# *DesignWatch*: Analyzing Users' Operations of Mobile Apps Based on Screen Recordings

Xiucheng Zhang
Sun Yat-sen University
Zhuhai, China
zhangxch58@mail2.sysu.edu.cn

Yixin Zeng*
Qichang Li*
Sun Yat-sen University
Zhuhai, China
zengyx53@mail2.sysu.edu.cn
liqch33@mail2.sysu.edu.cn

Guanyi Chen
Sun Yat-sen University
Guangzhou, China
chengy259@mail2.sysu.edu.cn

Qianyao Xu
Tsinghua University
Beijing, China
xuqy@mail.tsinghua.edu.cn

Xiaozhu Hu
Hong Kong University of Science and Technology
Hong Kong, China
huxz19@tsinghua.org.cn

Zhenhui Peng†
Sun Yat-sen University
Zhuhai, China
pengzhh29@mail.sysu.edu.cn

## ABSTRACT

Screen recordings of users' operations to complete tasks in the mobile app are vital resources for designers to assess the app's usability. However, analyzing these recordings at a large scale could be mentally challenging. In this paper, we present *DesignWatch*, which assists designers in analyzing users' operations of mobile apps based on collected screen recordings. *DesignWatch* supports interactive visual analyses of multiple users' operation paths in the app and prompts GPT-4 with vision to simulate users' thoughts during each operation. We conduct expert interviews with four designers, which highlight *DesignWatch*'s usefulness in helping them quickly understand users' operation patterns in the app, identify the potentially problematic UI design page, and get insights for improving the app design. We conclude with design implications for facilitating usability tests with interactive visualization and generative models.

## CCS CONCEPTS

• **Human-centered computing** → **Usability testing**; **Usability testing**; *User interface design*; *Graph drawings*; **User interface design**; *Graph drawings*; • **Computing methodologies** → *Natural language generation*; *Natural language generation.*

## KEYWORDS

Methods and Tools; Usability Test; User Interface Design; Visualization

*Both authors contributed equally to this research.
†The Corresponding author


*MOBILEHCI Adjunct '24, September 30–October 03, 2024, Melbourne, VIC, Australia*

ACM ISBN 979-8-4007-0506-9/24/09
https://doi.org/10.1145/3640471.3680231

**ACM Reference Format:**
Xiucheng Zhang, Yixin Zeng, Qichang Li, Guanyi Chen, Qianyao Xu, Xiaozhu Hu, and Zhenhui Peng. 2024. *DesignWatch*: Analyzing Users' Operations of Mobile Apps Based on Screen Recordings. In *26th International Conference on Mobile Human-Computer Interaction (MOBILEHCI Adjunct '24), September 30–October 03, 2024, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3640471.3680231

## 1 INTRODUCTION

Usability test, also known as user testing, is a crucial pathway that helps designers identify issues with their products, optimize their design, and understand user preferences [1]. In a mobile app usability test, designers usually prepare tasks for users and record their behaviours and perceptions during the task execution process [10, 12, 19]. During this process, screen recording captures visual changes on the screen, reflecting the user's operational procedures. This makes it a valuable analytical resource in usability tests.

However, it is difficult for designers to review the screen recordings of users at a large scale to evaluate the usability of their apps. On one hand, summarizing and organizing user operation paths from videos is time-consuming. Designers have to go through each recording video one by one, extract the operation path, make notes, and summarize the usability issues of their apps after analyzing many screen recordings. Visualization techniques could help to address this difficulty by enhancing the comprehensibility of a bunch of data samples and accelerating problem identification [14, 20, 23]. Nevertheless, few works have explored the design and usage of visualization techniques for facilitating usability tests based on screen recordings. On the other hand, it can sometimes be difficult to discern specific user actions on the screen from videos alone [4]. For instance, researchers must meticulously observe the animations of button clicks to determine which button was pressed that triggered a screen transition. Some studies supplement screen recordings with additional data collection such as event logs [5, 13] and IMU sensor [7] information to meet the needs of further analysis. However, this can complicate data collection, *i.e.,* making it difficult for users to operate independently in a remote setting. Multimodal

[1]https://www.nngroup.com/articles/usability-testing-101/

Large Language Models (MLLMs) [22] enhance the advanced visual comprehension capabilities of Large Language Models (LLMs), enabling them to integrate visual information for reasoning within specific contexts [16, 17]. Nevertheless, it is under-explored how designers perceive the usefulness of simulated user's thoughts by MLLMs in the usability test based on the screen recordings.

