IIAI Open Conference Publication Series
IIAI Letters on Informatics and Interdisciplinary Research
Vol.006, LIIR375
DOI: https://doi.org/10.52731/liir.v006.375

Experimental Evaluation based on a Technology Acceptance Model for the Use of Congestion Mitigation Applications

Negai Nakamoto *, Hiroyoshi Matsumoto *,
Shimpei Matsumoto *

Abstract

This study proposes a web application that manages and optimizes user movement in real time to reduce congestion and the risk of crowd-related accidents at tourist sites, event venues, and evacuation areas during disasters. The application features a navigation system utilizing checkpoints, guiding users step-by-step toward their destination to prevent crowding at specific routes or locations. By employing Dijkstra's algorithm, it calculates optimal routes based on congestion levels and the capacity of each checkpoint, enabling personalized route recommendations. In the event of a disaster, the system dynamically optimizes evacuation routes to support safe and efficient evacuation. User behavior data and congestion information are continuously recorded and analyzed to enhance system accuracy and optimize capacity settings. This system is expected to contribute to solving societal issues such as promoting sustainable tourism, preventing crowd accidents in urban areas, and supporting evacuation during emergencies.

Keyword: Simulation, Over Tourism, Application, Useability

1 Introduction

1.1 Research Background and Objective

In 2023, the Japanese government approved a new "Basic Plan for Promoting Tourism Nation," which aims to promote tourism policies based on three key words: "sustainable tourism," "expansion of consumption," and "promotion of regional tourism." Tourism has been positioned as a pillar of growth strategies even after the pandemic, and is seen as playing an important role in regional revitalization and international mutual understanding. On the other hand, overtourism has become a serious problem. The concentration of a large number of tourists has led to issues such as traffic congestion, illegal parking, problems at various facilities, and noise [1]. In particular, in Kyoto, friction has arisen between tourists and local residents as the number of tourists increases. For example, tourists parking illegally in residential areas or making noise late at night have disrupted the lives of local residents. Moreover, congestion around tourist attractions has strained local infrastructure, negatively affecting daily life. To resolve these issues, there is a strong need to spatially disperse tourists. For

---

* Hiroshima Institute of Technology, Hiroshima, Japan

example, in Venice, measures such as the introduction of a tourism tax and restrictions on visiting times have been taken to disperse tourists, but these have not yet proven to be fully effective [2].

This study aims to develop a web application utilizing digital communication technology to promote the spatial dispersion of tourists and ensure safe evacuation during disasters. In response to issues such as overcrowding at tourist destinations and overtourism, the application uses location data and check-in functions to monitor the real-time distribution of people and suggest routes that avoid congested areas. It seeks to alleviate crowding at tourist sites and reduce friction with local residents.

Furthermore, the application is designed to support evacuation during emergencies. By dynamically optimizing evacuation routes based on evacuees' current locations, damage status of buildings and roads, and the distribution of people, it helps reduce the risk of crowd-related accidents and supports safe and efficient evacuations. The application contributes to the realization of a sustainable and comfortable environment in both tourism and disaster preparedness contexts.

This paper consists of five chapters. Chapter 1 presents cases of population congestion at tourist destinations and specific evacuation sites, and discusses the significance of developing the proposed application. Chapter 2 explains related studies, including the usefulness of applications during congestion, and introduces the System Usability Scale (SUS) and Technology Acceptance Model (TAM), which highlight the benefits of providing personalized information. Chapter 3 describes an overview of the congestion mitigation system using Dijkstra's algorithm and the system for personalized route recommendations, along with the system configuration and details. Chapter 4 outlines the flow of the demonstration experiment, detailing user route recognition and movement using the application. It also discusses the experimental results and questionnaire findings. Chapter 5 evaluates the effectiveness of the proposed application through simulations using the Floor Field Model, reconsiders the internal system, and proposes a more efficient route recommendation method. Finally, Chapter 6 presents the conclusion and summarizes the study.

