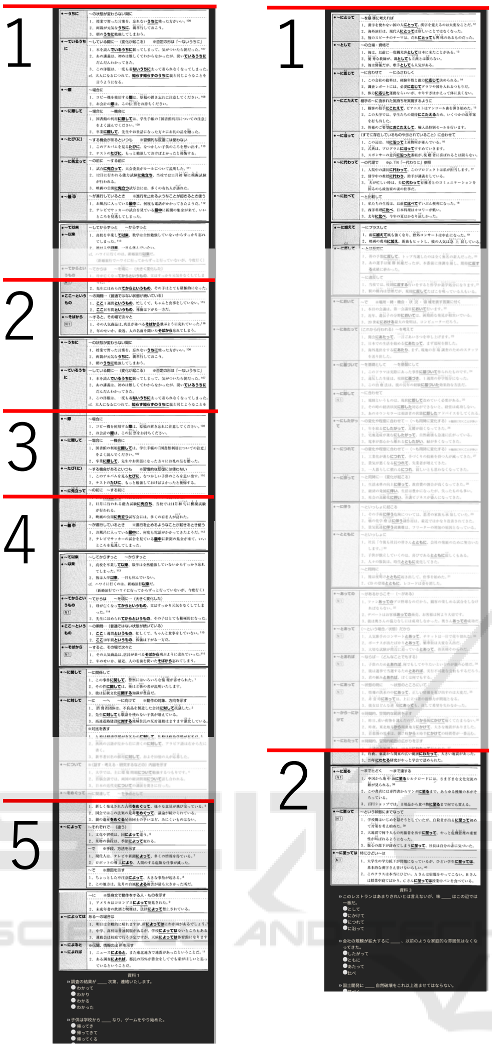


Figure 7: A plot of the losses on training and validation datasets for 50 epochs.



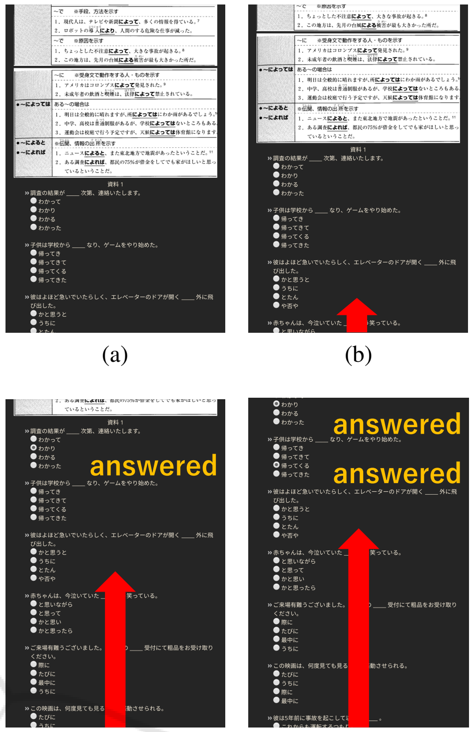
(a) (b)

Figure 8: The observed behavior in situation 1. The red lines indicate the top boundaries of the display range after IntelliSwipe stopped scrolling. In the first instruction section, IntelliSwipe displayed short scrollings when a swipe was provided (left), whereas it completely skipped the other instruction sections (right). In both sections, IntelliSwipe stopped the scrolling when the question sections appeared.

4.6 Analysis of IntelliSwipe's Behavior

With the prediction model acquired in subsection 4.4 and 4.5, we analyzed the behavior of IntelliSwipe towards a user's swipe operations. We prepared three situations that were likely to occur while people were working on the task. In this analysis, swiping means the successive movement in four steps at the speed of 50 pixels per step.

Situation 1 started from the tops of the instruction sections in the grammar page, and we repeated swiping until the following question sections appeared. The scrolling behavior was different between the first instruction section and the others. Figure 8 compares them. In the first instruction section (Fig. 8a), the amount of scrolling by one swipe gesture was shorter than it was in the other sections, where IntelliSwipe completely skipped the content (Fig. 8b). These results reflected the participants' actual behavior in the data acquisition. They tended to scroll shortly in the first instruction section to check the content but gradually noticed that they did not have to read it and skipped the later ones. Therefore, we can say that this behavior meets the participants' intentions towards the instruction sections. IntelliSwipe always stopped scrolling when the following question section appeared, and that was also reasonable for the participants who intended to answer all the questions.



(a) (b) (c) (d)

Figure 9: The results in situation 2. The length of the red arrows corresponds to the amount of scrolling by IntelliSwipe. The amount of scrolling inscreased as more answers were given. (a) Initial position. (b) With no answer. (c) The first question was answered. (d) The first two questions were answered.

In situation 2, we investigated the behavior in a question section. We compared the difference in scrolling behavior resulting from the progress states of answering, that is, how many answers were given. Figure 9 shows the results. IntelliSwipe stopped scrolling soon when no answer was selected, and the

(a) (b) (c) (d)

Figure 10: The behavior in situation 3. (a) Initial position. (b) The behavior when the user scrolled down from the intial position. (c) When scrolling up after b occurred. (d) Scrolling down from the position of c. Here, the first question had already been answered.

amount of scrolling increased as more answers were given. IntelliSwipe enabled users to skip questions that had already been answered and led them to unanswered questions.

Situation 3 focused on the reading comprehension page. We swiped back and forth between a document section and a question section. Figure 10 illustrates the transitions. At the initial position, we can see the whole document section, but the first question was cut off in the middle. We swiped to scroll down from here, and IntelliSwipe stopped the scrolling at the end of the first question. Then, we swiped back, and returned to the position in which the whole document section appeared. Here, we answered the first question and swiped again to scroll down. As a result, IntelliSwipe could stop scrolling when the whole second question appeared to be contrary to the behavior when we swiped first without answering (Fig. 10b).