To this end, we design, develop, and evaluate an interactive tool named *DesignWatch* to help designers analyze user interactions within mobile app tasks based on screen recordings. In *DesignWatch*, designers can analyze users' operation patterns in an interactive directed graph, in which a clickable node represents a user interface (UI) page in the path and a clickable link denotes a transition between two pages. Designers can also view the simulated user thoughts of operations for each page transition. To visualize users' operation paths, we use a pre-trained ResNet deep learning neural network [6] to represent frames as vectors and calculate their similarity, thereby extracting key frames and matching multiple operation paths. To inspire GPT-4 with vision (GPT-4V) [21] to simulate user thoughts of operations during page transitions, we collect human users' thoughts along with the screen recordings in the example apps and incorporate them into the prompt.

We evaluate *DesignWatch*'s usefulness and user experience with four designers. Our designers provide cases in which they use *DesignWatch* to understand users' operation patterns, identify potentially problematic UI design pages, and gain insights for improving the app design. designers highlight the *DesignWatch*'s usefulness in assisting them in analyzing screen recordings of user operations for mobile app usability tests. Based on our findings, we discuss how visualization and multi-modal LLMs can be used to facilitate usability tests.

## 2 DESIGN AND IMPLEMENTATION OF *DESIGNWATCH*

### 2.1 Design Process

We work with two design experts (E1 and E2) to develop *DesignWatch*. E1 has over 8 years of experience in design projects and is a postdoctoral researcher in the field of Computational Aesthetics and Human-Computer Interaction (HCI). E2 majored in design and is responsible for the design and marketing of the products in a startup company. We conduct semi-structured interviews with E1 and E2 separately via VooV Meeting, asking about 1) the need for designers to collect and understand screen recordings during the usability test process, 2) the challenges faced by designers in this process, especially in analyzing screen recordings, and 3) the expectations and requirements for a tool to assist analyses of screen recordings. Each interview lasts for about 50 minutes, and we give the expert about 27 USD as compensation.

Both experts confirm the value of collection and analysis of screen recordings in usability tests. First, screen recording is the main method of documenting the user operation process. This means that designers need to review these videos to understand the specific actions taken by users, and the pages they navigated through, and thereby gather information on how users completed tasks. Second, based on the information about users' operation paths, designers can gain further insights and suggestions for improving their apps. E2 provided an example, *"When designers notice from the screen recordings that a majority of users misinterpret the same UI page during the task completion process, they know there must be something wrong in that page and need to revise the design of the page to guide users complete the task."*

Despite the benefits, both designers raise two challenges in analyzing these recorded videos of users' operations. First, watching each video and extracting information from it are time-consuming. E1 points out, *"During a usability test, users focus on how to successfully operate the app. Their operations are affected by many factors, such as their perceptions of the UI design, their ability, and their habits of using the phone. Therefore, there could be some quick clicks and random swipes in any user's operations. Designers need to carefully go through the frames of each video to identify the operation path from the video"*. Second, it is hard to summarize the patterns of users' operations in the app. E1 states, *" It is hard to get overall impressions on the general operation patterns of all users by reviewing the videos one by one does not. Designers often need to organize their learned information from the usability tests to observe the patterns "*. E2 also expresses a similar viewpoint, emphasizing, *"We often recruit multiple users to complete the same target task in the app. The common operation patterns among multiple users are more valuable for reference, but it is time-consuming to obtain these patterns"*.

### 2.2 User Interface

To address these two challenges, we present *DesignWatch*, an interactive tool that facilitates designers to diagnose the usability of mobile apps based on screen recordings of users' operations. *DesignWatch* takes the following information as input: 1) Textual **description of an interaction task** that the designer requires the user to complete in the app, *e.g., "Check my ranking in the user group"*. 2) **The expected user operation path** to complete the task, which is represented by a series of ordered UI screenshots. 3) **A set of screen recordings** that capture how each user operates the app to complete the assigned task. 4) Optionally, a file of user background information that could be used to customize the analysis of different groups of users. As shown in Figure 1, *DesignWatch* contains an interactive directed graph that visualizes the operation path of all users and allows designers to click each link in the graph to get the inference of the detailed user's operation. We will describe the interaction with *DesignWatch* via a case presented in section 3.

### 2.3 Visualization of Users' Operation Paths

Figure 2 (a) shows our approach to extract and visualize users' operation paths from screen recordings.