2 Research Background and Related Studies

2.1 System Usability Scale

The System Usability Scale (SUS) is a usability evaluation method developed by John Brooke in 1986. It is widely used as a tool to evaluate the ease of use of systems and products. This method consists of a simple set of 10 questions, allowing users to provide subjective assessments of the system's usability, which can then be quantified. Due to its simplicity and versatility, SUS has been adopted as a standard usability evaluation method in both industry and academia, playing an important role especially in the design and improvement of new systems and applications [3].

To accurately measure the usability of a system, it is necessary to consider not only the evaluation of its functionality but also the context in which the system is used. Specifically, factors such as the characteristics of the users, the intended purpose of use, and the usage environment all influence the evaluation. Therefore, when using SUS, it is important to clarify what kind of users the system is intended for and in what scenarios it will be used. There are three main aspects to usability measurement:

1. **Effectiveness**: This indicator shows whether the user can achieve their goals by using the system. For example, whether a user can correctly purchase a product on an online shopping site or reach their destination using a navigation system is evaluated.
2. **Efficiency**: This indicator measures the cost, such as time and effort, required to achieve the user's goals. For instance, whether the search function quickly helps the user find the desired information or if the system responds in a timely manner is assessed.
3. **Satisfaction**: This indicator reflects the comfort and satisfaction the user feels when using the system. Aspects such as whether the system allows intuitive operation, if the design is easy to understand, and whether it can be used without stress are evaluated.

Among these indicators, effectiveness and efficiency are measured on different scales depending on the system's usage. For example, in a system controlling continuous industrial processes, it is crucial to avoid errors and ensure rapid response. On the other hand, in a text editor used daily, factors like ease of typing and command execution speed are more important. Thus, even though the concepts of "effectiveness" and "efficiency" remain the same, the evaluation perspective may vary depending on the type of system and the usage scenario. Therefore, it is necessary to set appropriate evaluation criteria.

2.2 Technology Acceptance Model

In this study, the Technology Acceptance Model (TAM) is used as a means to analyze whether the proposed system can be accepted. Originally, TAM was designed to clarify how employees within an organization accept and use information systems, serving as a model for human behavior intention. However, due to its simplicity and theoretical soundness, TAM is a general model that can be applied to many information systems and across various types of users. The model identifies five factors that influence users' adoption of information systems: Perceived Usefulness, Perceived Ease of Use, Attitude toward Using, Behavioral Intention to Use, and Social Influence [4].

The relationships between these factors in the Technology Acceptance Model are as follows: Perceived Ease of Use, which represents the degree to which users expect that using an information system will be effortless, positively influences Perceived Usefulness, which represents the degree to which users feel that using the system will improve their performance. Additionally, both Perceived Usefulness and Perceived Ease of Use positively influence Attitude toward Using, which reflects users' positive or negative feelings toward using the system. Attitude toward Using and Perceived Usefulness then positively influence Behavioral Intention to Use, which is the intention to actually use the system. Finally, Social Influence has a positive influence on Perceived Ease of Use, Perceived Usefulness, Attitude toward Using, and Behavioral Intention to Use [4].

2.3 Floor Field Model

The Floor Field Model is a lattice-based model used to simulate pedestrian movement. The walkable cells are represented as blank spaces, and the cells occupied by agents are marked with circles within the grid to represent the movement of pedestrians. The movement of agents has a degree of randomness. To modify the movement of agents, the Floor Field Model defines a static floor field. This floor field assigns a distance from a specific point to each grid cell, and the direction of movement for agents is determined by modifying the probability to make it closer to the destination [5].

In the simulations conducted in this study, the static floor field is set using a distance setting based on Manhattan distance. Manhattan distance is defined as the number of turns required to reach a specific cell from a particular point, where only vertical and horizontal moves are allowed.

With the Manhattan distance-based setting, a long vertical cluster near the exit can be observed, which allows the configuration of agents with minimal horizontal intrusion. On the other hand, with the Euclidean distance-based method, a semicircular cluster near the exit can be seen, allowing the configuration of agents with more horizontal intrusion.

In this study, we assume that users will be rational and able to follow the app's instructions, so the Manhattan distance-based static floor field setting was chosen. Additionally, in this study, static floor fields are defined for the same number of agents, so each agent is set to move towards its respective destination.

