SUGISAKA MASANORI

Journal of Advances in Artificial Life Robotics
Vol. 3(2); September (2022), pp. 102–112
ON LINE ISSN 2435-8061; ISSN-L 2435-8061
https://alife-robotics.org/jallr.html

Research Article

Restaurant Menu with Gesture Recognition

Ian Christian Susanto, Kasthuri Subaramaniam, Abdul Samad bin Shibghatullah
Institute of Computer Science & Digital Innovation, UCSI University, UCSI Heights, 1 Jalan Puncak Menara Gading, Kuala Lumpur, WP Kuala Lumpur 56000, Malaysia

ARTICLE INFO

Article History

Received 22 December 2021
Accepted 06 November 2022

Keywords

Digital restaurant menu
Hand gesture recognition
Hand pose estimation
Human-computer interaction

ABSTRACT

The COVID-19 pandemic has brought back concerns of microorganism contamination into the public consciousness. Efforts have been made to ensure minimal chances of virus transmission by using movement control orders, encouraging social distancing, obligation to wear masks, and encouraging washing of hands so that it will reduce the chance of transmission. Indirect transmissions through surface contact are more difficult to prevent. There are many places in which people touch potentially infectious surfaces in the public, for example, ATM or touchscreen menu ordering at restaurants. Businesses and the general public are still looking for ways to minimize surface contact transmission. There have been efforts to minimize the chances of transmission through surface contact using an antiviral coating but it may be inadequate. The antiviral coating requires time to be effective. Repeatedly coating surfaces with sanitizer has high operational costs. These concerns are unlikely to disappear after the pandemic either, other viruses can be transmitted through this method. There needs to be a permanent solution. One such solution is using touchless technology such as hand gesture recognition. However, developing a hand gesture recognition algorithm can be a difficult task. The goal of the hand gesture recognition library is to facilitate the implementation of future applications. Gesture recognition is an example of a message window, icons, menus, pointer (WIMP) user interface that has poor functionality in the real world, especially when used in digital food restaurants. In this article, we will look at colorful hand gesture detection styles and finish evaluating the hand camouflage of our system. We describe a contactless digital restaurant menu system and find that the system passes stoner acceptance tests.

1. Introduction

The evolution of user interface design consists of long periods with small changes followed by short periods with large changes. It can be categorized the history of user interface design into four periods [1]. In the early 1950s to 1960s, computers had no user interfaces. Instructions had to be inputted using punch cards. The outputs were printed through line printers. This gives very limited debugging capabilities. Few people had the capabilities to perform debugging with switches and lights. Others weren't so lucky; they had to make sure their programs were error-free before inserting the punch card. Starting in the early 1960s up to the early 1980s, computers started to have alphanumeric displays. This means users can now interact by typing commands into a terminal. This type of interface still survives today through command-line shells in many operating systems such as Windows, Mac OS X, and Linux. In the 1970s, user interfaces using windows, icons, menus, and pointers became popular. This type of interface is referred to as a WIMP interface. It was made popular in 1986 with the release of Macintosh and was adapted by Windows and Motif. WIMP interfaces are still prevalent, even today, due to their ease of learning, ease of use, and ease of knowledge transfer. User interface design trends correspond of long ages with small changes followed by

Corresponding authors' E-mail: 1001954676@ucsiuniversity.edu.my, kasthurisuba@ucsiuniversity.edu.my, abdulsamad@ucsiuniversity.edu.my
URL: www.ucsiuniversity.edu.my

short ages with large changes. Van Dam [1] categorizes these trends into four ages Punch cards, command line interfaces, WIMP(window, icons, menus, pointer), and post-WIMP. Post-WIMP interfaces calculate on relations other than menus, forms, or toolbars to achieve further intuitive gests . An illustration of a post-WIMP interface is hand gesture recognition. Hand gesture recognition has many advantages, videlicet sterility as the stoner no way touches the device, availability to people with physical debits, and effectiveness of using three confines [2]. Computer vision and wearable technology are the two main types of hand gesture recognition. Although the former is easier to wear, it can stand out and requires a fresh outfit [3].

Due to the COVID-19 pandemic, people have been more cautious of their hygiene. While there exists evidence of a low risk of transmission through surface contact, other variants of SARS-CoV-2 may have higher risks of fomite transmission [4]. Therefore, caution would still be preferred. Shin and Kang [5] found reduced perceived risks when technological alternatives are provided to customer-staff interactions. The usage of touchless technology, such as hand gesture recognition, will further reduce the risks of transmission, as no surface contact is initiated.

