SIM - Smart Interactive Map with Pointing Gestures

Hoang-An Le
Advanced Program in Computer Science
University of Science
Ho Chi Minh city, Vietnam
Email: lhan@apcs.vn

Khoi-Nguyen C. Mac
Advanced Program in Computer Science
University of Science
Ho Chi Minh city, Vietnam
Email: mcknguyen@apcs.vn

Truong-An Pham
Advanced Program in Computer Science
University of Science
Ho Chi Minh city, Vietnam
Email: ptan@apcs.vn

Vinh-Tiep Nguyen
Faculty of Information Technology
University of Science
Ho Chi Minh city, Vietnam
Email: nvtiep@fit.hcmus.edu.vn

Minh-Triet Tran
Faculty of Information Technology
University of Science
Ho Chi Minh city, Vietnam
Email: tmtriet@fit.hcmus.edu.vn

Anh-Duc Duong
University of Information Technology
Ho Chi Minh city, Vietnam
Email: ducda@uit.edu.vn

***Abstract*—Information kiosks, building directories, and area maps are useful for visitors to explore a new large place. Although several digital information or map systems have been equipped with touchscreens, many existing systems are still regular non-interactive screens to display static or dynamic information. This motivates the authors to propose an efficient and economic solution to transform existing non-interactive information kiosks and maps into natural interactive systems that can accept pointing gestures, a common type of gestures in communication between people. The authors' vision-based method exploits the visual geometrical characteristics of pointing gestures thus is independent from skin color and can recognize pointing gestures with various pointing objects, such as fingers, bare hands, hands with gloves, or any arbitrary pointing objects. Experiments show that a single regular processor can process up to 10 interactive maps with less than 17ms with the average accuracy of more than 90%. Furthermore, by using template matching, a user can get the detailed guidance for the best route into his/her mobile device just by capturing the information being displayed in the interactive map.**

***Keywords*-interactive map, pointing gesture, human computer interaction.**

I. Introduction

One of the common difficulties for visitors in a new large area such as an airport, a metro station, a museum, or a shopping mall, is to find a desired location and the route to that place. Therefore, directions and maps are placed at various corners to assist visitors. Such information boards can be in the form of traditional metal or wooden map boards with static information, interactive or non-interactive information screens with dynamic content (see Fig.1).

Interactive information screens provide visitors with not only static useful geographical information but also helpful functions such as place searching, path finding, etc. People can interact with such systems in a traditional method with regular mice and keyboards or in a more natural way, i.e. touch screens. It is obvious that users can enjoy exciting experiences when interacting with an information screen with their natural gestures. However there are still a large number of existing information screens that do not support interactions and it will be expensive to replace all existing non-touch displays with touchscreens, especially large screen displays.

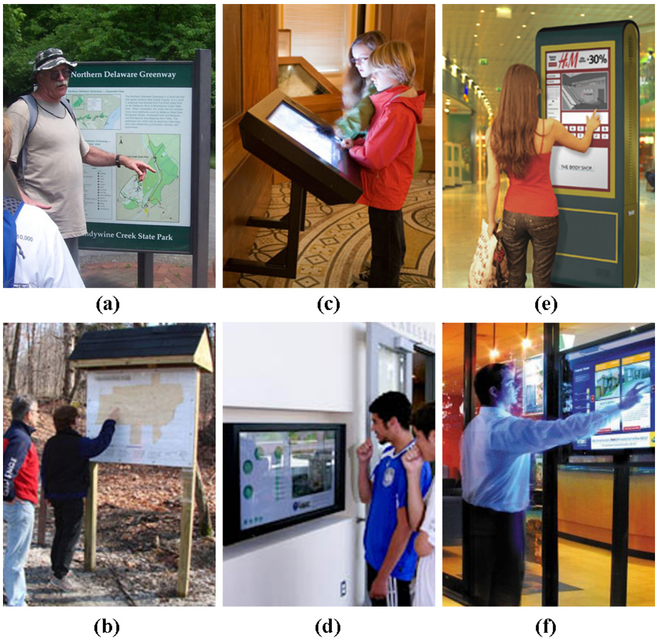


Fig. 1. Examples of information boards

In this paper, the authors propose Smart Interactive Map (SIM), an economic solution to transform existing non-interactive information screens into a smart system that can accept and understand pointing gestures, one of the most common types of gestures in communication. Each information station has a non-touch screen and a regular webcam to capture visitors' gestures on that screen. A single processor is used to process multiple information stations. By exploiting the visual properties of pointing gestures, the method can recognize