3 Proposed System

3.1 Overview of the Proposed System

In event venues and sports stadiums, where entrances and exits are highly trafficked, significant congestion is expected. Therefore, the proposed system uses a web application to alleviate congestion and guide users smoothly by managing their movement routes with checkpoints along the exit path. The system aims to

1. Users input their current location and destination.
2. Display all the checkpoints along their route.
3. Follow the app's instructions to progress to the checkpoints.
4. Check-in at each checkpoint.
5. If there are more checkpoints, return to step 3; if no further checkpoints are present, the destination is considered reached, and the system returns to step 1.

The flow of user actions is as follows: the user inputs their current location and destination, and the system displays checkpoints along their route in a card format. When users reach a checkpoint, they press a check-in button to indicate they passed it. This action updates the server, removes the checkpoint from the list, and displays the next checkpoint.

Before the system's operation begins, a database of checkpoints and routes must be accurately recorded, containing information about the building layout. The checkpoint database manages specific points inside the building, and the route database holds the path information between these points. Proper preparation and testing of these databases before system launch ensure smooth navigation for users.

3.2 Congestion Alleviation System Using Dijkstra's Algorithm

3.2.1 Overview of the Congestion Alleviation System

The system manages users' movements in real-time and guides people away from congested areas to less crowded sections, making the flow of people more efficient and suggesting the appropriate exits. The system uses Dijkstra's algorithm to compute optimal routes and guides multiple users with individual route recommendations. Additionally, a reset button allows route re-optimization when users are lost or when congestion changes.

3.2.2 System Interface and Processing Flow

The system estimates congestion based on other users' checkpoint passing statuses. This data allows the system to automatically reorder the checkpoint cards displayed to other users. For example, after an event, when many spectators exit at once, the app avoids congested areas and offers relatively less crowded routes. By passing checkpoints sequentially, user flow is dispersed, alleviating congestion.

When the user presses the pass button at a checkpoint, the following actions occur:

1. Dijkstra’s algorithm calculates the route from the current location to the destination using the database, excluding previously passed checkpoints.
2. If the capacity of an adjacent checkpoint exceeds its limit, the system excludes that checkpoint and recalculates the route.
3. If all adjacent checkpoints are full, the user is instructed to wait.
4. The user’s location is saved in the database.
5. User ID, checkpoint name, time, and capacity estimates are saved to a CSV file.

This approach dynamically suggests routes based on real-time congestion data, helping users avoid crowded areas and reach their destinations smoothly.

3.2.3 Dijkstra's Algorithm

Dijkstra's algorithm is a widely used method for solving the single-source shortest path problem, considering the movement cost between nodes to find the optimal route. In this system, each checkpoint is a node, and the routes between them form a complex network. The algorithm calculates the most efficient route by considering both the user's current location and the congestion status of surrounding checkpoints. This dynamic route optimization ensures that the user avoids congested areas and follows the most efficient path.

Here is the typical flow for Dijkstra's algorithm:

1. Select the unvisited node with the smallest current distance.
2. Calculate the total distance for each adjacent node, considering the edge weights.
3. Remove the processed node from the set of unvisited nodes.
4. Repeat until all nodes have been visited, determining the shortest paths to each node.

3.2.4 Determining Individual Routes

For individualized route suggestions, the relationship between relevant checkpoints is clarified from the database. A graph is defined in Python, where each checkpoint is connected.

Dijkstra's algorithm calculates the shortest route between the current and destination checkpoints. The system continuously updates the route and user location, ensuring that the user avoids congested areas by dynamically adjusting their route.

If an adjacent checkpoint exceeds its capacity, the user is instructed to wait, and their ID is added to the waiting list. When space becomes available, the route suggestion is updated, helping avoid bottlenecks at specific checkpoints.