In this paper, we propose a digital restaurant menu system using hand gesture recognition. We discuss the analysis, design, implementation, and evaluation of the application. The paper will be structured as follows; Section 1 discusses the problem statement, research objectives, and report structure; Section 2 discusses different methods of performing gesture recognition and several digital restaurant menus systems and compare different systems that use gesture recognition; Section 3 discusses the research methodology of the project, in particular the software development life cycle used and the method of gathering requirements. Results of the requirements gathering and the design of the system are elaborated in Section 4. Section 5 gives the implementation of the application, details on the hand gesture recognition module, and an evaluation of the functionalities of the application through tests. Section 6 is the conclusion of the experiments.

2. Literature Review

2.1. *Hand Gesture Recognition*

There are seven categories of vision based hand gesture according to Oudah et al [3]:

- *Color-based recognition*. Gestures are determined based on the color of each pixel. Wang and Popović [6] used the nearest-neighbor approach to track a colored glove. The drawback of this method is the need for an additional glove [7]..
- *Appearance-based recognition.* Gestures are determined by extracting features from the shape of the hand. One way to do this is through edge detection [8].
- *Motion-based recognition.* The gesture is determined by detecting changes between frames [9].
- *Skeleton-based recognition.* The image is used to extract the geometric characteristics of the hand, including joint orientation, joint spacing, skeleton joint position, and joint angle [10].
- *Depth-based recognition.* One benefit of using depth information to extract motions is the algorithm no longer needs to take lighting, shadow, and color into account [11].
- *3D model-based recognition.* The algorithm compares a 2D input to a 3-dimensional model. Ge et al [12]. produce a more detailed output by constructing a 3D model of the hand using Graph CNN.
- *Deep learning-based recognition.* Despite the labor-intensive process of gathering datasets, deep learning can be effective for gesture recognition [13].

In this project, gesture recognition is achieved with pose estimation. The advantage of this method is the ease of modifying the list of gestures to detect. Pose estimation is typically done in two steps: first, the location of the hand is determined. The initial image is cropped and resized according to the location. Next, pose estimation is performed on the image containing only the hand [14]. In a system proposed by Malik et al [15], the hand joint position, hand mesh vertices, and structural constraint are obtained by feeding a depth map into a convolutional neural network. To improve the precision of a pose estimation system and enable weakly supervised training, Spurr et al. [16] propose the introduction of biomechanical constraints. A real-time hand posture estimate solution with RGB input is presented by Zhang et al [17].

2.2. *Digital Restaurant Menus*

The most basic form of restaurant menu is paper-based. However, the consumer must wait to place their food order, which is a negative. Additionally, it wastes paper because each consumer needs an additional piece of paper to order [18]. To address these issues, some digital restaurant menu systems have been developed.

Digital restaurant menus are classified by Şahin [19] into two categories: touchscreen systems do not require staff interaction, while non-touchscreen menu systems require staff interaction. An Android-based digital ordering system was created by Bhargave et al [18]. in which each table has a tablet where customers can view the menu, allowing for self-service and a decrease in paper waste. A mobile application-based digital restaurant menu is suggested by Rusdi et al [20].

2.3. *Comparison of Hand Gesture Recognition Systems*

We compare various existing systems that use hand gesture recognition. Desai and Desai [21] proposed a home automation system operated with hand gesture recognition. Microsoft Kinect is used as input for recognition. Suriya and Vijayachamundeeswari [22] used hand gesture recognition to control the mouse on a desktop computer. Skin color detection is applied, then the convex hull algorithm is utilized to find the fingers. Using Gestix [23], a hand gesture-based interface, doctors can access electronic medical record (EMR) databases without touching a computer. Al et al [24]. created a Universal Robot (UR3) arm that recognizes hand gestures. The hand is located using proximity sensors. Zhang et al [25]. uses hand gesture recognition to control the movement of a player in a VR game. The input is gathered using the Leap Motion controller mounted in front of an Oculus Rift headset. Out of the system described above, Gestix bears the most similarities due to only using color and motion data to control a graphical user interface. The system described in [21] is analogous in that each gesture is assigned to a certain action. In contrast to [23] or [25], which interact with a robot arm or a VR player, respectively, gestures are also utilized to navigate a menu.

3. Research Methodology

3.1. *System Development Life Cycle*

The system development life cycle (SDLC) is the process by which a system is designed, built and delivered to its users so that they understand how the information system helps achieve business needs [26]. A typical SDLC can be divided into four main phases: Planning, where the project team decides how and why it will be developed; Analysis that examines currently existing systems, collects requirements and develops a system concept; Design in which details of system operation such as hardware, software, network infrastructure, user interfaces, databases, etc. are defined more precisely than in the analysis phase; The application where the system is built [27]. This includes system build, installation and support plan. The prototype model was chosen as the SDLC model for this project. This model is based on an iterative process of gathering requirements, building a model and verifying it. The process is best suited for systems where the system details are not yet clearly defined. This can happen in several ways, for example, the requirements are not clearly defined, the efficiency of the algorithm or the system architecture is unknown [28] [29].