978-0-7695-4721-3/12 $26.00 © 2012 IEEE
DOI 10.1109/IHMSC.2012.178

pointing gestures with various pointing objects, e.g. fingers, bared hands, hands with gloves, pens, etc. Furthermore, the method can solve the limitation of existing methods to depend on skin color [1], [2], [3], [4] or pre-defined colors [5].

Another interesting feature of SIM is that it also allows a visitor to download a suggested best path from an information screen into his or her mobile device in an innovative way. A mobile device takes a photo of the best path displayed on an information screen and send this photo to the server of SIM in that area. The SIM server then matches this photo against the current visual displays of all stations under its control and sends back to the mobile device the corresponding detailed navigation plan. This function is a bridge between and ubiquitous navigation assisting systems with mobile devices.

The content of this paper is as follows. The background and related works are briefly reviewed in Sec. II. In Sec. III, the authors present the proposed method to transform existing non-interactive information displays into interactive systems that support pointing gestures. Then in Sec. IV, the authors propose the mechanism to use a mobile device to capture the best route displayed on the screen by template matching. Experiments are presented and discussed in Sec. V. Finally the conclusions are presented in Sec. VI.

II. Background and Related works

Because of the usefulness of guidance systems in helping newcomers to navigate within a new area, different research topics and projects have been developed to improve system's usability. The multimodal interaction for pedestrians [6] by Jöst et al. proposes an evaluation study about guidance systems using the mobile SmartKom system. In [7], the pioneering indoor CyberGuide system takes the touch interaction approach by using touching sensors of handheld devices. The Campus Guidance System for International Conferences [8] by Ricky Jacob et al. is a guidance system implemented based on OpenStreetMap, a Web-Based Maps API, that supports both English and Chinese languages. The system is able to generate the shortest pedestrian paths using indoor corridors and outdoor pavements to navigating within the National University of Ireland Maynooth campus.

The main goal of Human-Computer Interaction (HCI) is to study and develop means of interaction between users and applications in the most natural way. Therefore various methods for interaction with users' hands have been proposed[9], [1], [2]. One of the most common problems in these methods is to detect hands. Because of the simplicity and invariance of skin color in comparison with hand shapes, most methods exploit the skin color to detect hands for interaction. Different skin color models have been proposed [3], [10] and skin color is widely used in the first step to detect and segment users' hands [1], [2], [4], [11]. However, these color-based approaches to detect hands cannot solve the practical situations when users wear gloves.

Besides bare hands, several other methods use color markers for interaction [5]. In these methods, users are required to wear some predefined colored markers [5] or have to pass through a preliminary learning process [4].

In this paper, the authors take a different approach to exploit special properties of pointing tips of pointing objects for interaction. Therefore the proposed method does not depend on pre-defined colors and can process various pointing objects, such as fingers, bare hands, hands with gloves, pens, etc..

III. Proposed System

A. System Setup

The proposed system consists of 3 modules: a regular CRT or LCD screen to display information, a webcam to capture a user's gestures, and a processor to analyse and interprete actions. In reality, a normal information display can be in one of the three orientations (as in Figure 2). For each layout, a regular webcam is placed above looking into the screen so that the webcam can capture the whole screen and all users' gestures on the screen.

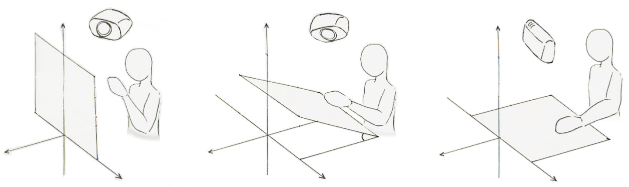


Fig. 2. The three orientations of regular information display and system setup

Instead of having each station map a separate processor, the information captured at a map station is assembled at a local server where the analyzing and detection is parallel carried out. Because map stations are scattered through a wide area, each server will be in charge of a cluster of stations depending on the landscape geography.