4 Experimental Methods and Results

4.1 Experimental Procedure

For the experiment, 15 students from the Faculty of Information Science at Hiroshima University of Technology were asked to simultaneously use the application to conduct a usability evaluation experiment. The application was designed with the premise that users would receive navigation instructions in unfamiliar places, so to carry out a more practical evaluation, the experiment was conducted in Building 16, a location that the participants were not familiar with. This allowed for a more accurate evaluation of the intuitive usability and the effectiveness of the navigation feature when the user first uses the app.

During the experiment, each student used their own smartphone to access the specified URL via "Sakura Network" and launched the web application. The application used HTML, CSS, and JavaScript for the front-end development, and the display updated whenever the user pressed the "pass button," which guided them to the next checkpoint. This simple operation design made it easier for users to intuitively understand their next destination, preventing unnecessary confusion.

For the experiment, all participants were provided with the same route to ensure a uniform testing condition. This allowed for the evaluation of the usability of the application without being influenced by individual route choices. The procedure involved each participant following the instructions of the application, pressing the pass button each time they reached a checkpoint, repeating the process three times until they reached the final destination.

After the experiment, participants were asked to complete a questionnaire immediately after using the app, collecting quantitative data on the usability of the application. The questionnaire used a 0-10 interval scale to quantify the subjective evaluations of the participants.

4.2 Technology Acceptance Model Evaluation

The hypotheses in this study based on the TAM model are as follows:

H1: "Perceived Ease of Use" has a positive effect on "Perceived Usefulness."

H2: "Perceived Ease of Use" has a positive effect on "Attitude Toward Use."

H3: "Perceived Usefulness" has a positive effect on "Attitude Toward Use."

H4: "Perceived Usefulness" has a positive effect on "Behavioral Intention to Use."

H5: "Attitude Toward Use" has a positive effect on "Behavioral Intention to Use."

H6: "Social Influence" has a positive effect on "Perceived Ease of Use."

H7: "Social Influence" has a positive effect on "Perceived Usefulness."

H8: "Social Influence" has a positive effect on "Attitude Toward Use."

H9: "Social Influence" has a positive effect on "Behavioral Intention to Use."

Based on these hypotheses, a model was constructed using Python and the survey results were analyzed using covariance structure analysis with the *semopy* library. The Technology Acceptance Model uses multiple fit indices such as GFI and RMSEA to determine the model's suitability [6]. The analysis of the model fit indices showed GFI (Goodness of Fit Index) = 0.99, AGFI (Adjusted Goodness of Fit Index) = 0.94, and RMSEA (Root Mean Square Error of Approximation) = 0. This indicates that the model fits well as it meets the general criteria of GFI being above 0.99 and RMSEA being between 0.05 and 0.08 [7].

The results are shown in Figure 1 and Table 1. All hypothesis paths were statistically significant at the 0.1% significance level, indicating that the causal relationships identified by TAM hold true for the proposed application [8].

