User-Generated Voice Navigation Editing System Using Block-Type Visual Programming Language

Daisuke Yamamoto
Nagoya Institute of Technology
Nagoya, Japan.
email:daisuke@nitech.ac.jp

Hiroki Sekiya
Nagoya Institute of Technology
Nagoya, Japan.
email:h.sekiya.976@nitech.jp

Shinsuke Kajioka
Nagoya Institute of Technology
Nagoya, Japan
email:kajioka@nitech.ac.jp

***Abstract*–Voice navigation systems on smartphones have become widespread in recent years. However, existing systems only provide one-way voice navigation based on uniform, machine-generated navigation instructions. It would be beneficial to have tourism navigation and two-way voice interaction, alongside normal navigation, for use in tourist hotspots. It is difficult to automatically generate interactive sentences for use in dialogue with map data alone. Therefore, this study aimed to create an environment in which users with knowledge of a given area can easily create voice navigation content. Google Blockly, a widely used block-type visual programming language, is extended and matched with map nodes (called "spots" in this paper) and links to produce a mechanism for achieving navigation at the spot level. Furthermore, a two-way voice navigation system is achieved by implementing a mechanism to convert from user-generated content to a format for voice interaction systems. The usefulness of the proposed system was evaluated by conducting experiments with a prototype system.**

Keywords-geographic information systems; user-generated content; voice interaction systems; voice navigation.

I. INTRODUCTION

Voice navigation systems on smartphones, such as Google Maps [1], have become widespread in recent years. These systems provide voice prompts for directions and distances at important locations such as junctions. However, existing voice navigation systems only provide one-way navigation based on uniform navigation sentences that are automatically generated from a map database. It would be beneficial for tourism if there were interactive voice navigation systems to supplement normal navigation with additional information in tourist hotspots. Additionally, if additional information is provided when on roads that are difficult to understand or require care when walking, users would be able to reach their destinations safely and without getting lost.

Data on tourism navigation and supplementary navigation are generally not included in map databases. As a result, it is difficult to automatically generate these types of voice navigation from map databases. Therefore, we focused on technology related to User-Generated Content (UGC) [2]. UGC allows users to edit content such as dictionaries and maps, e.g., Wikipedia and OpenStreetMap [3]. Incorporating user-generated mechanisms into voice navigation allows users to edit content based on tourism navigation and supplementary information. As demonstrated by OpenStreetMap, UGC is a technology intimately connected with the field of geographic information systems, and it has great potential.

Therefore, this study aimed to develop a mechanism based on UGC that allows users who are familiar with a given area to easily edit voice navigation scenarios ("voice navigation content") with two-way voice interaction functions. This enables voice navigation from departure to destination where it feels as though a pedestrian is listening to a tour guide.

The following requirements should be met to achieve the above objective.

Requirement 1. The mechanism, based on the concept of UGC, should allow anybody to easily edit two-way voice navigation content.

Requirement 2. In tourist hotspots, the shortest route is not necessarily the recommended route. The user should be able to change the navigation route.

Requirement 3. Pedestrians may get lost; therefore, it must be possible for the pedestrian to still be directed to their destination even if they do not follow the specified route.

In response to the above requirements, the proposed method has the following features.

Feature 1. We propose spot blocks, an extension of Google Blockly [4] matching blocks to map nodes (called "spots" in this paper). This allows editing of voice navigation content using a block-based visual programming language.
Feature 2. The navigation route can be edited easily by changing the combination and layout of spot blocks.
Feature 3. Setting the default state of the spot blocks to the shortest route enables the handling of brand-new routes.

Google Blockly is a technology that is widely used in Scratch [5], a visual programming environment for children.

There have not been many studies attempting to achieve voice navigation content based on UGC, it is a field of research with great potential. The results of our study contribute to the development of research fields related to geographic information systems, voice interaction systems, and user generated content, as well as the confluence of these fields.

In Section 2, we describe the related work. Section 3 describes the proposed method and Section 4 describes the proposed system. Section 5 describes the experimental results. Section 6 concludes.

II. Related Work

Many studies on UGC have been conducted in the field of geographic information systems. For example, the following are studies on methods for collecting, analyzing, and applying geographic information from social media. Giradin et al. proposed a method for collecting spatio-temporal information about tourists by analyzing UGC [6]. Li et al. proposed a system called VisTravel, which collects, visualizes, and makes analyzable the opinions of tourism networks through UGC [7]. Khoshamooz et al. proposed a mechanism for extending the parameters for multi-criteria route planning utilizing UGC [8]. Hu et al. proposed a method for mapping the brand positioning and competitive situation of hotel brands by text mining of UGC [9].