B. Interaction method

The process of detecting and tracking the fingertip's movement is devided into four phases working successively. They are the screen calibration, background subtraction, tip detection and system optimization.

1) Calibration: Because of the relative position between a webcam and a screen, a screen image captured by a webcam is usually distorted, no longer a rectangular shape but a trapezoid or quadrilateral (see Figure 4). Thus it is necessary to have a calibration step to get the correct screen coordinate from a detected position in a webcam's frame. In addition to, since the interaction can only take place on the screen surface, the concentration of camera only on the screen region not only helps the system to narrow down the processing region but also bypass noises appearing beyond the region of interest.

Let $\mathcal{I} = \{I_1, I_2, ..., I_n\}, \mathcal{I} \subset \mathbb{R}^2$ be original images shown in displays and $\mathcal{F} = \{F_1, F_2, ..., F_n\}, \mathcal{F} \subset \mathbb{R}^2$ be their images captured by webcams. Let $\mathbf{u}$ be the vector at a point $I \in \mathcal{I}$

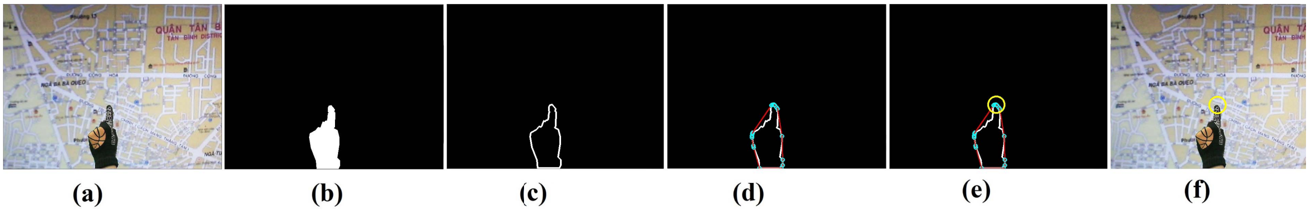


Fig. 3. Pointing tip detection method. (a) The pointing tip; (b) segmentation phase: a foreground mask of the pointing object; (c) pointing object contour; (d) the convex hull of the object contour with points passed through protrusion level k; (e) the farthest point among points in convex hull; (f): the pointing tip is detected

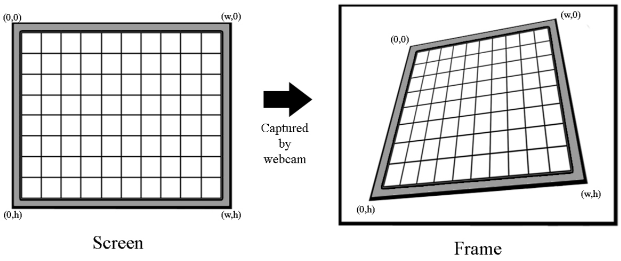


Fig. 4. The keystone effect

and and $\mathbf{v}$ be the vector at a point $F \in \mathcal{F}$, in homogeneous coordinate, we have:

$$\mathbf{u} = \begin{bmatrix} x_I \\ y_I \\ 1 \end{bmatrix}, \mathbf{v} = \begin{bmatrix} x_F \\ y_F \\ 1 \end{bmatrix}$$

A homography $\mathbf{T}$ is a transformation that maps a point from an image in the display to its correspondence in a frame captured by the webcam, i.e,

$$\mathbf{T} : \mathcal{I} \rightarrow \mathcal{F}$$

$$\mathbf{u} \mapsto \mathbf{v},$$

Thus, we have

$$\mathbf{v} = \mathbf{T} \cdot \mathbf{u} \tag{1}$$

Let

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} = [\mathbf{t}_1 \quad \mathbf{t}_2 \quad \mathbf{t}_3]$$

The equation 1 is equal to

$$\begin{bmatrix} -x_I & -y_I & -1 & 0 & 0 & 0 & x_I y_F & x_F y_I & x_F \\ 0 & 0 & 0 & -x_I & -y_I & -1 & x_I y_F & y_I y_F & y_F \end{bmatrix} \begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \mathbf{t}_3 \end{bmatrix} = \mathbf{0} \tag{2}$$