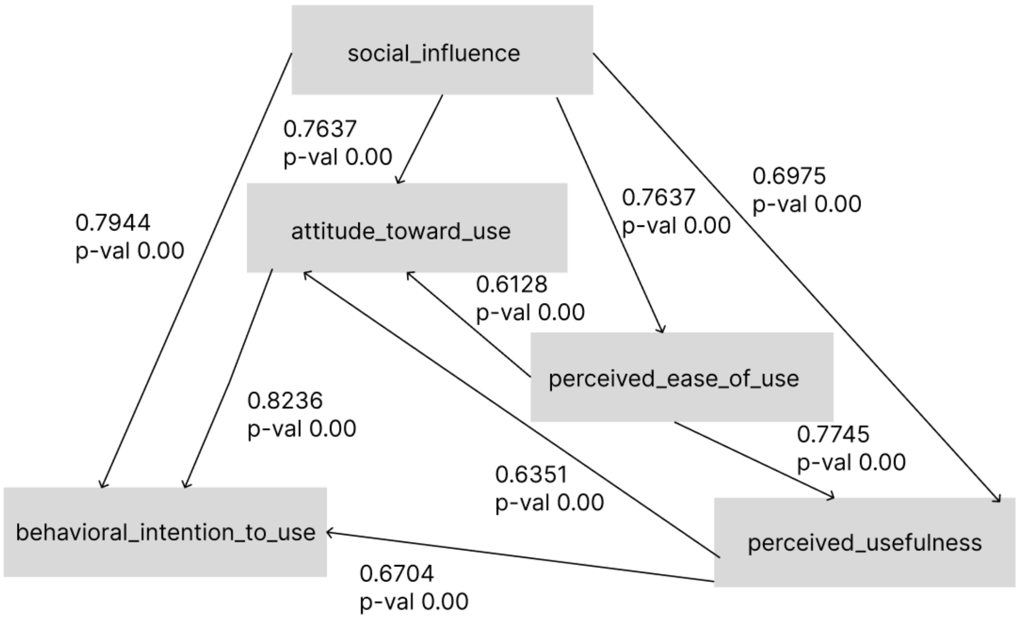

Figure 1: Results of Score Comparison

Table 1: Results of Technology Acceptance Model Analysis

Standardized Coefficient Estimate	Standard Error	t-value	Significance Probability (p-value)
H1	0.7745	0.1195	6.4787
H2	0.6704	0.1402	4.7819
H3	0.6128	0.1493	4.1041
H4	0.6351	0.1459	4.3514
H5	0.8236	0.1071	7.6855
H6	0.7637	0.1219	6.2616
H7	0.6975	0.1354	5.1510
H8	0.7637	0.1354	5.1510
H9	0.7944	0.1147	6.9226

Based on the analysis results, each hypothesis path was tested. For the impact on "Perceived Usefulness," the standardized coefficients from "Perceived Ease of Use" were 0.774 (H1: $t = 8.314$, $p < 0.001$), and from "Social Influence" were 0.697 (H7: $t = 8.314$, $p < 0.001$). Users' perception of the application's usefulness was influenced by the ease of use of the UI and by the positive perception of the application's utility within the users' social group.

For the impact on "Attitude Toward Use," the standardized coefficients from "Perceived Ease of Use" were 0.670 (H2: $t = 4.781$, $p < 0.001$), from "Perceived Usefulness" were 0.612 (H3: $t = 4.104$, $p < 0.001$), and from "Social Influence" were 0.763 (H5: $t = 5.151$, $p < 0.001$). This suggests that making the route easy to recognize through the UI and highlighting the application's usefulness encourages users to actively use the application.

For the impact on "Behavioral Intention to Use," the standardized coefficients from "Attitude Toward Use" were 0.823 (H2: $t = 7.685$, $p < 0.001$), from "Perceived Usefulness" were 0.635 (H3: $t = 4.351$, $p < 0.001$), and from "Social Influence" were 0.794 (H5: $t = 6.922$, $p < 0.001$). This suggests that users who are positively disposed toward the app or understand its usefulness are more likely to consider using the application.

From this, it is inferred that in order to encourage users to actively use the proposed application, the app's usefulness should be made clear through the UI. Specifically, making the situation where congestion is alleviated visually clear in the app's interface could encourage users to use the app.

4.3 System Usability Scale Evaluation

SUS is a simple questionnaire used to evaluate the ease of use of systems or products. The average score of the 15 participants was 65.87, which is below the general SUS score benchmark of 68. The discrepancies between the images used for checkpoints and the actual view, depending on lighting, were likely factors contributing to this result. Throughout the experiment, some participants mistakenly followed unintended routes due to misidentifying the checkpoints. The 16th building, which was used for the experiment, lacked clear landmarks for identifying checkpoints, which made it difficult to distinguish them. It is essential to select easily recognizable checkpoints, but it was also noted that external factors play a significant role in the selection of checkpoints, and clearer route suggestions within the system are necessary. Possible solutions include using AR to show people actually moving along the route or enabling users to view the checkpoint area in 360 degrees, which could improve recognition accuracy.

5 Simulation to Demonstrate the Utility of the App

5.1 Overview of the Simulation

In this study, a simulation was conducted using the processing software and a floor field model to demonstrate the utility of the proposed application., as demonstrated in Figure 2 [5]. As shown in the figure, the simulation was set up with a 50×50 grid. Two checkpoints were set, and a program was implemented where 500 agents move toward the goal. The checkpoints were set as 3×3 grid squares. The agents move toward the checkpoints, and once an agent reaches the area of a checkpoint, it continues to the next checkpoint. This process repeats, and each agent aims to reach its destination. In this study, simulations were run 100 times under each condition, and the average values were calculated to compare the results.