*2.3.1 Similarity.* Different from natural scene videos, UI videos have clear shot boundaries of different interfaces, *i.e.,* the start and end frames of a fully rendered UI. Following the approach of [4], to detect shots, we attempt to calculate a similarity score for consecutive frame comparisons. We employ a ResNet-18 model [6] that was pre-trained on the ImageNet dataset [3] (hereafter referred to as ResNet-18-ImageNet). We leverage this model's feature extraction capabilities to transform input images into 512-dimensional feature vectors. This transformation process involves passing the images through multiple convolutional layers, activation functions, and pooling layers of the ResNet-18 model until

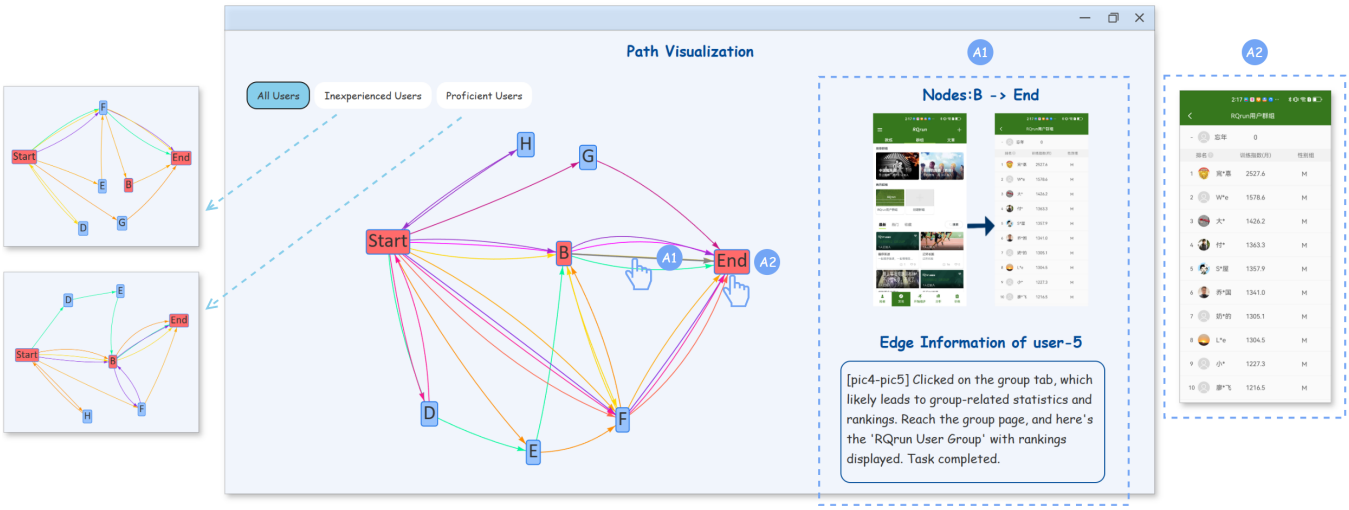



**Figure 1: Interactive directed graph of all users' operation paths. Designers can click each node (A2) to view a UI page of the app and click each link (A1) to view the transition between two UI pages and LLM-simulated user's thoughts on this transition. "All Users", "Inexperienced Users", and "Proficient Users" provide filters to customize the graph if user information is available.**

reaching the penultimate layer. The output from this layer is a highly compressed feature vector that encapsulates key visual information of the image. Through this approach, we can simplify complex image data into a one-dimensional vector form. Consider a video $\{f_0, f_1, \ldots, f_{N-1}, f_N\}$, where $f_N$ is the current frame and $f_{N-1}$ is the previous frame. We apply the ResNet-18-ImageNet to represent images frame by frame resulting in a sequence of vectors $\{\mathbf{V}_0, \mathbf{V}_1, \ldots, \mathbf{V}_{N-1}, \mathbf{V}_N\}$, and calculate the cosine similarity $S$:

$$S(\mathbf{V}_{N-1}, \mathbf{V}_N) = \frac{\mathbf{V}_{N-1} \cdot \mathbf{V}_N}{\|\mathbf{V}_{N-1}\| \|\mathbf{V}_N\|} \tag{1}$$

between consecutive frames (the result is between 0 and 1, where a higher value indicates a strong level of similarity).

*2.3.2 Extraction.* Figure 2 (a) shows the relationship of $S$ as it changes with $N$. Since the interface remains in a stable state when the user is not performing any actions, we consider a period during which the similarity curve remains stable as indicating the user is staying on a particular interface. This is based on the following considerations: 1) Interfaces during loading sometimes also remain stable, but a loading screen should not be considered as an interface. 2) When the user only makes minor changes to the interface, such as slight scrolling, it should still be considered as staying on the same interface. Therefore, we provide two judgment thresholds, $T$ (frame count) and $R$ (range, defined as the range of similarity change over a period), to determine whether an interface is stable. After testing with the collected dataset, we suggest that an interface is considered stable when $T \geq 8$ and $R \leq 0.94$. After detecting shots that are in a state of stable similarity, we extract the middle frame of each duration as the key frame, representing the interface where the user stayed. Up to now, a screen recording $SR_N$ has extracted the corresponding sequence of key frame nodes $\{N_0, N_1, \ldots, N_{t-1}, N_t\}$ (assuming there are $t$ keyframes). Hereafter, this series of ordered images is referred to as the $SR_N$'s *user's page flow*.