There are several proposals for navigation based on the concept of UGC. Holone et al. proposed a mechanism based on UGC, which weights the routes navigated by the user themselves using a smartphone [10]. Yanagi et al. implemented voice navigation using tagged information that was posted on Twitter [11]. However, these studies used one-way voice navigation, unlike the proposed method which uses two-way voice navigation.

A system that achieves two-way voice interaction is called a voice dialogue system. Many studies have been conducted on voice interaction systems. MMDAgent (used in this study) is a system that combines the functions needed for achieving a voice interaction system, such as voice synthesis, voice recognition, 3D model drawing, and interaction control [12]. Dialogue scenarios can be formatted as Finite State Transitions (FST). Nishimura et al. proposed a framework for easily creating and sharing web-based scenarios for voice interaction systems based on the concept of UGC [13]. Wakabayashi et al. enabled the creation of content for voice interaction by manipulating the state transition diagram using a tablet screen [14]. Additionally, Furuichi et al. used Google Blockly blocks to define voice interaction systems (voice recognition block, voice synthesis block, etc.) [15]. As shown in Figure 1, combining these blocks enables the creation of two-way voice interaction content. In this example in Figure 1, if a user says "Hello," the agent will respond with "Hello" while bowing. If the user asks "What is the like weather today?," the agent will respond with "It is sunny." These studies were conducted on generic voice interaction systems, and they do not deal with navigation. Hayashi proposed a method for editing voice navigation content [16]. The voice interaction content could be edited in the form of state transition diagrams in association with maps by improving upon the work of Wakabayashi et al. in [14].

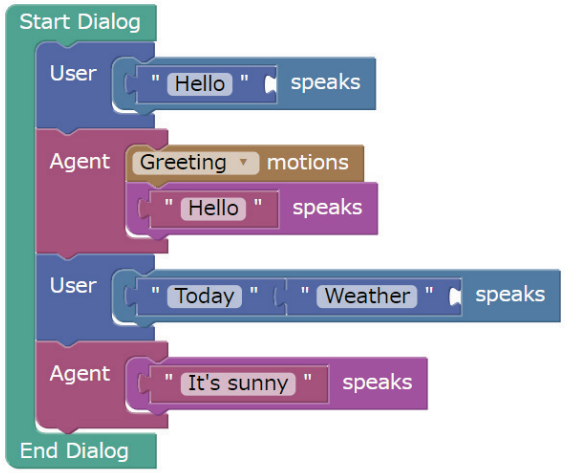

Figure 1. Example of voice interaction content blocks created using Google Blockly.

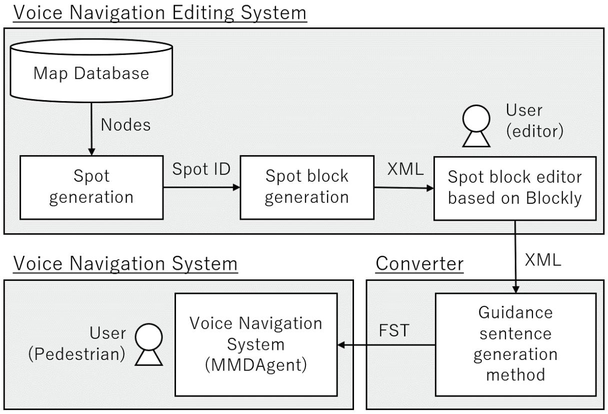
Voice Navigation Editing System
Map Database
Nodes
User (editor)
Spot generation
Spot ID
Spot block generation
XML
Spot block editor based on Blockly
XML
Voice Navigation System
Converter
User (Pedestrian)
Voice Navigation System (MMDAgent)
FST
Guidance sentence generation method

Figure 2. Configuration diagram of proposed system.

III. Proposed Method

In this section, we describe the configuration of the proposed system and the proposed method.

