Interactive Gestures for Liver Angiography Operation

Dina A. Elmanakhly[2](✉), Ayman Atia[1], Essam A. Rashed[2], and Mostafa-Samy M. Mostafa[1]

[1] HCI-LAB, Department of CS, Faculty of Computers and Information,
Helwan University, Helwan, Egypt

[2] Image Science-LAB, Department of Mathematics, Faculty of Science,
Suez Canal University, Ismailia, Egypt

dinaelmanakhly@yahoo.com

Abstract. The main challenge of creating large interactive displays in the operating rooms (ORs) is in the definition of ways that are efficient and easy to learn for the physician. Apart from traditional input methods such as mouse and keyboard, we have developed a multimodal system with two different vision based human-computer interaction (HCI) systems that can simplify the way surgeons interact with the medical images shown on the LCD display. The purpose of this work is to construct a gesture recognition system with a fast, accurate, and easily attainable method. The first system is a laser pointer interaction framework that supports a 2D stroke gesture interface. The recorded laser gestures are recognized using two different algorithms: dynamic time warping (DTW) and one dollar (1$) recognizer. Our experimental results showed that the DTW algorithm performs better with an overall accuracy of 90 %. The second prototype presents an intuitive HCI to manipulate images using freehand gestures. In order to strengthen the gesture recognition process, the system incorporates contextual information to determine the intent of the user of interacting with the large display. Two cameras are used to observe the surgeon's hand movements to continuously determine and monitor what the surgeon intends to perform. Experimental results showed that the system accuracy is 95 % for recognition with the effect of contextual integration.

Keywords: Gesture recognition · Laser pointers · Hand gestures

1 Introduction

Recently, the touchless technique has received considerable attention as one of the promising methods due to its ability to provide a natural interaction between human and computer. Interaction is said to be touchless if there is no contact between the user and any part of the system. For example, the interaction with a Nintendo Wii through its wireless controller is not touchless. Voice based technique, eye gaze, and hand

V.G. Duffy (Ed.): DHM 2016, LNCS 9745, pp. 409–421, 2016.
DOI: 10.1007/978-3-319-40247-5_41

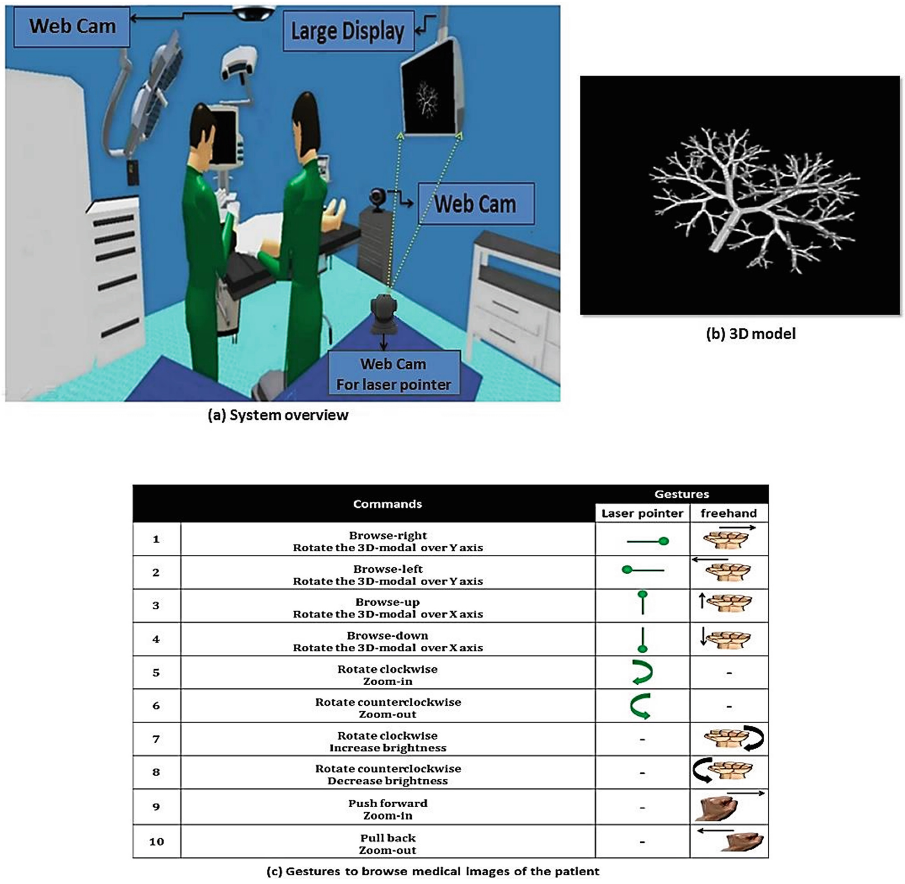
gestures are just a few examples of touchless interactions [1, 2]. There have been several works on touchless interaction that cover different fields, for instance, the medical applications [3].