3.2. *Questionnaire*

Requirements gathering is done using a questionnaire. Respondents consist of 29 UCSI University undergraduate and master's students. The questionnaire is shared through the UCSI Course Networking page. To understand the requirements deeper than through multiple choice, some long-form questions were included. Topics of interest include the respondent's opinions on digital restaurant menus, desirable features, concerns about the hygiene of conventional menus, and opinions on the advantages and disadvantages of using hand gesture recognition.

4. Analysis and Design

4.1. *Analysis*

We go over the results of the survey that was developed in Part 3 in this section. 2. The first query concerns the use of digital menus in eateries. Using a 5-level Likert scale, respondents selected their responses. Figure 1. shows the end outcome. 15 people selected "4" (agree), while the overall average was 4.17. We get to the conclusion that the

majority of individuals feel that restaurants should use digital menus.

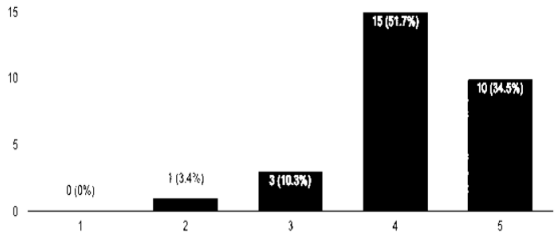

Fig. 1. Number of responses on digital menus.

The next two questions are regarding features needed in a digital restaurant menu. The first question is divided into several sub-questions, where respondents rate features on a Likert scale, ranging from 1 (very undesirable) to 5 (very desirable). In Table 1, we show the average scores of each feature. In the next question, respondents may write other features they desire that are not yet listed.

Features requested by respondents include dish pricing, special offers, special sets, digital payment methods, shopping cart systems, waiter notification systems, and ingredient freshness descriptions.

Table 1. Average score of each feature

Feature	Avg. Score	Feature	Avg. Score
Image of the dish	4.52	Non-allergy warnings	4.59
Description of the dish	4.34	Recommended	4.38
Nutritional content	3.76	List of ingredients	4.17
Dish variations	4.14	Estimated preparation time	3.93
Allergy warnings	4.59		

The respondents are then asked about their worries regarding the cleanliness of touchscreen digital restaurant menus. Concerns about the hygiene of touchscreen digital menus were raised by 17 respondents. The fact that numerous individuals use the device and could potentially leave bacteria or viruses behind is one reason for the worry. Some people expressed doubt about the restaurant's capacity to maintain the hygienic standards of their touchscreen surfaces. Those respondents who don't have any issues with cleanliness are confident in the restaurant's hygiene policies.

The next question asks respondents if they think gesture recognition can help with this issue. This is supported by 19 replies, the majority of which point to the absence of surface contact. Three respondents express no opinion. Four responses indicated skepticism toward the technology and the need for additional testing to determine its viability.

The final question concerns possible flaws with the system. Fifteen replies are concerned about the recognition system's potential for poor accuracy and sluggish performance. Seven of the replies expressed concern about the learning curve. Two comments raised the subject of maintenance. Two responses address the potential for error. Five of the respondents express no worries about any problems.

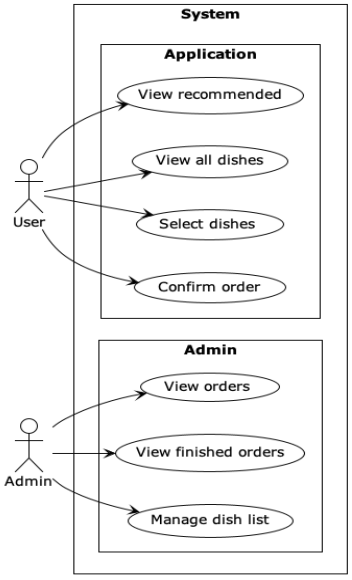

Fig. 2. Use case diagram of the system.

4.2. *Design*

In this section, we look at the system's design created to fulfill the requirements [30]. The use case diagram as shown in Figure 2 consists of two main parts: a customer-oriented department and an employee-oriented department.

Figure 3, the class diagram, shows what types of data will be needed. The gesture recognition system and the

actual digital menu do not share any data other than to control the application, therefore we can split the diagram into two. The Server part includes all classes that are used in the digital menu and the server that stores them, while the Gesture Recognition part includes all classes needed in the recognition system.