A. System Configuration

Figure 2 shows the configuration of the proposed system. The proposed system was implemented by extending the system of Furuichi et al. [15]. Specific extended functions are as follows. First, in the spot generation function, all intersections in the target area are acquired as spots in the route database. Acquired spots can be confirmed via the map interface. The spot block generator converts each spot into a spot block that is associated with a point on the map interface. By default, the generated spot blocks are connected along the route order from the departure point to the destination. The navigation route can be changed by manipulating the spot blocks using the spot block editing function. Furthermore, voice recognition and voice synthesis blocks can be inserted to enable the creation of two-way voice interaction scenarios. Finally, the navigation sentence generation function converts the information into a format that can be handled by the voice interaction system.

B. *Spot Generation*

In this study, a spot was the smallest unit for navigation and corresponded to an intersection node. As shown in Table I, a spot has an ID, latitude and longitude coordinates, a set of neighboring spot IDs, and the ID of the next spot on the route. Furthermore, the construction of a spot network enables the calculation of the distance and direction between spots, and the acquisition of information needed for navigation.

The procedure that the spot generation function follows is shown below. First, the latitude and longitude of the destination are input, then all intersections within an x km radius around the destination are searched, and all obtained intersections are set as spots. Next, the shortest paths from all spots to the destination spot are determined using Dijkstra's algorithm. The next spot to be navigated to can be determined by finding the shortest route; therefore, = the minimum amount of voice navigation was generated without using the spot block editing function (which will be described later).

C. *Spot Block Editing*

In the method described by Furuichi et al., there are voice recognition and synthesis blocks, but there are no blocks for navigation or location information. Therefore, we propose the spot block as a new block for describing navigation. Spot blocks correspond in a one-to-one manner with map spots.

TABLE I. DEFINITION OF A SPOT.

name	type	description
ID	int	Spot number
Latitude	double	Latitude
Longitude	double	Longitude
NextSpot	int	ID of next spot
NeighborSpot	int[]	IDs of neighboring spots

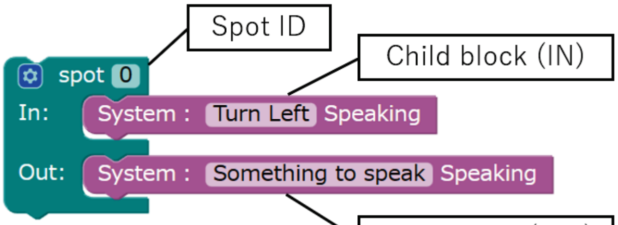
Spot ID
Child block (IN)
spot 0
In:
System : Turn Left Speaking
Out:
System : Something to speak Speaking
Child block (Out)

Figure 3. Example of a spot block.

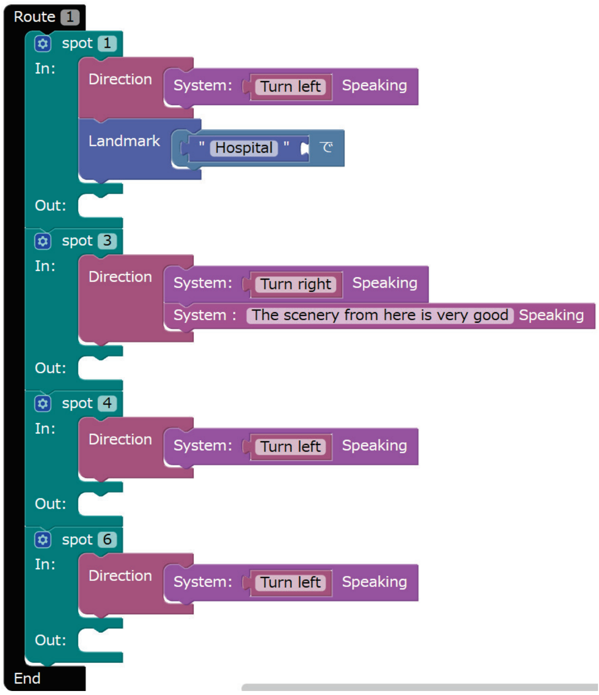
Route 1
spot 1
In:
Direction
System: Turn left Speaking
Landmark
" Hospital "
Out:
spot 3
In:
Direction
System: Turn right Speaking
System : The scenery from here is very good Speaking
Out:
spot 4
In:
Direction
System: Turn left Speaking
Out:
spot 6
In:
Direction
System: Turn left Speaking
Out:
End

Figure 4. Example of linking spot blocks.

As shown in Figure 3, the spot block has a corresponding spot ID, a child block that fires when a user approaches the spot, and a child block that fires when the user leaves the spot. The spot blocks can be linked back-to-back with each other to determine the order of navigation.