In the operating room (OR), doctors should minimize the action of touching the surrounding area because of sterilized operation theater. In such situation, the surgeon usually asks another staff member to help him to interact with the display (e.g. zoom in/out images). This way of interaction might lead to some latency and inaccurate decisions between doctor's decisions and actions to be performed by the other member. Thus, several interfaces aiding direct control for the physicians have been developed. Some of these systems depend on voice control devices. However, their use may be associated with obstacles related to misinterpretation of commands, and if there is noise in the room, the error will increase [4]. In other systems, freehand gestures are utilized. The Opect system introduced intangible interface depends on the Kinect device [5]. Although the Opect has a high degree of accuracy, our system is characterized by a low price compared to the Kinect device.

Hepatic angiography is a study of an X-ray of the blood vessels that supply the liver. The procedure uses a thin and flexible catheter that is placed into a blood vessel through a small cut. A trained doctor called an Interventional radiologist usually performs the procedure. In this paper, we present two approaches that will help surgeons in the operating room to interactively control the 3D image on the large screen using pre-set gestures, which can be executed at a distance from the display. One method uses an inexpensive laser pointer with an on/off button and the other one uses freehand gestures.

2 System Overview

The Proposed system aims to help interventional radiologists in the liver angiography operation to control the 3D image (Fig. 1(b)) on the large screen using pre-set gestures (Fig. 1(c)). The system is composed of three cameras and a large screen (Fig. 1(a)). The first method uses one web camera to capture the movement of the laser pointer. The other system uses two web cameras to recognize freehand gestures. The first camera is placed in front of the doctor and the second one is held on a ceiling position.



(a) System overview

(b) 3D model

Commands		Gestures	
		Laser pointer	freehand
1	Browse-right Rotate the 3D-modal over Y axis	—●	→
2	Browse-left Rotate the 3D-modal over Y axis	●—	←
3	Browse-up Rotate the 3D-modal over X axis	●	↑
4	Browse-down Rotate the 3D-modal over X axis	●	↓
5	Rotate clockwise Zoom-in	↻	-
6	Rotate counterclockwise Zoom-out	↺	-
7	Rotate clockwise Increase brightness	-	↻
8	Rotate counterclockwise Decrease brightness	-	↺
9	Push forward Zoom-in	-	→
10	Pull back Zoom-out	-	←

(c) Gestures to browse medical images of the patient

Fig. 1. Proposed system

3 System Details

3.1 Laser Pointer

After the image has been captured by the web camera, it is considered as the input of the laser detection algorithm. Laser spot detection step is performed to identify which one of the detected foregrounds is the actual laser spot. Background subtraction is the most widespread method for detecting the laser spot. In a laser pointer system, most cameras are fixed in their position, thus allowing the use of background subtraction operation to get the foreground object. Frame differencing is the simplest background subtraction technique where a current frame is compared to the previous one in which a significant difference between those frames is identified as the foreground object.

First, the web camera captures the first and the second frames and convert them to grayscale for the process of frame differencing (Fig. 2(b)). Thus, performing frame differencing with the sequential images will output an “intensity image” that needs to be converted into a binary image that provides better information (Fig. 2(c)). However, the thresholded function still needs more enhancement to detect the laser spot. The “Blur” function is performed in order to get rid of the noise and improve the appearance of the laser spot (Fig. 2(d)). Unfortunately, this function will output an intensity image again, so the threshold binary operation is performed for a second time. Fig. 2(e) shows the final threshold image after it’s been “blurred”.