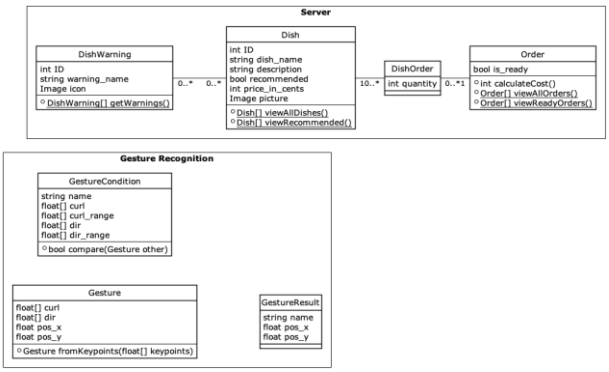

Fig. 3. The system class diagram

The activity diagram shows the user's activity as shown in Figure 4.

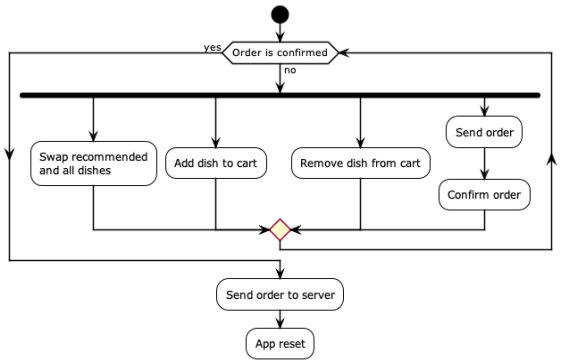

Fig. 4. Users interaction shown in the activity diagram

5. System Implementation and Evaluation

In this section, we look at the system's list of features, how gestures are recognized, and the tests that are run to make sure the system is functioning properly

5.1. *Implementation*

First, we will discuss the customer-facing application, as shown in Figure 5. Most of the screen is taken up by the menu. Other information is located on the right side of the screen. On the top right of the screen, the user is shown the

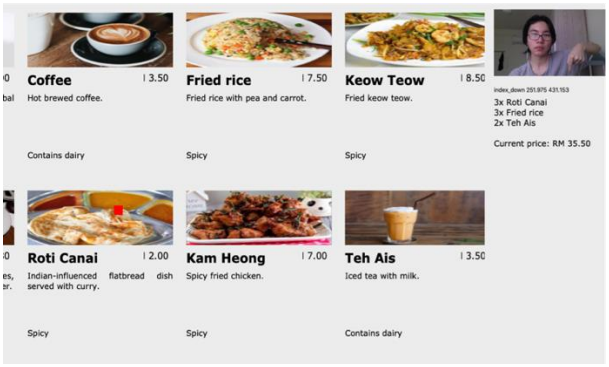

Fig. 5.A screenshot shows the cart being filled.

camera view as feedback. Below it, the detected gesture is shown. Next, the list of items in the current cart is shown, which includes which items are picked, and how many of each item are added. Finally, the current total price is shown.

By making a peace sign, the user can switch between viewing the recommended dishes and all available dishes. To prevent accidental swaps, the gesture must be detected for one second. To scroll the menu, the user can hold a "fist" gesture and move their hand to the left or right.

Adding an item to the cart requires the user to perform a "point up" gesture. To add a specific item, the user must aim the cursor at the image, name, or description of the menu. The user can redo or hold the gesture to add more than one item. To remove the item, conversely, the user can perform a "point down" gesture.

The "thumb up" gesture acts as a confirmation for various purposes. To finish editing the cart, the user can perform a "thumb up" gesture, which opens a confirmation dialog. The user can do the same gesture again to confirm their order or perform a "thumb down" gesture to cancel. If the user confirms the order, a thank you dialog box is displayed along with the total price and order number. Finally, if the previous user did not clear their cart, a new user may do the "thumbs down" gesture to reset the cart.

The staff can manage orders through web pages. As shown in Figure 6, the left part of the screen shows the current order list. Each order shows the order ID, the list of items being ordered, and the quantities of each item. If the order is done and ready to be served, the staff can click the "Ready" button on the bottom left of each order.

Fig. 6. Screenshot of the order management page.

On the right side, the ready orders can be seen. If the item is claimed by a customer, the staff can click the "Taken" button. The order is then removed from the list. In the case where the order is stated as ready by mistake, the staff can click the "Undo" button.

After each action, a confirmation message will appear at the top of the page. This will say "Operation successful!" if the action succeeds or "Operation failed!" if the action cannot be performed for one reason or another.

Fig. 7. A screen showing the list of ready orders.

Finally, the ready orders could be viewed on another page. This page is meant to be viewed by customers waiting for their orders. A screenshot is shown in Figure 7. Ready orders are gathered by polling the server.

5.2. *Hand Gesture Recognition*

Hand gesture recognition is done by first estimating the pose using Mediapipe Hands [17]. As shown in Figure 3, gestures are defined as the curl of each finger and its direction on the image plane.

The curl of a finger is the average angle between each finger bone (the vector between two key points on the finger), multiplied with a constant such that 0 is "fully straight" and 1 is "fully curled". The direction of a finger on the image plane is determined by the angle between the X axis and the vector from the base to the tip of the finger.