The user can link spots to create their own route. Additionally, child blocks can be added to enable voice interactions at spots. Akin to those in the method from Furuichi et al. blocks that conduct complex interaction control (control blocks and function blocks) were used in the present study as well.

Spot blocks that are not linked are called orphan spot blocks. An orphan spot block works as a single spot block and fires when a user approaches the spot. The intersection to be navigated to next is determined based on the shortest route. The introduction of the orphan spot blocks enables navigation to the destination even if a route other than the one set in advance by the user was taken.

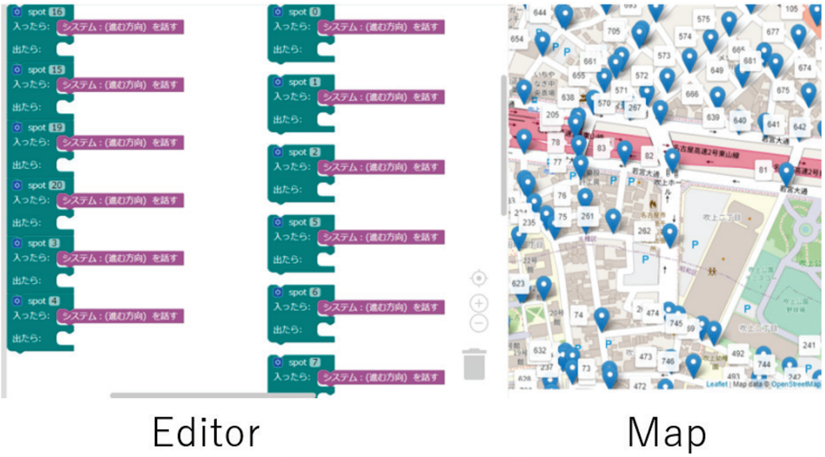
spot 16
入ったら: システム：(進む方向) を話す
出たら:
spot 15
入ったら: システム：(進む方向) を話す
出たら:
spot 19
入ったら: システム：(進む方向) を話す
出たら:
spot 20
入ったら: システム：(進む方向) を話す
出たら:
spot 3
入ったら: システム：(進む方向) を話す
出たら:
spot 4
入ったら: システム：(進む方向) を話す
出たら:
spot 0
入ったら: システム：(進む方向) を話す
出たら:
spot 1
入ったら: システム：(進む方向) を話す
出たら:
spot 2
入ったら: システム：(進む方向) を話す
出たら:
spot 5
入ったら: システム：(進む方向) を話す
出たら:
spot 6
入ったら: システム：(進む方向) を話す
出たら:
spot 7
入ったら: システム：(進む方向) を話す
Leaflet | Map data © OpenStreetMap
Editor
Map

Figure 5. Prototype system edit screen.

Figure 4 shows an example of linked spot blocks. In this example, navigation is set in the order of spot ID 1, 3, 4, and 6, and when the spot ID approaches 3, supplementary interaction becomes possible with the voice prompt, “The view from here is very nice.”

D. Navigation Sentence Generation

In the navigation sentence generation function, the block set generated by the spot block editing function is converted into a suitable format for the voice interaction system. The voice interaction system used is the Japanese voice interaction system construction toolkit MMDAgent.

The method for generating navigation sentences is the same as that of Furuichi et al., so the function for the voice navigation is explained here.

1) Generation of navigation sentences relating to directions and distance

Directing a pedestrian requires communicating the direction to turn (turn right, turn left, go straight, etc.) and the distance to the next spot. Therefore, if the ID of the spot where the pedestrian is currently located is S_1, the ID of the next spot to advance is S_2, and the ID of the immediately preceding spot for the pedestrian is S_3, then the direction A and distance D are obtained by the following procedure.

1. Let V_1 be the vector from S_3 to S_1.
2. Let V_2 be the vector from S_1 to S_2.
3. A is determined using the angle between V_1 and V_2.
4. Let the length of V_2 be the distance D.

However, if S_3 cannot be defined, such as at the starting point, then A also cannot be defined. In this case, navigation is provided using the direction of vector V_2 (e.g., north, south, east, or west).

2) Navigation route change function

The navigation route change function changes the route according to how the user interacts with the spot block. As previously mentioned, the user can determine the next spot to head toward by combining spot blocks. This is processed as follows according to the spot block operation event.