These results indicate that IntelliSwipe could successfully adapt to user operations. It learned user intentions in various situations from the usage history, and the behavior seemed to assist the accomplishment of this task.

5 LIMITATIONS AND FUTURE WORKS

The results of the case study indicated the potential of IntelliSwipe, but challenges remain to apply IntelliSwipe in actual situations.

One limitation of this study is the assumption that what users want to see or skip is similar. The prediction model may not work if a wide variety of people use a UI with various intentions. For example, what IntelliSwipe acquired in the case study will not work for a user who is not willing to choose the right answer because all the participants in the data acquisition phase seriously made an effort for the task. At least two approaches are possible for this problem. One direction is to collect a large amount of usage history from many users so that the training data can cover any kind of intention that users have. Another way is to train the prediction model with the data only from the target user and personalize IntelliSwipe. We also need to examine whether or not screen images in the last four steps are enough to infer users' intentions. Using not only visual features but also linguistic information on content is a promising approach.

In addition, the usability of IntelliSwipe should be investigated further. Adaptive behavior sometimes decreases the predictability of the systems, leading to distrust (Antifakos et al., 2005). We need a long-term user study to evaluate the learnability of IntelliSwipe (Paymans et al., 2004).

6 CONCLUSION

In this paper, we proposed IntelliSwipe, which enables touch-sensitive UIs to respond to a user's swipe operations proactively by considering the user intentions behind her or his manipulations. IntelliSwipe learns what the users want to see (or skip) based on visual features of the content that they watch and enhances their swipe operations by adjusting the scroll amount to the desireable position. In a case study, we applied IntelliSwipe to a web-based task in which users solved exercises for Japanese language examination. We collected users' usage history on the task and analyzed the behavior acquired with the data. The results showed that IntelliSwipe could adaptively scroll pages to the desired positions for the users.

ACKNOWLEDGMENT

This work was supported by the New Energy and Industry Technology Development Organization

(NEDO).

REFERENCES

Alvarez-Cortes, V., Zayas-Perez, B. E., Zarate-Silva, V. H., and Ramirez Uresti, J. A. (2007). Current trends in adaptive user interfaces: Challenges and applications. In *Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007)*, pages 312–317.

Antifakos, S., Kern, N., Schiele, B., and Schwaninger, A. (2005). Towards improving trust in context-aware systems by displaying system confidence. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices &Amp; Services*, MobileHCI '05, pages 9–14, New York, NY, USA. ACM.

Asano, T., Sharlin, E., Kitamura, Y., Takashima, K., and Kishino, F. (2005). Predictive interaction using the delphian desktop. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 133–141, New York, NY, USA. ACM.

Ferrell, W. R. and Sheridan, T. B. (1967). Supervisory control of remote manipulation. *IEEE Spectrum*, 4(10):81–88.

Fowler, A., Partridge, K., Chelba, C., Bi, X., Ouyang, T., and Zhai, S. (2015). Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 649–658, New York, NY, USA. ACM.

Gobert, C., Todi, K., Bailly, G., and Oulasvirta, A. (2019). Sam: A modular framework for self-adapting web menus. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 481–484, New York, NY, USA. ACM.

Gupta, A. and Jha, R. K. (2015). A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, 3:1206–1232.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Kwok, T. C., Fu, E. Y., Wu, E. Y., Huang, M. X., Ngai, G., and Leong, H.-V. (2018). Every little movement has a meaning of its own: Using past mouse movements to predict the next interaction. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 397–401, New York, NY, USA. ACM.

Malik, S., Ranjan, A., and Balakrishnan, R. (2005). Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pages 43–52, New York, NY, USA. ACM.

Michelman, P. and Allen, P. (1994). Shared autonomy in a robot hand teleoperation system. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, volume 1, pages 253–259 vol.1.

Norman, D. (2013). *The Design of Everyday Things: Revised and Expanded Edition.* Basic Books, New York.

Paymans, T. F., Lindenberg, J., and Neerincx, M. (2004). Usability trade-offs for adaptive user interfaces: Ease of use and learnability. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, IUI '04, pages 301–303, New York, NY, USA. ACM.

Rahman, P. and Nandi, A. (2019). Transformer: A database-driven approach to generating forms for constrained interaction. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 485–496, New York, NY, USA. ACM.

Seno, T., Okuoka, K., Osawa, M., and Imai, M. (2018). Adaptive semi-autonomous agents via episodic control. In *Proceedings of the 6th International Conference on Human-Agent Interaction*, HAI '18, pages 377–379, New York, NY, USA. ACM.

Soh, H., Sanner, S., White, M., and Jamieson, G. (2017). Deep sequential recommendation for personalized adaptive user interfaces. In *Proceedings of the 22Nd International Conference on Intelligent User Interfaces*, IUI '17, pages 589–593, New York, NY, USA. ACM.

Todi, K., Jokinen, J., Luyten, K., and Oulasvirta, A. (2018). Familiarisation: Restructuring layouts with visual learning models. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, pages 547–558, New York, NY, USA. ACM.

Tsutsui, Y., Omura, R., and Kita, T. (2010a). *Japanese-language proficiency test N1, N2: grammar in 40 days (in Japanese).* Kirihara Shoten.

Tsutsui, Y., Omura, R., and Kita, T. (2010b). *Japanese-language proficiency test N1, N2: reading comprehension in 40 days (in Japanese).* Kirihara Shoten.

Wahlster, W. and Maybury, M. T. (1998). *Readings in Intelligent User Interfaces.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wu, M., Chia Shen, Ryall, K., Forlines, C., and Balakrishnan, R. (2006). Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '06)*, page 8.