To determine the gesture, for each GestureCondition object, we check whether the user's pose is within the acceptable range of the condition. While this approach worked for our use case, other use cases may require more sophisticated methods, such as support vector machines.

Figure 8 shows how the application interacts with the hand gesture recognition module.

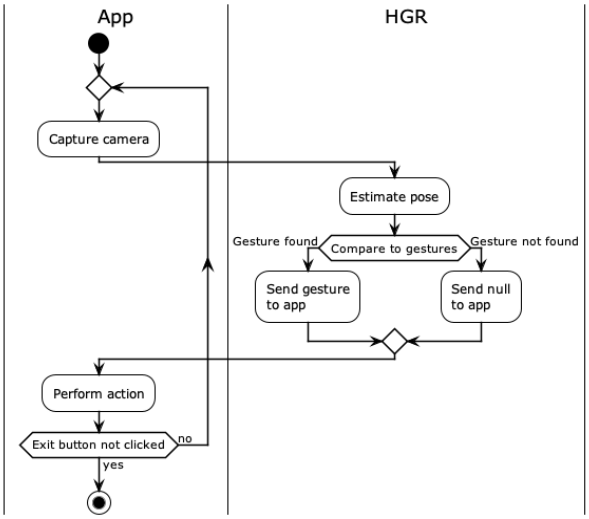

Fig. 8. Activity diagram of how the application interacts with the hand gesture recognition module.

5.3. *Evaluation*

To determine whether the system is working as expected, some evaluation should take place. This section will elaborate on the tests that were created as requirements for a satisfactory system. This includes unit tests, module tests, integration tests, and user acceptance tests.

Firstly, unit tests are performed. A unit test verifies that a single component is functioning as expected. In Table 2, Table 3, Table 4, and Table 5, we describe the unit tests performed on each module.

Table 2. Unit test for the user interface.

No.	Description	Result
1	*Item widget:* An item widget is created. The widget should have the correct layout.	Pass
2	*Item widget:* Feed dummy data into the widget. The data should be displayed correctly.	Pass
3	*Main application*: The top right part should show what the camera is seeing.	Pass
4	*Main application*: Each part should be in the correct position.	Pass
5	*Main application*: Feed dummy data into the widget. The data should be displayed correctly.	Pass

6	*Order management menu*: Feed dummy data into the page. The layout should display correctly.	Pass
7	*Order management menu*: Go to the URL that shows the confirmation alert. The confirmation message should appear for a few seconds before disappearing.	Pass
8	*Ready orders view*: Feed dummy data into the page. The layout should display correctly	Pass
9	*Ready orders view*: Modify the data on the page. The ready orders should update automatically.	Pass

Table 3. Unit test for the networking module.

No.	Description	Result
1	*REST API*: Go to the URL for the recommended menu. The server should return JSON data containing all relevant information.	Pass
2	*REST API*: Go to the URL for the entire menu. The server should return JSON data containing all relevant information.	Pass
3	*REST API*: Go to the URL for adding an order. The server should return JSON data that confirms the success along with the order ID. The new entry should exist in the database.	Pass
4	*REST API*: Go to the URL for viewing the list of warnings. The server should return JSON data containing all relevant information.	Pass
5	*Order management menu*: Insert dummy orders into the database. Go to the menu page. The page should show all orders that were inserted.	Pass
6	*Order management menu*: Click "Ready" on one of the orders. The order should now be marked as ready in the database.	Pass
7	*Order management menu*: Click "Undo" on one of the orders on the ready list. This should unmark the order as ready.	Pass
8	*Order management menu*: Click "Taken" on one of the orders on the ready list. This should remove the order from the database.	Pass
9	*Ready orders view*: Go to the order viewing URL. The page should show all ready orders.	Pass
10	*Ready orders view*: Performing tests 7 and 8, this page should change accordingly.	Pass
11	*Main application*: Given JSON data, output the data in Qt Map representation.	Pass

Table 4. Unit test for the gesture recognition module.

No.	Description	Result
1	*Module initialization*: The initialization function should initialize the Python module correctly. This is indicated when the Python module prints out a success message.	Pass
2	*Keypoint detection*: Given an image, the Python module should produce a list of 21 key points.	Pass
3	*Curl calculation*: Given a list of five key points that describe a curled finger, the curl function should output a high number (close to 1).	Pass
4	*Curl calculation*: Given a list of five key points that describe a straight finger, the curl function should output a low number (close to 0).	Pass
5	*Curl calculation*: Given a list of five key points that describe a semi-curled finger, the curl function should output a medium number (close to 0.5).	Pass
6	*Direction calculation*: Given a list of five key points that describe a finger, the direction function should output its direction correctly.	Pass
7	*Gesture comparison*: Given a GestureCondition instance and a Gesture instance that suits it, the comparison function should return true. Challenging cases such as angles that wrap from 360 degrees to 0 degrees should be tested as well.	Pass
8	*Gesture comparison*: Given a GestureCondition instance and a Gesture instance that does not suit it, the comparison function should return false.	Pass

Table 5. Unit test for the app control module.