By this way, each pair of points I, F gives 2 equations as in equation 2, thus, to solve for 9 parameters of $\mathbf{T}$ it requires at least 4 correspondences on the 2 images. Utilizing the resemblance of the local features of the two images $\mathcal{I}$ and $\mathcal{F}$ the method applies the SURF algorithm [12] to extract the interest points from the two images. The points in the two sets are then tried to match with each other by RANSAC paradigm [13]:

- The system chooses randomly K pairs of point from the 2 sets to construct a transformation matrix.
- The matrix created is tested with other keypoints for reliability; a probability number is calculated based on the number of correctly matched interest points
- The process is iterated until the probability number is accepted.

After the transformation matrix is available, as long as the relative distance of the camera and the display is kept unchanged, any position in the captured frame can be mapped correctly to a position on the display by multiplying the position coordinate with the derived transformation matrix.

2) Background subtraction: In screen images captured by a webcam, a user's hand or an arbitrary pointing object is the only moving object over a background. Therefore the authors use background subtraction to segment a user's hand or a pointing object. However, the background is not always a constant image but contains two parts: the static environmental region around the screen and the dynamic content in the screen which is changed when a user triggers an event. Thus the background training is carried out not only in the initialization phase but right after an event is generated by a user. It should be noticed that the system does not need to continuously update the background but only re-train the background when the system accepts a user's event and changes the visual content of a display. In this paper, the authors decide to use codebook algorithm [14] which can deal with illumination change and moving-background training.

When re-training the background, some portion of the new background may be occluded by a user's pointing object. To solve this issue, the authors correct the occluded area in the background by applying the homographic transform on the current screen content and mapping the corresponding region into the background.

Instead of having the whole foreground object extracted from the background subtraction, the authors obtain only the foreground mask, binary images whose black pixels belong to the background and white ones belong to the foreground object, for convenience in later image processing (see Fig.3b).

3) Pointing Tip Detection: The pointing tip detection process requires a preliminary step to extract the detected region's contour. The boundary extraction is done by taking the difference between a binary image and its erosion (see Fig.3c). In SIM, the authors use a 3×3 matrix so that the boundary is $1-$pixel thickness.

Based on the special geometrical shape of pointers' tips, that is the protrusions of the objects, to detect the pointing tip the authors use the the peak-valley detection by Segen and Kumar [11]. A point $P_i = (x_i, y_i)$ on the convexhull of the contour is considered as a pointing tip candidate if it has the z_i value greater than a positive threshold V_{thr} (see Fig.3d).

Let $\mathbf{p}^-$ and $\mathbf{p}^+$ be the two vectors, defined by:

$$\mathbf{p}^- = (p_1^-, p_2^-) = \overrightarrow{P_{i-k}P_i},\ and\ \mathbf{p}^+ = (p_1^+, p_2^+) = \overrightarrow{P_iP_{i+k}}$$

The z_i value is the third component of the cross product of $\mathbf{p}^-$ and $\mathbf{p}^+$:

$$z_i = p_1^- p_2^+ - p_2^- p_1^+$$

Because people tend to stretch their arm in order to reach for something, the farthest candidate point computed from the frame edge that the object had appeared will be detected as the pointer tip(see Fig.3e).

4) Optimization phase: The users' purpose of using the system is to help themselves in finding information. So usually, the moving speed of of pointing action is not fast or getting outside the region of interest. Thus, within a small interval of $10s$, the pointing tip movement is assumed to be linear model. On this manner, the Kalman filter is applied to correct the tip detection in order not only to accelerate the system performance but prevent the system from false fingertip's recognition in certain cases such as occlusion, cluttered background, interaction of multi users. The method tracks the movement of fingertip by constructing a linear model for the movement and estimating the position of fingertips. According to [15], the system get the predicted fingertips location from previous knowledge, then with the newly detected one, it evaluates the optimal location and update errors to help the prediction in the future to be more accurate.