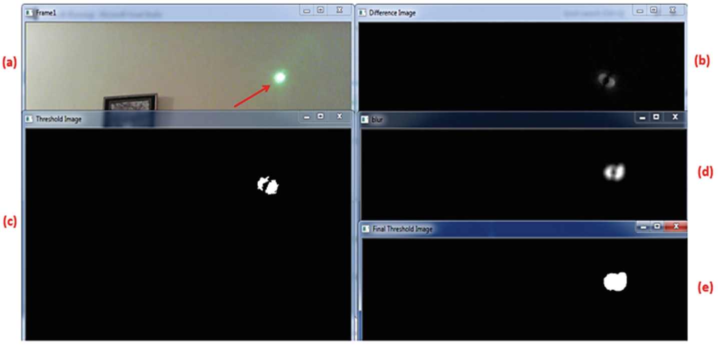


Fig. 2. Background subtraction: (a) Original image, (b) Intensity image, (c) Threshold-binary image, (d) Blur images, and (e) Final thresholded image.

Find contours approach is performed for locating a moving object (i.e. Laser spot). The final threshold binary image is treated as the input image for the tracking process (Fig. 2(e)). First, “findContours” Process is used to find all contours in the binary image where each contour is stored as a point vector (i.e. we search only for the extreme outer contours). Then, we make the assumption that the biggest detected contour is the object we are looking for. The circumscribed rectangle or bounding box has been drawn around the largest contour. Then, the centroid of the rectangle will be the object’s final estimated position.

Finally, to recognize the laser gestures, two different algorithms are tested in order to choose the best algorithm for the proposed system. The first algorithm is called dynamic time warping (DTW) algorithm [6]. The second one is called 1$ recognizer algorithm [7].

3.2 FreeHand Gestures

Hand Segmentation. We propose a hand detector that splits into two stages: find the position of the hand only in the first frame, then extract the hue component of the glove color. As, the hue value will be used for further processing.

First, the images are captured with the front camera. The hand region is extracted from the background with the background subtraction method. The hand detection output image is a binary image as the white pixels represent the hand region and the black pixels belong to the background (Fig. 3(a)). Then, “findContours” function is used to find all contours in the binary image. A bounding box has been drawn around the largest contour. Then, the centroid of the rectangle will be the hand’s estimated position. However, the subtracted image contains the whole arm region causing a false hand detection as seen in Fig. 3(b).

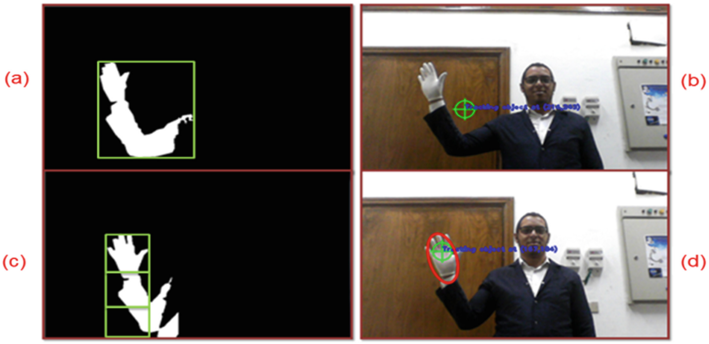


Fig. 3. Hand detection process: (a) and (c) background subtracted images, (b) false detection, (d) true detection.

Therefore, the bounding box width has been reduced to 50 % and the height is divided into three equal parts. The upper part is assumed to be the rectangle that specifies the palm and finger regions (Fig. 3(c)). Then, we calculate again the center of the new rectangle to be the final position of the hand (Fig. 3(d)). Finally, the hue color of the segmented rectangle is calculated in order to be used in the tracking algorithm that depends only on the color that is extracted from the detection stage.

Hand Tracking. The surgeon’s right hand is tracked by using a CamShift algorithm [8]. The algorithm works by tracking the hue color value of an object where the color probability distribution for a 2D search window is calculated.