1. When spot block S_2 is connected to spot block S_1, the NextSpotID of the spot is the ID of the spot corresponding to S_2.
2. When the spot block S_2 is separated from spot block S_1, the NextSpotID of the spot corresponding to S_1 is set as the initial spot ID based on the shortest route.

Here, S_1 comes before S_2.

E. Prototype System

Figure 5 shows a prototype system based on the proposed method. In the prototype system, spots are displayed as blue pins on the map. The user edits the spot block using the spot block editing function while referring to the displayed spot. The edited blocks are automatically converted into the FST format for voice interaction systems, which enables the creation of voice navigation scenarios. Additionally, the created navigation scenario operates on a smartphone, and interactions related to voice navigation are possible according to the location and interaction content.

IV. Experimental Results

We conducted three experiments to evaluate the effectiveness of the proposed method. OpenStreetMap was the source of the map data.

A. Experiment on Navigation Sentence Generation

The proposed system should provide simple voice navigation based on Dijkstra’s algorithm without needing to use the spot block editing function. Here, we verify if this function is operating effectively.

We use the prototype system to determine 10 destination points, acquire intersections, and place spots in the range of 500 m, 1 km, 1.5 km, and 2 km around the destination. We verify at this time whether it is possible to reach the destination by the route that is automatically obtained by Dijkstra’s algorithm from an arbitrary spot. Table II shows the average arrival rate for each distance. The arrival rate exceeded 98% for all distances.

These results showed that a minimal amount of voice navigation was necessary even without user editing. This allows the user to concentrate on editing such as tourism navigation and supplementary navigation.

B. *Experiments on the Editing Interface*

Next, we tested the interface of the spot block editing function based on the block-type visual programming language.

We gave eight university students the task of creating an interactive route for voice navigation between two points using both the prototype system and a conventional method. We measured the time for creation and conducted a survey after the experiment. We used the system usability scale (SUS) [17] as a usability evaluation scale. The SUS score has a maximum of 100 points and an average of 68.

TABLE II. AVERAGE ARRIVAL RATE FOR EACH AREA.

	500 m	1 km	1.5 km	2 km
Average arrival rate (%)	98.75	99.01	99.35	99.81

TABLE III. EXPERIMENTAL RESULTS ON EDITING INTERFACE.

	Hayashi's method	Proposed method
SUS Score	45.5	61.5
Creation time	170 seconds	157 seconds
Waypoint	4.8	6.2

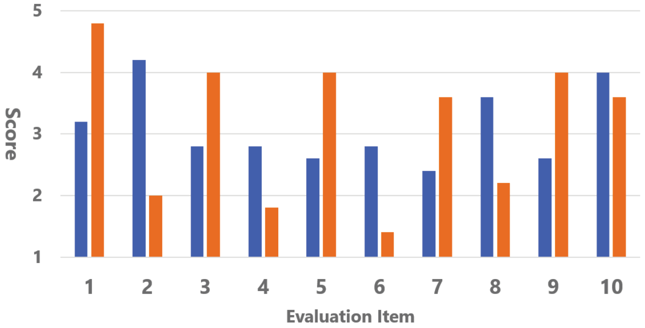

Figure 6. Details of SUS score relating to editing interface. Left: Hayashi et al. method, right: proposed method.

The systems to be compared are the following two methods.

1. Proposed method.
2. Method of Hayashi [16].

In the prototype system, there were 44 spots including the start and destination points.

Table III shows the experimental results. Figure 6 shows the average score for each item of the SUS score. In this figure, for odd-numbered items, the higher the number, the higher the rating. For even-numbered items, the lower the number, the higher the rating.

The proposed method scored the highest across all metrics of the SUS score. The conventional method scored 45.5, and the proposed method scored 61.5. This result suggests that the block-type description format of the proposed method is also effective in creating voice navigation scenarios. However, this score is lower than the score of the interaction scenario editing system by Furuichi et al., which attained 67.8. We think the reason for this is that the operation of the proposed system has become more complex due to the added functions for navigation. Therefore, further improvement of the interface is likely needed.

C. *Navigation Scenario*

Next, we verified the usefulness of the voice navigation scenarios generated by the prototype system.

Nine university students participated in this experiment. The systems to be compared were:

1. Automatically generated simple voice navigation (assuming Google Maps).
2. Method of Hayashi et al. [16].
3. Proposed method.