C. Route Finding

In any general areas, there are several regions, each with some relationships with the others, i.e. a shopping mall has several connected floors; an amusement park has several resort sections, some of them may or may not directly lead to the others. On this manner, the authors propose to construct a map system consisting of several sub-maps, i.e. floors of an shopping center, each corresponds for a region in a large scale area and contains several sections, i.e. shops of a mall's floor. Each sub-map and section may or may not have relationships with others. In the scope of this paper, the relationships between two arbitrary sections is defined by if there is any route between them, how many routes connecting them together, and for each route, how much time it costs, the distance, and the means of transportation, which varies in different areas such as on foot, stairs, lifts, escalators, or cruises, to travel back and forth between them.

In reality, the transition information between arbitrary locations changes over time. For instance, whether an elevator is full, a certain route is temporarily out of order, or a golf cart is out of battery, etc. The information, therefore, is dynamic data updated automatically. As a result, if a user checks the relationship information at two different times, or two users check the information at the same time, they might get different results. Hence, the system's output has to be updated real-time and offer users with the travel plan with lowest possible cost at specific times. Moreover, the suggested routes should guarantee to reduce overlap occurrences between suggested routes and thus increase the number of possible flows in an area.

IV. Mobile Devices Application

This application is an extra function of the system that helps users to be more convenient in navigating with the result route from an SIM station. The application helps users to get not only the found route on their smart phone but also with augmented information such as GPS location, distance and travelling time, etc.

The biggest concern of the application is how to download the result route from the station's display to an individual devices. Instead of requiring users to connect their phone to the station by a cable or similar things, the authors propose capturing the displayed result route using the device's camera (as in Figure 5) and getting the relevant information from a local server. After being started, an SIM client application can automatically find and connect to a server in the local area. The captured image is sent the to server where the analyzing process is carried out: the captured image is then tried to match with the display at each map station in the area. After successfully matching, the relevant information at the corresponding station will be sent back to the users' device.

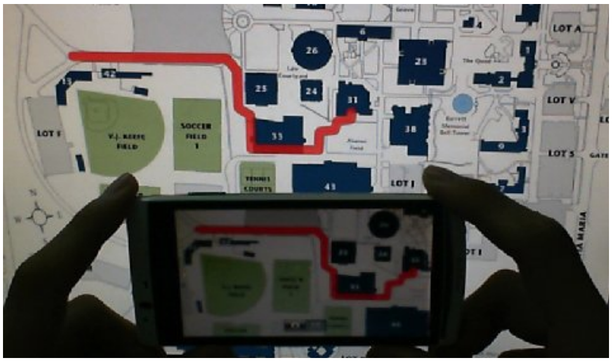


Fig. 5. Getting suggested route via mobile device

Similar to the screen calibration described in Sec III-B1, the matching between the captured image and a station display is processed in 2 steps:

- The scale- and rotate- invariant keypoints of each image is extracted using the SURF technique [12].

- From the two set of keypoints, the RANSAC method [13] is applied to find the transformation matrix. The matching quality is evaluated based on the number of outliers resulting from the derived matrix. The station that has the smallest amount of outliers below a predefined threshold is chosen.

By this way, the application on users' devices are required to directly connect with the map station nor bare all the complex computing but only sends the captured image to a server, and receives the result.

V. Experimental Result

A. System Accuracy

The objective of this experiment is to assess the system's accuracy for specific types of pointing object. As a result, 4 groups of dataset are used as test scenarios, corresponding to different types of pointing objects, namely: (a) bare hand, (b) hand with glove, (c) pen, and (d) glasses, as shown in Fig. 6. The first group illustrates normal interaction situations since people tend to use their bare hands to interact with the system. The second group demonstrates the situations when the hand's color is abnormal, i.e. different from human's normal skin color. The third group indicates the situations that users use extra objects for pointing. The final group indicates situations where the shapes of pointing objects are deformable. For each group the authors process 1000 frames from video clips with the frame rate of 25 frames/ms.