After the extraction of multiple groups of videos, we obtain several corresponding user's page flows, mapping interfaces to nodes. Continuing with the same method of similarity calculation, we merge nodes whose similarity exceeds the threshold $t$ (the recommended value for $t$ is 0.86). The transition from interface A to interface B is mapped as a directed edge from node A to node B. Finally, we label the pages included in the ground truth, and set up click interactions for each node (displaying the interface mapped to the node) and edge (displaying the adjacent nodes connected by the edge, as well as the corresponding user's thought simulation at this step, with thought simulation information coming from subsection 2.5).

*2.3.3 Alternative.* We compare the performance of our method for calculating similarity based on ResNet-18-ImageNet to the performance of that based on Structural Similarity Index Measure (SSIM) [18]. SSIM is a technique for measuring the similarity between two images based on their structural information, brightness, and contrast, reflecting more accurately how humans perceive image quality. The similarity calculation method based on SSIM could be more sensitive to changes in interfaces. As shown in Figure 2 (b) and (c), the similarity curve obtained based on SSIM yields one more keyframe compared to our method. However, the results indicate that this is only due to an unexpected popup during operation.

## 2.4 Data Collection

To demonstrate our process of visualizing user operations and support the LLM simulation of user thoughts described below, we collect a set of screen recordings in mobile apps. We recruit 20 students (17 males, 3 females; $Age_{\text{mean}} = 21.95$) via word-of-mouth in a local university to perform a usability test in 12 Android smartphone apps [2], each with three tasks. With the participants' permission,

[2]These apps are Tik Tok (Video), Judou (Reading), Taobao (Shopping), Qishui Music (Music), Meituan Delivery (Lifestyle), RQrun (Exercise), Weibo (Social Media), Little