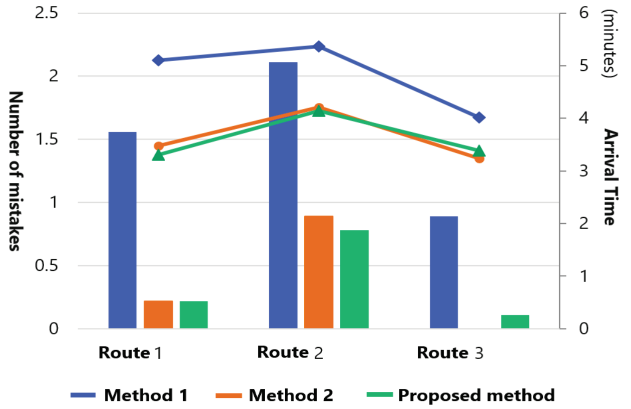

Figure 7. Results of experiments relating to voice navigation. The bar graph shows the number of mistakes, and the line graph shows the arrival time.

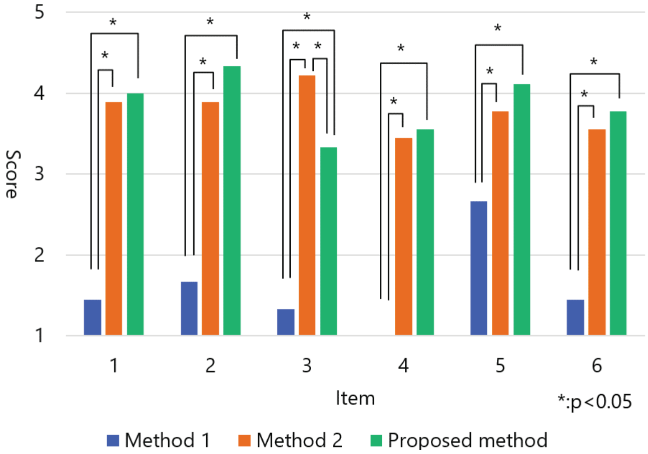

Figure 8. Results of questionnaire relating to voice navigation. * indicates a significant difference.

The automatically generated navigation sentences only gave instructions to turn at intersections. The navigation sentences generated by the method of Hayashi et al. supplemented the turning directions with landmarks. The navigation sentences of the proposed method included an interactive explanation of parts where users are likely to make a mistake to supplement the turning directions. The subjects used all three methods. The subjects were asked to record their arrival time and the number of times they made mistakes, and to respond to a five-level questionnaire that contained the following six items. Each level indicates how strongly the participant agreed with the phrase, a 5 indicated strong agreement.

1. Satisfaction with voice navigation.
2. I felt uneasy at corners.
3. I felt uneasy on the road.
4. I want to use this again.
5. The system speaks and responds in an appropriate manner.
6. Navigation is easy to understand.

Figure 7 shows the time required for each navigation and the number of times the wrong road was taken, and Figure 8 shows the results of the questionnaire.

The proposed method and that of Hayashi obtained better results than automatically generated sentences in the questionnaire and the experiments. For example, subjects often took the wrong turning when roads were narrow or at consecutive intersections. Such situations required nuanced navigation instructions such as “turn left in front of the cafe,” instead of just “turn left at the cafe.” Additionally, we think that the conventional method’s arrival time was affected by the drivers checking carefully at turnings due to a lack of confidence about being on the right route. In the questionnaire, the conventional method and the proposed method scored significantly higher than the automatically generated method. In particular, items 2 and 3, uneasiness at junctions and on the road, were significantly reduced, which may have led to a higher overall satisfaction and an easier understanding of the navigation.

V. CONCLUSION

In this paper, we detailed the creation of a mechanism, based on the concept of UGC, that allows users with knowledge of a given area to easily edit voice navigation scenarios with two-way voice interaction functions. This allows for voice navigation from beginning to destination that feels like being guided by a tour guide. We extended Google Blockly by adding functions such as the spot block function.

We developed a prototype system based on the proposed method and discussed the effectiveness and problems of the proposed method. In the experiment relating to automatic generation of directions, the arrival rate from any spot in a range of 2 km centered on the target point was 98% or more. In the experiment relating to the editing system, the proposed method scored higher in the SUS and all questionnaire items; therefore, we conclude that the directions generated by the proposed method were easy to follow. When the directions were tested by test participants, it was found that they did not become anxious when using the voice navigation of the proposed method. Future work should include improving the UI of the spot block editing function and integrating multiple voice navigation content.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Numbers JP19H04115 and JP21K19766.