No.	Description	Result
1	*Camera data capture*: Give the signal to perform camera capture. An image should be outputted.	Pass
2	*Camera data capture*: An image in YUV color space is given. The conversion function should output the same image in RGB.	Pass
3	*Control*: A dummy sequence of the "thumbs up" signal is given. The module should send a "confirm" signal.	Pass
4	*Control*: A sequence of "thumbs down" signals is given. The module should send a "cancel" signal.	Pass

5	*Control*: A sequence of "point up" signals is given, where the coordinate of the hand is changing. The module should send the same signal. The "point down" signal should behave similarly.	Pass
6	*Control*: A sequence of "peace" signals is given. The module should send a "swap" signal.	Pass
7	*Control*: A sequence of "fist" signals is given, where the coordinate of the hand is changing. The "move" signal should be sent.	Pass

Next, we perform module tests. These tests are meant to test each module separately before the interactions between different modules are implemented. Table 6 shows the list of module tests performed.

Table 6. Module tests along with each result.

No.	Description	Result
1	*User interface*: Start up the application. The layout of the application should be as specified in Figure 5.1.	Pass
2	*User interface*: Go to the URL for ordering data. The layout of the data should be as shown in Figure 5.9.	Pass
3	*User interface*: Go to the URL for viewing ready orders. The layout of the data should be as shown in Figure 5.11.	Pass
4	*Networking*: Using the application, send a request to the server for the recommended menus. After receiving the data, the application should output the data in Qt Map representation. Request for the entire menu and the list of warnings should behave similarly.	Pass
5	*Networking*: Using the application, send a request to the server to order an item. After receiving the data, the application should return what the order ID is.	Pass
6	*Networking*: Open the URL for the order management menu. The page should request order data from the database and show it accordingly.	Pass
7	*Networking*: Go to the URL for viewing ready orders. The page should request the list of ready orders every 10 seconds.	Pass
8	*Gesture recognition*: Initialize the module and add a list of gestures to check. Send a series of images with different gestures and one for none of the checked gestures. The module should be able to correctly determine which gestures are being performed.	Pass
9	*App control*: Start up the app control module. The module should request the camera for a new image every frame and output the RGB values of the image.	Pass
10	*App control*: Send a sequence of different gestures. The signals that are being outputted should correspond to the gestures performed.	Pass

A series of integration tests are performed. The purpose of the integration test is to check that the interaction between multiple modules works as intended. Table 7 shows the list of integration tests performed.

Table 7. The integration tests along with each result.

No.	Description	Result
1	Start up the application. It should immediately request a list of recommended items and show it as in Figure 5.2.	Pass
2	Using the app control module, send a "swap" signal. The application should request a list of the entire menu and refresh the screen accordingly.	Pass
3	Using the app control module, send a series of "move" signals. The list of items should move left and right accordingly.	Pass
4	Using the app control module, send a few instructions to "add item to cart" and "remove item from cart". The application should behave accordingly.	Pass
5	Using the app control module, scroll the entire menu and perform Test 4 again. The items being added and removed should be different.	Pass
6	Using the app control module, send two "thumbs up" signals. The interface should ask for confirmation and send a request to create a new order.	Pass
7	Add a few items to the cart, then using the app control module send a "thumbs down" signal. The cart should be reset.	Pass
8	Using the app control module, send a "thumbs up" signal and a "thumbs down" signal. The interface should ask for confirmation and then cancel it.	Pass
9	Initialize the hand gesture recognition module and the app control module. The camera should start up and the module should be correctly initialized.	Pass
10	The tester moves their hand on camera. The coordinates should be sent as a signal by the app control module.	Pass
11	The tester performs a thumbs up, a thumbs down, and a peace sign. Each gesture should be sent as a signal by the app control module.	Pass

12	The tester points up at one point, and again at another. The app control module should send a signal to “add items to cart” at those two coordinates. Similar behavior should apply to the “point down” gesture.	Pass
13	The tester moves their hand while performing a “fist” gesture. The list of items should scroll correctly.	Pass
14	Go to the order management menu. The page should load the list of orders on the left and the list of ready orders on the right.	Pass
15	Open the order viewing page. Click “Ready” on one of the orders on the order management page. This order should move to the ready list and be updated on the order viewing page.	Pass
16	Click “Undo” on one of the ready orders on the order management page. This order should go back to the unfinished orders list and be removed from the order viewing page.	Pass
17	Perform Test 15 and now click “Taken” on the ready order. This order should be removed from the ready orders list and the order viewing page.	Pass

6. Conclusion

The need for touchless technology is surfaced due to the pandemic. People are more cautious of the hygiene of surfaces. While the responsibility of maintaining hygiene has always been on the business, this type of awareness pushes businesses to be even more thorough in maintaining it. Touchless technology will be able to eliminate the need to touch surfaces, in hope of reducing maintenance costs and increasing efficiency while still being hygienic.