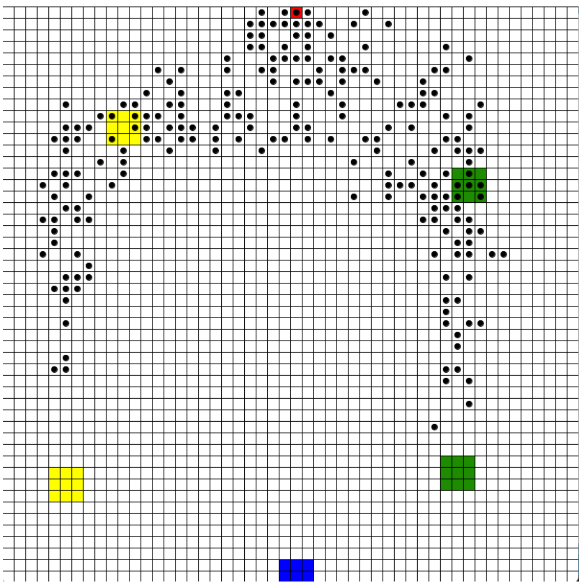

Figure 2: Simulation of floor field model

5.2 Conditions of the Simulation

As shown in the figure, four different conditions were set for the simulation in this study. The first condition defines the route passing through the two checkpoints as the shortest path and moves 500 agents along this route.

The second condition, in addition to the shortest route, sets up a detour route. The first 250 agents move along the shortest route, while the remaining 250 agents take the detour route. The detour route is set to be longer than the shortest route. Like the shortest route, the detour route passes through the same two checkpoints and leads to the same goal.

The third condition sets up both the detour and the shortest route, similar to the second condition. Agents whose numbers are not multiples of 2 move along the shortest route, while those whose numbers are multiples of 2 take the detour route. This method allows for a more distributed movement of agents.

5.3 Evaluation of the Simulation

After conducting 100 trials for the four simulations, the average values were found as follows: the first condition had 656.59 turns, the second had 442.18 turns, the third had 379.6 turns, and the fourth had 458.07 turns. When comparing the first and second conditions, it was clear that directing all agents to the shortest route resulted in more congestion. In contrast, directing half of the agents to the detour route significantly alleviated the congestion in the later stages. In the first simulation, many agents were observed to be stalled near the checkpoints, suggesting that by dispersing agents, smoother movement could be achieved. These results suggest that individual route suggestions are effective for congestion alleviation, demonstrating the utility of the proposed application.

Additionally, when comparing the second and third conditions, it was found that the third condition resulted in about 80 fewer turns. This indicates that it is more effective for congestion alleviation to alternate agents between the shortest route and the detour route, rather than directing half of the agents to the shortest route and the other half to the detour. Since the internal system of this application directs a certain number of users to the shortest route and then to alternative routes, improvements in the internal system could lead to more effective congestion alleviation.

5.4 Proposal for New Route Suggestion Method

The internal system of this study adopts a method where users are first guided to the shortest route. When the capacity of the checkpoints along the shortest route reaches its limit, useful alternative routes are recommended to subsequent users. This method aims to efficiently control the flow of people by providing the most optimal route while preventing excessive concentration on specific routes. In particular, the simulation results in the second condition confirmed that the route selection follows the same priorities as this method. Furthermore, by developing a system that alternates users between the shortest route and the detour, as seen in the third condition, further congestion alleviation is possible.

A specific approach is to dynamically increase or decrease the capacity of surrounding checkpoints in relation to the number of people staying at each checkpoint. In this approach, Dijkstra's algorithm is first used to calculate the shortest route and propose it to users. Then, the total distance of the proposed route and the adjacent checkpoints are recorded, and Dijkstra's algorithm is re-applied to propose a new route by temporarily excluding the current checkpoint. This process is repeated until no further routes can be suggested, calculating the total distance of multiple routes that pass through adjacent checkpoints and reach the destination.