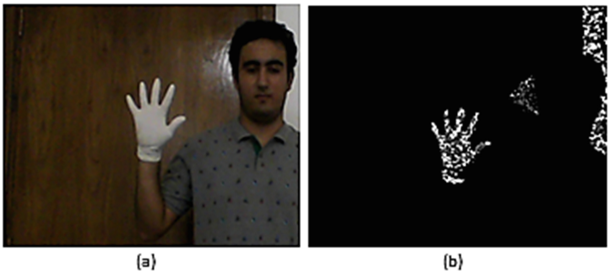


Fig. 4. (a) Test image, (b) Back-projection image

First, we create a hue histogram of an image containing the object of interest (i.e. Surgeon's hand). The hand should fill the image as far as possible for better results. Second, the histogram back-projected image is created. Backprojection image is used to find the object of interest in an image. In other words, it creates an image of the same size as that of our input image (i.e. single channel), as each pixel corresponds to the probability of that pixel belonging to the hand. Therefore, the output image will have our object of interest in more white compared to the remaining part (Fig. 4). Finally, the backprojection image will be the input image to the CamShift algorithm that finds the location with the maximum density. Obviously, when the object moves, the movement is reflected in histogram back-projection image.

Spotting Problem. One of the challenges in gesture interaction in the OR is the spotting problem, the problem of discrimination between intentional and unintentional gestures. Surgeon's hand movements can be divided into two categories: gestures that are performed in the operation mode and the others are performed in the interaction mode. The interaction mode is detected without asking the surgeon to perform a start gesture, we detect it when the surgeon intends to manipulate the images on the large screen by raising his hand up from the patient at a level slightly above the chest that exceeds a threshold value. In contrast, when the surgeon returns his hand down at the patient's level or any level below the level of the chest that means he enters the operation mode and the system is no longer able to recognize any gestures until the surgeon switches to the interaction mode again.

The surgeon's hand movements are divided into periods identical in length, each of them is composed of 20 frames. In practice, we found number of frames = 20 to be adequate. Each period is divided into two segments to merge the second one with the next period in order to make up a gesture. Then, generate a vector representation for this gesture that is classified later through the matching procedure using DTW against a set of templates. The classification results provide information that will enable the system to detect the surgeon's mode. Each two adjunct vectors have a segment length overlap, as shown in Fig. 5.

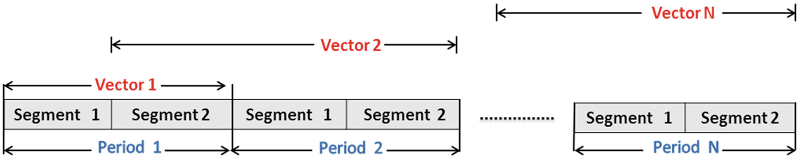


Fig. 5. Illustrations of frame segmentation

In the interaction mode, the surgeon will interact with the system by gesturing in the view of the front camera. However, the start and the end point of the intended hand gesture in a continuous hand trajectory should be identified. Each gesture is recognized by the hand-close and hand-open events that represent the two possible states for the hand. The hand state (open and close) detection is performed by measuring the height of the track window (i.e. the ellipse shape) that is drawn around the user's hand, the size of the track window is updated in each frame (Fig. 6).

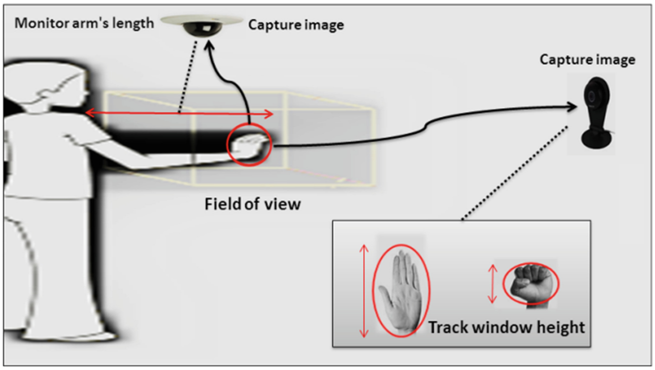


Fig. 6. Image capture with two cameras

Hand Gesture Recognition. In the hand gesture recognition process, a problem of losing information needs to be solved. To overcome this problem, we use two cameras. The trajectory of each gesture is recorded twice, one time from the view of the front web camera and the other time from the view of the ceiling camera in order to make the recognition process easier. The data that is collected from the front camera is used to recognize gestures such as the circle in the two directions and the line with the four directions: up, down, left, and right. The collected data of the ceiling camera recognizes only gestures that are unclear for the front camera (i.e. push-forward and pull-back gestures). It is an important issue to identify when the system starts using the recording data from the ceiling camera. The arm's length is measured in each frame in order to know whether or not the hand is stretched out (Fig. 6). In recognition process, we depend on the 1$ algorithm. This algorithm performs well and achieves good results in differentiating the shape of the gestures (i.e. mouse gestures) at overall accuracy more than 99 % using only a small number of training samples. Take into consideration that we may attempt to increase the number of the command gestures in our system, 1$ classifier is well suited for our situation. In the following section, we conduct a small study to measure the performance of the 1$ algorithm in classifying the shape and the direction of the hand gestures.