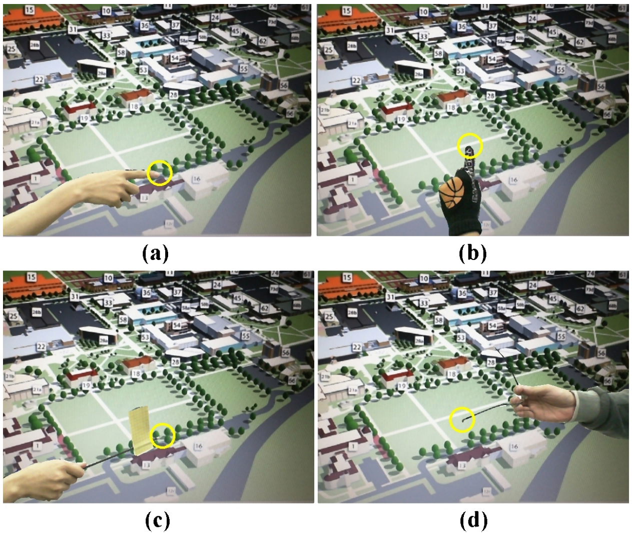
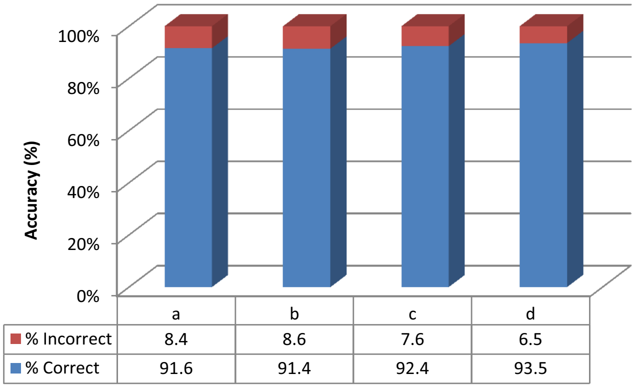

(a) (b) (c) (d)

Fig. 6. Four kinds of test scenarios used in the experiments: (a) bare hand, (b) hand with glove, (c) extra pointing object, and (d) glasses

Figure 7 illustrates the result of accuracy experiment, where the blue parts and red parts respectively mean the correct and incorrect percentage of the result within a group, the horizontal axis is the name of each group, and the vertical axis is the accuracy in percentage terms.

The accuracy is computed based on the number of correctly detected frames. On the whole, the overall accuracy has the mean of 92.2% and standard deviation of 1.0%. Therefore, the system is able to produce highly accurate and stable result, which means the system can satisfy different situations in real life.



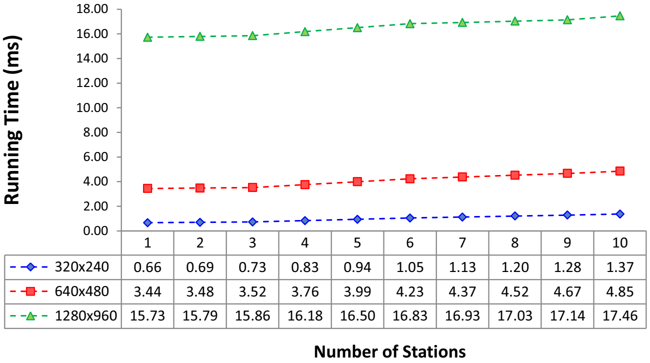
	a	b	c	d
🟥 % Incorrect	8.4	8.6	7.6	6.5
🟦 % Correct	91.6	91.4	92.4	93.5

Fig. 7. System Accuracy's Comparison among Different Pointing Objects

B. System Performance

In this experiment, the test cases are reconstructed and classified based on the number of stations, which varies from 1 to 10. In each test case, the experiment is conducted on 3 different resolutions: 320 × 240, 640 × 480, and 1280 × 960. The horizontal axis represents the number of stations while vertical axis describes the running time in milliseconds.

From the experimental results in Fig. 8, it is clear that the average running time slightly increases when the number of stations increases. Furthermore, the average running time has the tendency to linearly increase when the total number of pixels in each frame increases. It should be noticed that even with 10 map stations having HD resolution, the process time with a single regular processor is 17.46 ms. This allows the proposed system to be applied in reality.