Based on this information, the capacity of each checkpoint is adjusted while considering the total distance to the destination. Specifically, users are preferentially guided to the shortest route, and when the capacity limit is reached, the next most useful route is suggested sequentially. By adopting this method, users can use both the shortest route and the detour route appropriately, avoiding excessive concentration on specific routes.

Moreover, there is a feedback loop between the simulation and the internal system. By modifying the internal system's behavior based on data obtained from the simulation and evaluating whether these changes are appropriate through further simulations, it becomes possible to develop a more effective application. By repeating this process, a system that can withstand real-world conditions can be built, and it will be possible to propose effective routes during actual congestion situations or emergency evacuations.

6 Conclusion

This study developed a new web application aimed at alleviating congestion at tourist destinations and event venues, and its utility and reliability were verified. The experimental results indicated that there were issues with the user interface (UI) design of the application, and it was suggested that improvements should be made to ensure users can appropriately recognize the correct routes they should follow. To help users make proper route decisions, it is necessary to use more distinctive markers.

In the future, a feature will be implemented that visually represents the impact of congestion alleviation in real-time. This feature will clarify the benefits of following the app's instructions, which is expected to encourage more active use of the application by users. The app shows potential in two areas: tourism and evacuation.

In tourism, it can help alleviate overtourism by visualizing the congestion levels at tourist spots and commercial facilities in real-time, promoting distribution among travelers.

In evacuation scenarios, crucial to propose routes that avoid disaster areas, guiding users to evacuation points while ensuring their safety. It is essential for the app to immediately notify users of the congestion at evacuation shelters and provide information about safe routes to prevent panic. If a proposed route leads to a dangerous area, it could become a social issue, so it would be necessary to quickly collect information and grant the building managers the authority to make route suggestions to ensure flexible responses.

Additionally, it is essential to develop a multilingual version of the application for users who do not understand Japanese, particularly for foreign visitors. Since it is crucial to avoid disaster zones, a decentralized simulation should be conducted to verify the app's usefulness in this context.

Acknowledgement

This research was partially supported by Grants-in-Aid for Scientific Research (C) 22K02815 and (B) 23K21016

References

[1] Kimjin Choi, "The Emergence of Overtourism and Issues of Sustainable Tourism De velopment," Kyushu International University Journal of International Economics, vol. 5, 2020, pp. 193–206.

[2] Yukiko Tanimoto, Yoshitaka Tanimoto, "A Study on the Concept of Overtourism in Venice and Its Implications for Japan and Kyoto (1)," Doctoral Dissertation, Kansai Gaidai University, 2020.

[3] Tsuyoshi Nakayama, Yuka Miyaji, Seishi Kato, Nobuhisa Sakurada, Noriyuki Ueda, Ta kayuki Nomura, Kazunori Okayu, Hiroshi Uematsu, Eiji Kimura, "Study on Mobile Phone Applications to Support People with Brain Dysfunction," IEEJ Transactions C (Electronics, Information and Systems), vol. 130, no. 3, 2010.

[4] Yoshihisa Takada, Yoshiharu Fujita, "A Study on the Acceptance Factors of Mobile Data Services among Smartphone Users," Journal of the Institute of Information and Communication Research, 2013.

[5] Daichi Yanagisawa, Katsuhiro Nishinari, "Cellular Automaton Model of Crowd Move ment," Journal of Information Processing, vol. 58, no. 7, 2017, pp. 570–571.

[6] Bagozzi, Richard P., Youjae Yi, "On the Evaluation of Structural Equation Models," Journal of the Academy of Marketing Science, vol. 16, no. 1, 1998, pp. 74–94.

[7] Hopper, Daire, Joseph Coughlan, Michael R. Mullen, "Structural Equation Modelling: Guidelines for Determining Model Fit," Electronic Journal of Business Research Methods, 2008.

[8] Davis, Fred D., et al., "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," Management Science, vol. 35, no. 8, 1989, pp. 982–1003.