4 Primitive Experiment

We have conducted a primitive study for calculating the accuracy of 1$ classifier, we found that the algorithm performs well and achieves good results in differentiating the shape of the gestures, but it also has some limitations. The 1$ can not distinguish gestures whose identities depend on aspect ratios, locations, or specific orientations. This means that separating up-lines from down-lines or left-lines from right-lines is not possible without modifying the conventional algorithm. So, we conducted another study to calculate the accuracy of DTW in recognizing only the directions of the line gestures. We have found that the DTW can achieve 95 % accuracy for the line's direction, Table 1 shows the results for both algorithms. After these two primitive studies, the recognition algorithm in the proposed system depends mainly on two algorithms: the 1$ for recognizing the shape of the gesture as it gives good results with only a few loaded templates and the DTW to recognize the direction of the line gestures.

Table 1. Detection of the line gesture direction

Gesture	1$ detection (%)	DTW detection (%)
Left-line	84	100
Right-line	72	88
Up-line	68	92
Down-line	80	100
Push forward	76	96
Pull back	72	92

5 Experiments and Results

5.1 Laser Pointer

The experiment was performed by 15 subjects, there were 5 males and 10 females aged between 19 and 25. The users were asked to stand close to the large display at a distance of 1.5 m (Fig. 7). Each user is asked to use three input ways: the traditional way, i.e. input with a mouse and keyboard, and the laser pointer per each classifier (DTW and 1$ recognizer) in order to complete tasks as shown in Fig. 8.

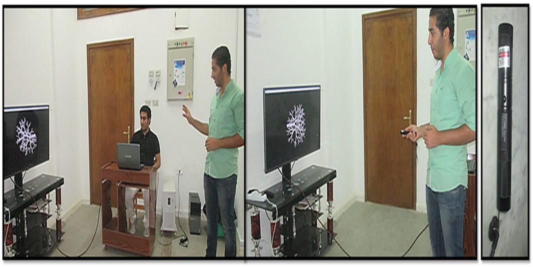


Fig. 7. Laboratory experiment

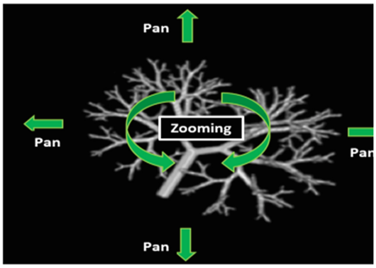
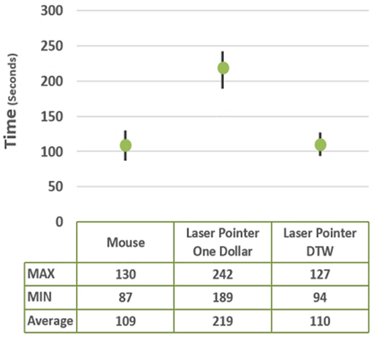


Fig. 8. Laser command mappings

Results and Discussion. Experimental data shows that mouse-based interactions are faster than the laser pointer either with the DTW or the 1$ (Fig. 9). However, the average time cost to complete the total task with the mouse and the keyboard is just one second faster than that with a laser pointer using the DTW recognizer. Considering that, the latency problem or the slowness of the spot recognition can be solved by using better and faster cameras connections, we are satisfied with the performance of the laser pointer with DTW recognizer relative to the traditional method.

The averaging accuracy of DTW is 89.6 %, it means that the system misinterprets the laser pointer gestures at a rate of 10 %. As shown in Figs. 9 and 10, the laser pointer with the 1$ recognizer including error correction is almost twice as slow as the mouse and the averaging accuracy of 1$ is 75 %. The 1$ can not distinguish up-lines from down-lines and left-lines from right-lines. This explains why the 1$ comes last in the total time results. When the users make mistakes, they need extra time to correct their errors as we give them two trial for error correction.