	1	2	3	4	5	6	7	8	9	10
320x240	0.66	0.69	0.73	0.83	0.94	1.05	1.13	1.20	1.28	1.37
640x480	3.44	3.48	3.52	3.76	3.99	4.23	4.37	4.52	4.67	4.85
1280x960	15.73	15.79	15.86	16.18	16.50	16.83	16.93	17.03	17.14	17.46

Fig. 8. System Performance with Different Number of Stations and Resolutions

VI. Conclusion and Future Works

In this paper, the authors propose the system SIM. This system allows transforming usual display maps into fully interactive systems using a common gesture: pointing. The proposed system has the advantages of being natural and friendly to users. It does not depend on any predefined color or shape, but provides users with a high level of flexibility by supporting the pointing action in any means such as fingers, hands, hands with gloves, pens, sticks, glasses, etc.

Besides the system can run in real-time even with only one single processor operating on multiple map stations. By experiments, by using just one regular CPU, up to 10 stations can be processed in real-time criteria. In addition, the system is able to find the appropriate routes for users, depending on the state of transportation, for instance, the level of crowdedness. Thus, users' received results are not fixed in every situation, but instead are dynamic and able to adapt to current situations, e.g. when a lift or elevator has too many users, the system would automatically suggest users another optimal path to arrive at the chosen destination.

In the future, to further develop the system, the author will study and focus on the problems of vehicle routing or load balancing. Hence, users at different places would be suggested with appropriate travel plans that can balance the overall work load and means of transportation, such as lifts, elevators, escalators, cruisers, etc. Moreover, this system also has an extra function allowing users to easily receive the suggested route by using mobile devices.

Acknowledgement

This research is supported by the research funding from the Advanced Program in Computer Science, University of Science, Vietnam National University of Ho Chi Minh City. Besides, the authors would like to give the special thanks to professor Vu Duong, JVN institute, Vietnam National University - Ho Chi Minh city for all the valuable courses and scientific advices during the research.

References

[1] A. Y. Dawod, J. Abdullah, and M. J. Alam, "Fingertips detection from color image with complex background," in *2010 The 3rd International Conference on Machine Vision*, ser. ICMV 2010, 2010, pp. 88–96.

[2] S.-H. Choi, J.-H. Han, and J.-H. Kim, "3d-position estimation for hand gesture interface using a single camera," in *Proceedings of the 14th international conference on Human-computer interaction: interaction techniques and environments - Volume Part II*, ser. HCII'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 231–237.

[3] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, Jan. 2002.

[4] B. Fernandes and J. Fernández, "Bare hand interaction in tabletop augmented reality," in *SIGGRAPH '09: Posters*, ser. SIGGRAPH '09. New York, NY, USA: ACM, 2009, pp. 98:1–98:1.

[5] P. Mistry, P. Maes, and L. Chang, "Wuw - wear ur world: a wearable gestural interface," in *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, ser. CHI EA '09. New York, NY, USA: ACM, 2009, pp. 4111–4116.

[6] M. Jöst, J. Häussler, M. Merdes, and R. Malaka, "Multimodal interaction for pedestrians: an evaluation study," in *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*. New York, NY, USA: ACM, 2005, pp. 59–66.

[7] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide: a mobile context-aware tour guide," *Wirel. Netw.*, vol. 3, no. 5, pp. 421–433, Oct. 1997.

[8] R. Jacob, J. Zheng, B. Ciepluch, P. Mooney, and A. C. Winstanley, "Campus guidance system for international conferences based on openstreetmap," in *Proceedings of the 9th International Symposium on Web and Wireless Geographical Information Systems*, ser. W2GIS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 187–198.

[9] P. Song, S. Winkler, S. O. Gilani, and Z. Zhou, "Vision-based projected tabletop interface for finger interactions," in *Proceedings of the 2007 IEEE international conference on Human-computer interaction*, ser. HCI'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 49–58.

[10] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recogn.*, vol. 40, no. 3, pp. 1106–1122, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2006.06.010

[11] J. Segen and S. Kumar, "Shadow gestures: 3D hand pose estimation using a single camera," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 1, 1999, p. 485 Vol. 1.

[12] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *In ECCV*, 2006, pp. 404–417.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[14] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," vol. 5, 2004, pp. 3061–3064 Vol. 5.

[15] G. Welch and G. Bishop. (1995) An introduction to the kalman filter. Chapel Hill, NC, USA.