To reduce this worry, we propose a digital restaurant menu that uses touchless technology, in particular hand gesture recognition. We look at the hand gesture recognition techniques that currently exists. The method that is selected for this project is hand pose estimation due to being more versatile. We gather requirements by using a questionnaire. Questions that were asked include whether the respondent thinks digital menus should be used at all, what features should be present, and whether hand gesture recognition would be an improvement upon these systems. We find out that most respondents agree restaurants should be using digital menus, and that gesture recognition would improve upon it.

The design of such system follows. The system is divided into two main parts: the application and the server. The application consists of all customer-facing functionalities, such as viewing menus and ordering. The server hosts all staff-facing functionalities, such as editing the menu, managing orders, and showing ready orders.

Implementation of the system follows. The application is implemented using Qt library on C++. The gesture recognition module is written in Python, but the functions were called with C++. The server uses Django for backend and staff pages. The frontend does not use any libraries, but is implemented with HTML, CSS, and JavaScript. This system is evaluated through several unit tests, module tests, integration tests, and user acceptance tests, and was deemed satisfactory.

The customers can interact with the application without ever touching the surface. Therefore, the main purpose of the system is satisfied. However, there were some missing features that could be subject to future research.

References

1. A. Van Dam, “Post-WIMP user interfaces”, Communications of the ACM vol. 40(2), pp.63-67, 1997.
2. J.P. Wachs, M. Kölsch, H. Stern, and Y. Edan, “Vision-based hand-gesture applications”, Communications of the ACM, vol. 54(2), pp.60-71, 2011.
3. M. Oudah, A. Al-Naji, and J. Chahl, “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques”, Journal of Imaging, vol. 6(8), p.73, 2020.
4. T. Chen. "Fomites and the COVID-19 pandemic: An evidence review on its role in viral transmission." Vancouver, BC: National Collaborating Centre for Environmental Health, 2021.
5. H. Shin, J. Kang. “Reducing perceived health risk to attract hotel customers in the COVID-19 pandemic era: Focused on technology innovation for social distancing and cleanliness.” International Journal of Hospitality Management, 91, p102664, 2020.
6. R.Y. Wang, J. Popović, “Real-time hand-tracking with a color glove”, ACM Transactions on Graphics (TOG), vol. 28(3), pp.1-8, 2009.
7. M. Perimal, S.N. Basah, M.J.A. Safar, H. Yazid, “Hand-Gesture Recognition-Algorithm based on Finger Counting”, Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol.10(1-13), pp.19-24, 2018.
8. V.S. Kulkarni, S.D. Lokhande, “Appearance Based Recognition of American Sign Language Using Gesture Segmentation”, International Journal on Computer Science and Engineering, vol.2(03), pp.560-565, 2010.