REFERENCES

[1] Google Maps, https://www.google.com/maps/. [retrieved: 03, 2023]

[2] J. Krumm, N. Davies, and C. Narayanaswami, "User-Generated Content," in IEEE Pervasive Computing, vol. 7, no. 4, pp. 10–11, Oct.-Dec. 2008. doi: 10.1109/MPRV.2008.85. OpenStreetMaps.

[3] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," in IEEE Pervasive Computing, vol. 7, no. 4, pp. 12–18, 2008. doi: 10.1109/MPRV.2008.80.

[4] M. Seraj, E. S. Katterfeldt, K. Bub, S. Autexier, and R. Drechsler, "Scratch and Google Blockly: How Girls' Programming Skills and Attitudes are Influenced," Proceedings of the 19th Koli Calling International Conference on Computing Education Research, No. 23, pp. 1–10, 2019. https://doi.org/10.1145/3364510.3364515

[5] M. Resnick et al. "Scratch: programming for all," Communication of ACM, vol. 52, no. 11, pp. 60–67, 2009. https://doi.org/10.1145/1592761.1592779

[6] F. Girardin, F. Calabrese, F. D. Fiore, C. Ratti, and J. Blat, "Digital Footprinting: Uncovering Tourists with User-Generated Content," in IEEE Pervasive Computing, vol. 7, no. 4, pp. 36–43, Oct.-Dec. 2008. doi: 10.1109/MPRV.2008.71. Giradin.

[7] Q. Li, Y. Wu, S. Wang, M. Lin, X. Feng, and H. Wang, "VisTravel: visualizing tourism network opinion from the user generated content," Journal of Visualization, vol. 19, pp. 489–502, 2016. https://doi.org/10.1007/s12650-015-0330-x.

[8] G. Khoshamooz and M. Taleai, "Multi-Domain User-Generated Content Based Model to Enrich Road Network Data for Multi-Criteria Route Planning", Geogr. Anal., vol. 49, no. 3, pp. 239–267, 2017. https://doi.org/10.1111/gean.12124.

[9] F. Hu and R. H. Trivedi, "Mapping hotel brand positioning and competitive landscapes by text-mining user-generated content," Int. J. of Hosp. Manag., vol. 84, 102317, 2020. https://doi.org/10.1016/j.ijhm.2019.102317.

[10] H. Holone, G. Misund, and H. Holmstedt, "Users Are Doing It For Themselves: Pedestrian Navigation With User Generated Content," Proceeding of the 2007 International Conference on Next Generation Mobile Applications, Services and Technologies, pp. 91–99, 2007. doi: 10.1109/NGMAST.2007.4343406.

[11] A. Lee, K. Oura, and K. Tokuda, "Mmdagent—A fully open-source toolkit for voice interaction systems," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 2013, pp. 8382-8385, doi: 10.1109/ICASSP.2013.6639300.

[12] T. Yanagi, D. Yamamoto and N. Takahashi, "Development of mobile voice navigation system using user-based mobile maps annotations," 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS), Las Vegas, NV, USA, 2015, pp. 373-378, doi: 10.1109/ICIS.2015.7166622.

[13] R. Nishimura, D. Yamamoto, T. Uchiya, and I. Takumi, "MMDAE: Dialog scenario editor for MMDAgent on the web browser," ICT Express, vol. 5, no. 1, pp. 47–51, 2019. https://doi.org/10.1016/j.icte.2018.03.002.

[14] K. Wakabayashi, D. Yamamoto, and N. Takahashi, "A Voice Dialog Editor Based on Finite State Transducer Using Composite State for Tablet Devices," Studies in Computational Intelligence, vol. 614. pp.125-139, 2015. https://doi.org/10.1007/978-3-319-23467-0_9

[15] M. Furuichi, D. Yamamoto, and N. Takahashi, "Voice Interaction Scenarios Editor with Block-based Visual Programming Facilities," Transaction on IPSJ, DCON, vol. 8, No.2, pp.1–15, 2020. (in Japanese)

[16] K. Hayashi, "Voice Dialog Scenario Editor Based on Spatial Extension of Finite State Transducer, " Master thesis, Nagoya Institute of Technology, 2019. (in Japanese)

[17] A. Bangor, P. T. Kortum, and J. T. Miller. "An empirical evaluation of the system usability scale," Int. J. Hum. –Comp. Int., vol. 24, no. 6, pp. 574–594, 2008.