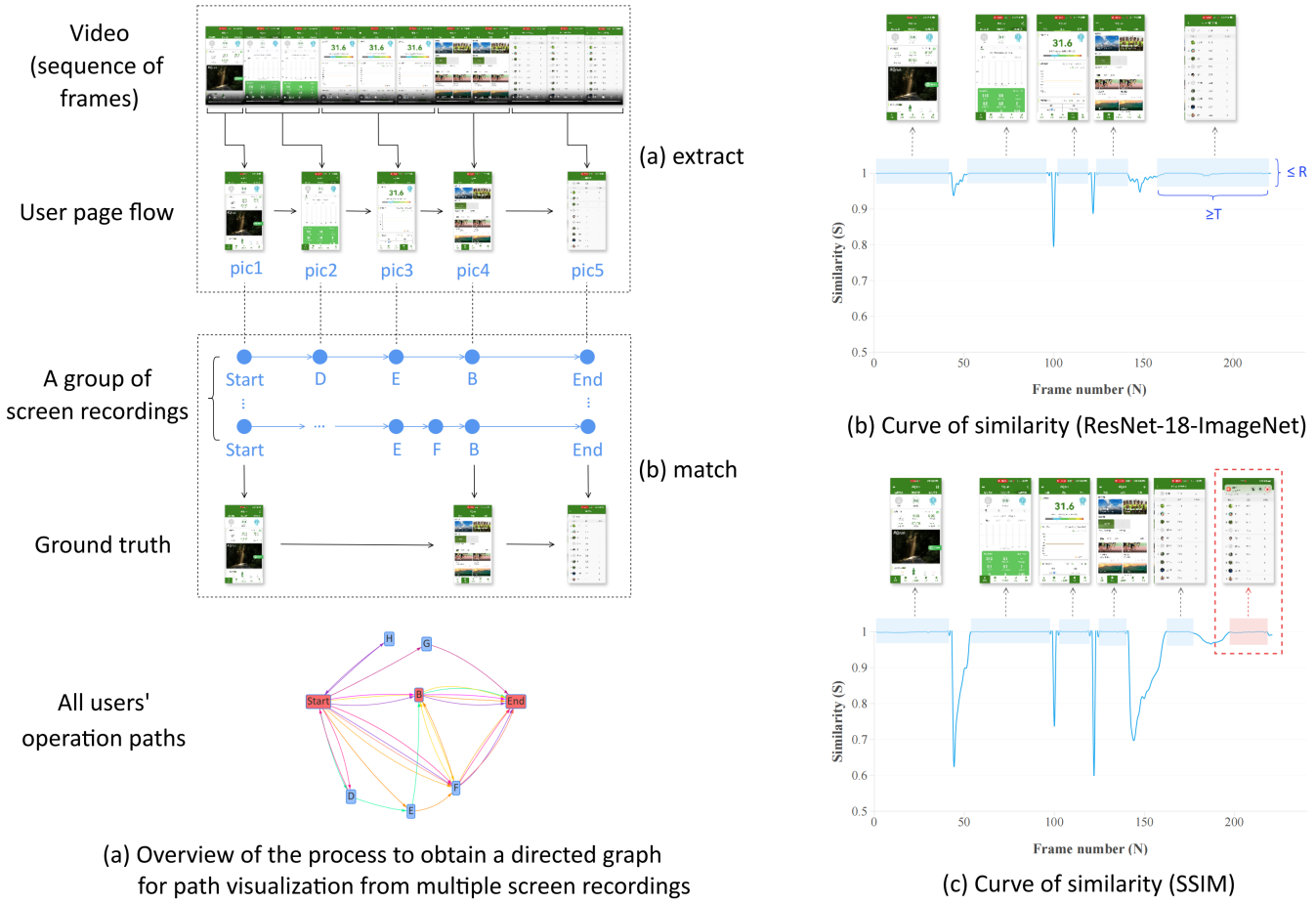



**Figure 2: Our process (a) and its performance in an example (b) compared to an alternative (c) for visualizing users' operation paths of mobile apps based on screen recordings.**

we invite them to sequentially complete three tasks related to each app in a phone (Huawei Honor 20), with each task being recorded separately via the phone's built-in screen recording tool, which does not include operation log records, touch feedback, or any additional information. For each task, we first ask participants to rate their familiarity of it on a 5-point Likert scale, an optional step in our demonstrated *DesignWatch*. During each task, we encourage the participants to think aloud, *e.g.,* describe their thoughts on the operations. Each participant has a 5-minute break after completing every 6 out of 36 tasks across 12 apps. Each participant spends 90-120 minutes in our study and gets about 8 USD for compensation.

## 2.5 Simulation of User Thoughts

We explore the potentials of GPT-4V for simulating user thoughts in the extracted user operation from screen recordings. We provide a prompt structure to inspire GPT-4V to offer interpretations of the user operations from a first-person perspective (Figure 3). Every prompt begins with a *preamble*, an explanation of the purpose of the prompt. After the *preamble*, there are multiple *exemplars* composed of inputs and outputs for each task. The input of each example includes a textual description of the task and a series of ordered UI screenshots. The output part is a structured text obtained by manually annotating and organizing real users' think-aloud. The left part in Figure 3 shows an example of a one-shot prompt. During prediction, we feed the model with the prompt, appending a new input screen at the end. Therefore, for N-shot learning, the prompt will consist of a *preamble*, N *exemplars*, and the test screen for prediction, as shown in the right part of Figure 3.

## 3 PRELIMINARY STUDY

To evaluate the usefulness of *DesignWatch* for helping designers analyzing users' operation of mobile apps based on screen recordings, we conducted a preliminary study with four designers. P1 obtained a master's degree in design and has 5 years of experience as a UX designer. P2 is E2 who helped us in the design process (subsection 2.1). P3 is a Ph.D. student in Human-Computer Interaction and has 8 years of experience in user experience (UX) design. P4

Sleep (Health), Ctrip (Travel), Tencent Meeting (Office), Kingsoft Dictionary (Tool), Boss Zhipin (Job search). We attach the details of the user tasks in Supplementary Material.