9. J. Molina, J.A. Pajuelo, J.M. Martínez, “Real-time Motion-based Hand Gesture Recognition from Time-of-Flight Video”, Journal of Signal Processing Systems, vol.86(1), pp.17-25, 2017.
10. D. Konstantinidis, K. Dimitropoulos, P. Daras, “Sign language recognition based on hand and body skeletal data”, in 2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-Con), pp.1-4, 2018.
11. X. Ma, J. Peng, “Kinect Sensor-Based Long-Distance Hand Gesture Recognition and Fingertip Detection with Depth Information”, Journal of Sensors, 2018.
12. L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, J. Yuan, “3D Hand Shape and Pose Estimation from a Single RGB Image”, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.10833-10842, 2019.
13. X.Y. Wu, “A hand gesture recognition algorithm based on DC-CNN”, Multimedia Tools and Applications, pp.9193-9205, 2019.
14. C. Zimmermann, T. Broz, “Learning to Estimate 3D Hand Pose from Single RGB Images”, in Proceedings of the IEEE International Conference on Computer Vision, pp.4903-4911, 2017.
15. J. Malik, A. Elhayek, F. Nunnari, D. Stricker, “Simple and effective deep hand shape and pose regression from a single depth image”, Computers & Graphics, vol.85, pp.85-91, 2019.
16. A. Spurr, U. Iqbal, P. Molchanov, O. Hilliges, J. Kautz, “Weakly Supervised 3D Hand Pose Estimation via Biomechanical Constraints”, in Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVII 16, pp.211-228, 2020.
17. F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.L. Chang, M. Grundmann, “Mediapipe Hands: On-device Real-time Hand Tracking”, 2020.
18. A. Bhargave, N. Jadhav, A. Joshi, P. Oke, S.R. Lahane, “Digital Ordering System for Restaurant Using Android”, International Journal of Scientific and Research Publications, vol.3(4), 2013.
19. E. Şahin, “An Evaluation of Digital Menu Types and Their Advantages”, Journal of Tourism and Gastronomy Studies, vol.8(4), pp.2374-2386, 2020.
20. J.F. Rusdi, N.A. Abu, N. Agustina, M. kchouri, S. Dewi, “Software Development Stages of Mobile Computing Implementation in Restaurant Food Ordering”, SciTech Framework, vol.1(1), pp.24-33, 2019.
21. S. Desai, A. Desai, “Human Computer Interaction Through Hand Gestures for Home Automation using Microsoft Kinect”, in Proceedings of International Conference on Communication and Networks, pp.19-29, 2017.
22. R. Suriya, V. Vijayachamundeeswari, “A survey on hand gesture recognition for simple mouse control”, in International Conference in Information Communication and Embedded Systems (ICICES2014), pp.1-5, 2014.
23. J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, J. Handler, “Gestix: A Doctor-Computer Sterile Gesture Interface for Dynamic Environments”, in Soft Computing in Industrial Applications, vol.39, pp.30-39, 2008.
24. G.A. Al, P. Estrela, U. Martinez-Hernandez, “Towards an intuitive human-robot interaction based on hand gesture recognition and proximity sensors”, in 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp.330-335, 2020.
25. F. Zhang, S. Chu, R. Pan, N. Ji, L. Xi, “Double hand-gesture interaction for walk-through in VR environment”, in 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), pp.539-544, 2017.
26. N.Pepin, A.S. Shibghatullah, K.Subaramaniam, R.A. Sulaiman, Z.A. Abas, S.Sarsam, "A Reusable Product Line Asset in Smart Mobile Application: A Systematic Literature Review", (2022) International Journal of Advanced Computer Science and Applications, Vol. 13, No. 9, pp. 53-60, 2022.
27. A.A.N.Akhla, T.C. Ling, A.S.Shibghatullah, C.S. Mon, A.K. Cherukuri, C.L. Yen, L.C. Yi, "Impact of Real-Time Information for Travellers: A Systematic Review", Journal of Information and Knowledge Management,2022.
28. A.S.Shibghatullah, A. Jalil, M.H.A. Wahab, J.N.P. Soon, K. Subaramaniam, T. Eldabi, "Vehicle Tracking Application Based on Real Time Traffic", International Journal of Electrical and Electronic Engineering and Telecommunications, Vol. 11, No. 1, pp. 67-73, 2022.
29. M.S.B. Haini, C.S.Mon, A.S.B. Shibghatullah,A.B.,Jalil, K.A.P.Subaramaniam, A.A.A.Hussin, "An investigation into requirement of mobile app for apartment residents", International Journal on Advanced Science, Engineering and Information Technology, Vol. 9, No.6, pp. 1841-1848, 2019.
30. A.F.N.A. Rahman, A.S. Shibghatullah, Z.A. Abas, T. Eldabi, C.S. Mon, A.A.A.Hussin, "Solving late for relief event in bus crew rescheduling using multi agent system", Journal of Engineering Science and Technology, Vol. 15, No. 3, pp. 1972-1983, 2020.

Authors Introduction

Ian Christian Susanto

Ian is currently an undergraduate student pursuing his BSc (Hons) in Computing studies at the Institute of Computer Science and Digital Innovation (ICSDI) at UCSI University, Malaysia.

Asst. Prof. Ts. Kasthuri Subaramaniam

Dr Kasthuri is currently an assistant professor at the Institute of Computing and Digital Innovation (ICSDI), UCSI University, Kuala Lumpur, Malaysia. She holds a Bachelor's degree in Computer Science and a Master's degree in Computer Science from the University of Malaya. She has supervised many college students as principal supervisor and co-supervisor. She has publications in Scopus-Indexed and Web of Science. She is also a co-investigator of the Pioneer Science Incentive Fund (PSIF) in the field of augmented reality. His research interests include human-computer interaction, human personality types, augmented reality, e-learning, mobile commerce, and e-commerce.

Assoc. Prof. Dr. Abdul Samad bin Shibghatullah

He obtained a Bachelor of Accounting from Kebangsaan University Malaysia, Bangi, Malaysia in 1999, a Master of Computer Science from Teknologi University Malaysia, Skudai, Malaysia in 2002 and a Ph. degree in Computer Science from Brunel University in Uxbridge, UK. He is currently Associate Professor at the Institute of Computing and Digital Innovation at UCSI University in Kuala Lumpur, Malaysia. His current research interests include optimization, modeling, and scheduling.