Figure 11 displays the number of errors for all participants in each session for both algorithms. It is observed that the errors in DTW in session 3 and 4 are much less than those in session 1 and 2 which indicates that a user can be expected to improve in the usage of the laser pointer with the DTW. However, we cannot expect that in the 1$ classifier. There was no discernible difference in user performance between any of the four sessions.



	Mouse	Laser Pointer One Dollar	Laser Pointer DTW
MAX	130	242	127
MIN	87	189	94
Average	109	219	110

Fig. 9. Time cost in seconds

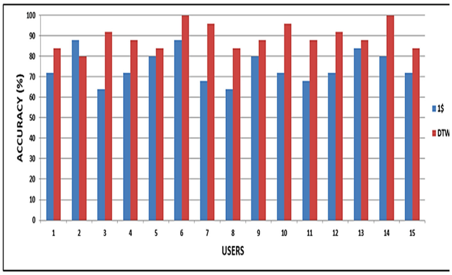


Fig. 10. Accuracy of laser pointer with two algorithms

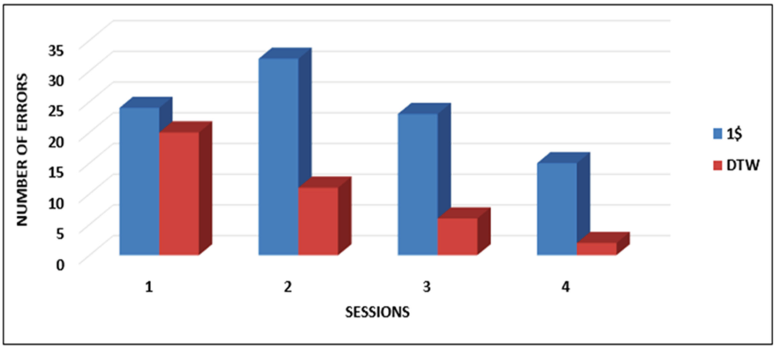


Fig. 11. Number of incorrect gestures recognitions for all participants in each session

The obtained data from participants gestures are then analyzed using an ANOVA test. First, a one-way ANOVA test is performed in regards to the time of the total task for the three input ways. The test indicates that there is a significant difference between the three algorithms ($p<0.05$). However, the ANOVA analysis only indicates that if there is a significant difference between at least one pair of the group means. It does not indicate what the pair or pairs are significantly different. To find which method is of better performance, a Tukey HSD test needs to be performed. A Tukey test is interested in examining mean differences where any two means that are greater than HSD are significantly different. From ANOVA results, the mean values of the three input ways: mouse, 1$, and DTW are M1, M2, and M3, respectively. A post-hoc analysis using a Tukeys HSD test showed that HSD = 13.18, with $|M1 - M2| = 109.36$, $|M2 – M3| = 108.43$, and $|M3 - M1| = 0.93$. Thus, the total time of the 1$ recognizer was significantly larger than those with the other input methods (i.e. no difference was found between the traditional method using the mouse and the laser pointer using DTW recognizer). Second, another one-way ANOVA test is performed in regards to recognition rate (accurracy) of the laser pointer with the two algorithms: DTW and 1$. The test indicates that there is a significant difference between the two algorithms ($p<0.05$). Thus, participants within the DTW method group generated significantly more accurate gestures than the 1$ recognizer.

5.2 Freehand Gestures

Our goal of this experiment was to measure the performance of the proposed hand gesture recognition algorithm, the classification accuracy is evaluated in the experiment using three algorithms: DTW recognizer, 1$ recognizer, and the proposed algorithm. We asked each user to perform a set of movements in the first two minutes (i.e. operation mode). As seen in Fig. 12(a), the user performs hand movements. For example, deals with the patient, talks with a staff member or points to any object. Then for 30 s, the user enters the interaction mode by raising his hand above the chest level next to his/her face (Fig. 12(b) and (c)).