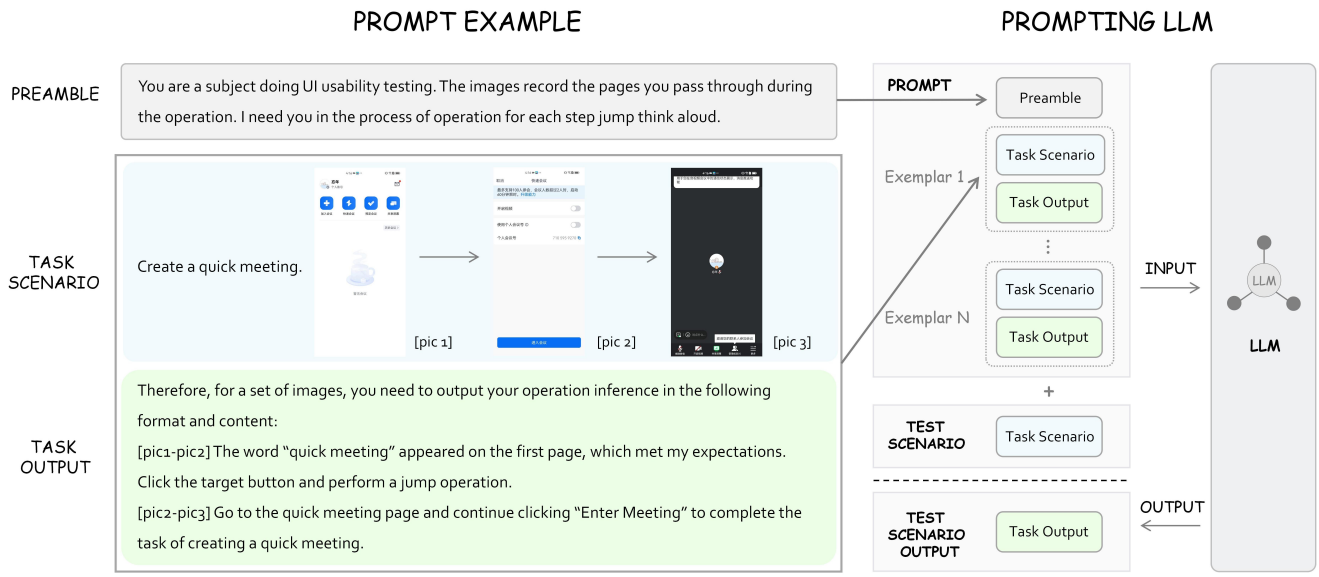



**Figure 3: Left: Example of proposed prompt structure. Begins with task description, followed by zero or more task examples with input screens and outputs. Right: Illustration of prompting multimodel LLMs in our use cases. The prompt includes N examples from target tasks, plus the user's page flow. Prompt + test info (scenario, *user's page flow*) input to GPT-4V, which generates word tokens to infer the user's operation.**

has over 10 years of product design background and is currently a UI/UX course lecturer at a university.

Each participant was invited to a 60-75 minute remote session with us via VooV Meeting. After obtaining permission, we recorded the online meeting. We first introduced the motivation and usage scenarios for developing *DesignWatch* to the participants, followed by a walk-through of *DesignWatch*. Then, we asked them to identify UX design issues within the screen recording data of a mobile app. We provided each participant with three tasks in our collected data, *i.e.*, finding the 'Situational Dialogues' in the Kingsoft Dictionary app, viewing the sleep report in the Little Sleep app, and checking their ranking in the user group in the RQrun app. For each scenario, we provided 10 screen recordings from different users, along with groups that have been sorted based on the users' familiarity with the operations. We informed the participants that they would play the role of a UX designer for their chosen scenarios and use *DesignWatch* to analyze the screen recordings. Participants were asked to think aloud during use. The app in the test scenario was one that the participants had not seen before. *DesignWatch* was locally deployed on one of the authors' computers, and participants used the tool remotely via VooV Meeting control. Lastly, participants reported their overall impressions and suggestions on *DesignWatch*. Each designer spent around one hour in our interview and got about 27 USD for compensation.

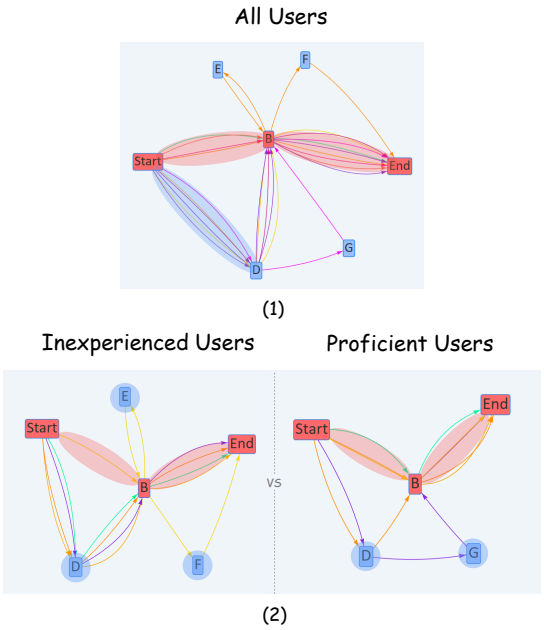
## 3.1 Case: View sleep report in the Little Sleep app

P2 had a trial on analyzing the usability of the Little Sleep app for supporting users to view the reports of their sleep.