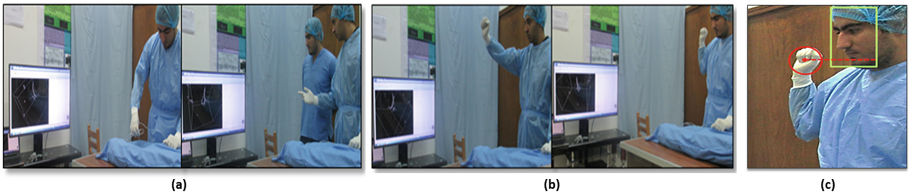


Fig. 12. Laboratory experiment

Results and Discussion. Figure 13 shows the performance of each user in the three algorithms. The averaging accuracy of the proposed algorithm, DTW, and 1$ is 95 %, 84 %, and 80 %, respectively. In the confusion matrix (Fig. 14), the entries of the matrix record the numbers of the gesture predicted as the corresponding gestures.

For example, the numbers in the first row 95 %, 2 %, and 3 % are in the columns corresponding to the gestures circle-clockwise, up-line, and down-line, respectively. It means that the proposed system misinterprets the circle-clockwise gesture at a rate of 5 %. Generally, there are some obvious sources of error in our experiment. For instance, some circle gestures and line gestures are misclassified as down-line, i.e. right-line and left-line. The reason is some users perform lines with a slightly skewness to the down and circles with a very small diameter. So, in these cases the system misunderstands these gestures as down-line. In Push-Forward gesture, the user should perform this gesture with a stretched-out arm, otherwise the system will misunderstand it as a right or left line.

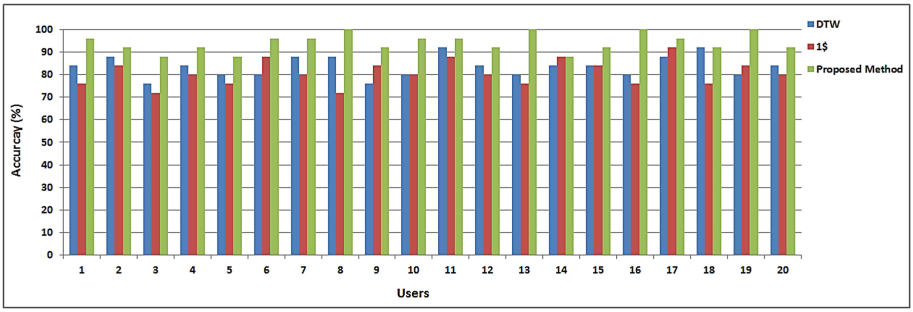
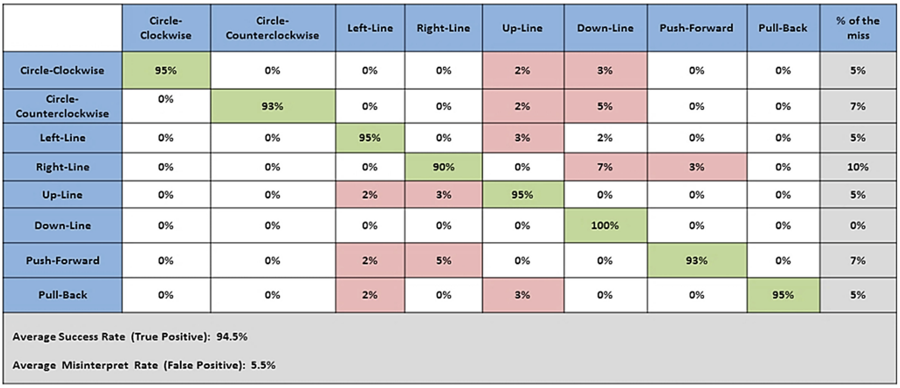


Fig. 13. Accuracy of the three algorithms achieved from the second experiment



	Circle-Clockwise	Circle-Counterclockwise	Left-Line	Right-Line	Up-Line	Down-Line	Push-Forward	Pull-Back	% of the miss
Circle-Clockwise	95%	0%	0%	0%	2%	3%	0%	0%	5%
Circle-Counterclockwise	0%	93%	0%	0%	2%	5%	0%	0%	7%
Left-Line	0%	0%	95%	0%	3%	2%	0%	0%	5%
Right-Line	0%	0%	0%	90%	0%	7%	3%	0%	10%
Up-Line	0%	0%	2%	3%	95%	0%	0%	0%	5%
Down-Line	0%	0%	0%	0%	0%	100%	0%	0%	0%
Push-Forward	0%	0%	2%	5%	0%	0%	93%	0%	7%
Pull-Back	0%	0%	2%	0%	3%	0%	0%	95%	5%

Average Success Rate (True Positive): 94.5%

Average Misinterpret Rate (False Positive): 5.5%

Fig. 14. Confusion matrix for the proposed algorithm

Figure 15 displays the number of errors for all participants in each session for each algorithm. In DTW, there is a slight improvement from session 1 to session 3. However, there is no discernible difference in user performance between any of the three 1$ sessions. Furthermore, the number of errors in session 1 and 3 is almost equal. It is

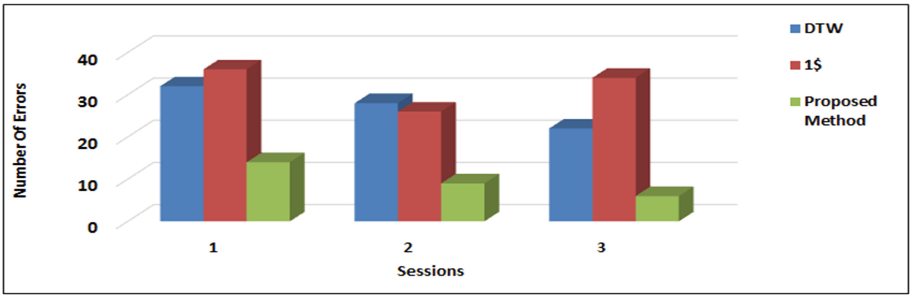


Fig. 15. Number of errors for all participants in each session

observed that the errors in the proposed algorithm in session 3 are much less than those in session 1 which indicates that the performance of the user can be expected to improve with the proposed algorithm.

From ANOVA results, the mean values of the three algorithms DTW, 1$, and the proposed one are M1 = 83, M2 = 80.8, M3 = 94.2, respectively. A post-hoc analysis using a Tukeys HSD test showed that HSD = 3.56, with |M1 – M2| = 2.2, |M2 - M3| = 13.4, and |M3 -M1| = 11.2. Thus, participants within the proposed method group generated significantly more accurate gestures than the other two groups and there were no significant differences between the results of DTW and 1$.

6 Conclusion

A multimodal system with two input modality is introduced in this paper. It provides a good solution to help interventional radiologists in the liver angiography operation to control the 3D image on the large screen using pre-set gestures. Our Future work includes firstly testing the proposed multimodal system in a real-world scenario such as the OR. However, the OR is a very critical place so we need to increase the accuracy of the proposed algorithm. Secondly, extend this work to support multiple users with personalized service.

References

1. Jafari, R., Ziou, D.: Eye-gaze estimation under various head positions and iris states. Expert Syst. Appl. **42**(1), 510–518 (2015)
2. Munteanu, C., Jones, M., Whittaker, S., Oviatt, S., Aylett, M., Penn, G., Brewster, S., Alessandro, N.: Designing speech and language interactions. In: CHI 2014 Extended Abstracts on Human Factors in Computing Systems, Toronto, Canada (2014)
3. Gallo, L., Placitelli, A. P., Ciampi, M.: Controller-free exploration of medical image data: experiencing the kinect. In: Computer-Based Medical Systems (CBMS), Bristol, United Kingdom (2011)

4. Alapetite, A., Andersen, H.B., Hertzum, M.: Acceptance of speech recognition by physicians: a survey of expectations, experiences, and social influence. Inter. J. Hum. Comput. Stud. **67** (1), 36–49 (2009)
5. Yoshimitsu, K., Muragaki, Y., Maruyama, T., Yamato, M., Iseki, H.: Development and initial clinical testing of opect: an innovative device for fully intangible control of the intraoperative image-displaying monitor by the surgeon. Oper. Neurosurg. **10**(1), 46–50 (2014)
6. Myers, C., Rabiner, L.: A level building dynamic time warping algorithm for connected word recognition. IEEE Trans. Acoust. Speech Sig. Proc. **29**(2), 284–297 (1981)
7. Wobbrock, J. O., Wilson, A. D., Li, Y.: Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In: Proceedings of the 20th Annual Symposium on User Interface Software and Technology, New York, USA (2007)
8. Bradski, G.R.: Real time face and object tracking as a component of a perceptual user interface. In: Proceedings of IEEE Workshop on Applications of Computer Vision, Princeton, NJ (1998)