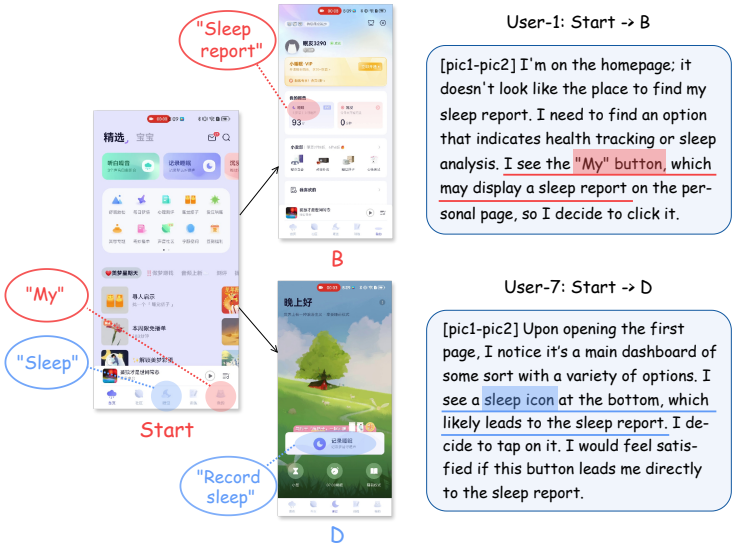
**Understanding users' operation patterns.** P2 first observed all users' operation paths (Figure 4a (1)). *"I see the most densely connected paths highlighted in red, indicating that most users performed the correct operations with few errors. However, Node D, an unexpected operation from my perspective, attracted many users in their task completion process"*. He filtered the operations performed by inexperienced users and proficient users (Figure 4a (2)) by clicking the corresponding buttons on the page. *"I would like to check whether different types of users would perform differently in this task. It turns out users who were not familiar with this kind of app were more likely to enter the UI page noted as D"*. P2 had a preliminary conclusion that the UX design for performing the "View sleep report" task in the Little Sleep app is generally successful, but there was potential to improve it, especially for new users.

**Identifying the potentially problematic UI design pages.** P2 switched back to the "All users" mode and started to repeatedly click on the nodes and edges. He toggled between $D$ (the erroneous node) and $B$ ($D$'s sibling node) and noticed that both $B$ and $D$ interfaces contained prompts for "sleep recording" or "sleep". He then traced back to their shared parent node, $start$, to understand how the original design differentiated guidance between $B$ and $D$. He observed an interface dense with content, which requires further analysis to pinpoint the cause of ambiguity. At this stage, P2 had identified the start interface as the potentially problematic UI design page.

**Gaining insights for improving the app design.** P2 would like to understand how many users think when taking action in this interface. He clicked on the edges $start- > B$ and $start- > D$ from user-1 and user-7 to understand their specific operations. As




(a) Visualized operation paths.




(b) UI pages of nodes *B* and *D* users' operation inference.

**Figure 4: Illustration of the results obtained by P2 using *DesignWatch* in the case "View sleep report in the Little Sleep app".**

shown in Figure 4b, user-1's pathway to node *B* reflected that there were no clear prompts on the start interface, leading them to select the "My" button for further exploration. Conversely, user-7, who additionally navigated to node D, interpreted the "Sleep" icon as directly relating to sleep reports and tried this button. From this operation analysis, P2 learned that the "Sleep" icon in the starting interface could be misleading if users want to view their sleep reports. *"If we continue to refine the UI, these two distinct pathways clearly highlight the design's vulnerabilities. Designers could make specific improvements based on actual needs."*

## 3.2 Perceptions towards *DesignWatch*

All designers agreed that using interactive directed graphs to summarize a group of screen recordings is an innovative idea, providing them with an intuitive information presentation. This feature has helped them save time and effort in manually reviewing recordings and summarising the operation path characteristics of multiple users. They concurred that the interaction with edges and nodes is necessary, and *DesignWatch*'s interactive design can organize the interface's image information and contextual relationships reasonably. In some scenarios (as mentioned by P2 in case 1), considering data filtering based on user background information met the application needs of designers. We also received suggestions for improving the visualization design. For example, P3 pointed out that the hints for interaction with edges and nodes in the directed graph are not obvious, which will be easily overlooked when designers use it for the first time. Our experts believe that the simulated thoughts can provides reference information about specific user operations, such as where they clicked. P2, P3, and P4 expressed their willingness to refer to the simulated information from *DesignWatch* in practical usability tests, while P1 suggested the need for further judgment and filtering for this function to enhance the credibility of the information.

## 4 DISCUSSION AND FUTURE WORK

In this paper, we contribute an interactive system *DesignWatch* for assisting designers in analyzing users' operations of mobile apps based on screen recordings. Our study findings provide insights into using visualization techniques and multimodal LLMs for facilitating usability tests. In our work, *DesignWatch* transforms multiple screen recordings collected during usability tests into an interactive directed graph, which helps designers quickly understand the operation patterns of all users in the app. In line with related work on simulating user behaviours by LLMs [1, 8, 24], we show that simulating the user's thoughts on their actions by a multimodal LLM is promising in multimodal scenarios, *e.g.*, operating the mobile apps in our case. This could encourage future work to leverage multimodal LLMs to explain what a human would see, how a human would feel, and what would a human do in other multimodal tasks like viewing a graphic poster [2, 15] and learning with the video lectures in MOOC [9, 11]. In the future, we plan to evaluate *DesignWatch* via a user study that assesses its effectiveness compared to the baseline approach for analyzing the screen recordings and a field study with mobile app designers in their usability tests.

## ACKNOWLEDGMENTS

This work is supported by the General Projects Fund of the Natural Science Foundation of Guangdong Province in China with Grant No. 2024A1515012226.

# REFERENCES


[1] Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. 2024. Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 8–17.

[2] Siyuan Chen and Julien Epps. 2020. Multimodal coordination measures to understand users and tasks. *ACM Transactions on Computer-Human Interaction (TOCHI)* 27, 6 (2020), 1–26.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[4] Sidong Feng, Chunyang Chen, and Zhenchang Xing. 2023. Video2Action: Reducing human interactions in action annotation of app tutorial videos. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–15.

[5] Vagner Figueredo de Santana and Felipe Eduardo Ferreira Silva. 2019. User test logger: An open source browser plugin for logging and reporting local user studies. In *International Conference on Human-Computer Interaction*. Springer, 229–243.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[7] Xiaozhu Hu, Yanwen Huang, Bo Liu, Ruolan Wu, Yongquan Hu, Aaron J Quigley, Mingming Fan, Chun Yu, and Yuanchun Shi. 2023. SmartRecorder: An IMU-based Video Tutorial Creation by Demonstration System for Smartphone Interaction Tasks. In *Proceedings of the 28th International Conference on Intelligent User Interfaces*. 278–293.

[8] Yan Leng and Yuan Yuan. 2023. Do LLM Agents Exhibit Social Behavior? *arXiv preprint arXiv:2312.15198* (2023).

[9] Nan Li, Łukasz Kidziński, Patrick Jermann, and Pierre Dillenbourg. 2015. MOOC video interaction patterns: What do they tell us?. In *Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, September 15-18, 2015, Proceedings 10*. Springer, 197–210.

[10] Griselda Manzano-Monfort, Guillermo Paluzie, Mercedes Díaz-Gegúndez, and Carolina Chabrera. 2023. Usability of a mobile application for health professionals in home care services: a user-centered approach. *Scientific Reports* 13, 1 (2023), 2607.

[11] Ahmed Ali Mubarak, Salah AM Ahmed, and Han Cao. 2023. MOOC-ASV: Analytical statistical visual model of learners' interaction in videos of MOOC courses. *Interactive Learning Environments* 31, 5 (2023), 3055–3070.

[12] Fatih Nayebi, Jean-Marc Desharnais, and Alain Abran. 2012. The state of the art of mobile application usability evaluation. In *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 1–4.

[13] Marcus Nyberg, Mikael Goldstein, and Ying Leung. 2001. Visualising data using the ActionMapper: A proposed interactive event logger for user interface evaluation. In *Proceedings Fifth International Conference on Information Visualisation*. IEEE, 147–154.

[14] Ekaterina Olshannikova, Aleksandr Ometov, Yevgeni Koucheryavy, and Thomas Olsson. 2015. Visualizing Big Data with augmented and virtual reality: challenges and research agenda. *Journal of Big Data* 2 (2015), 1–27.

[15] Roope Raisamo. 1999. *Multimodal Human-Computer Interaction: a constructive and empirical study*. Tampere University Press.

[16] Daniel Rose, Vaishnavi Himakunthala, Andy Ouyang, Ryan He, Alex Mei, Yujie Lu, Michael Saxon, Chinmay Sonar, Diba Mirza, and William Yang Wang. 2023. Visual chain of thought: Bridging logical gaps with multimodal infillings. *arXiv preprint arXiv:2305.02317* (2023).

[17] Yuqing Wang and Yun Zhao. 2023. Gemini in reasoning: Unveiling commonsense in multimodal large language models. *arXiv preprint arXiv:2312.17661* (2023).

[18] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[19] Paweł Weichbroth. 2020. Usability of mobile applications: a systematic literature study. *Ieee Access* 8 (2020), 55563–55577.

[20] Kam Kwai Wong, Xingbo Wang, Yong Wang, Jianben He, Rong Zhang, and Huamin Qu. 2023. Anchorage: Visual analysis of satisfaction in customer service videos via anchor events. *IEEE Transactions on Visualization and Computer Graphics* (2023).

[21] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421* 9, 1 (2023), 1.

[22] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549* (2023).

[23] Haipeng Zeng, Xingbo Wang, Yong Wang, Aoyu Wu, Ting-Chuen Pong, and Huamin Qu. 2022. Gesturelens: Visual analysis of gestures in presentation videos. *IEEE Transactions on Visualization and Computer Graphics* (2022).

[24] Erhan Zhang, Xingzhu Wang, Peiyuan Gong, Yankai Lin, and Jiaxin Mao. 2024. USimAgent: Large Language Models for Simulating Search Users. *arXiv preprint arXiv:2403.09142* (2024).