REAL-TIME VISION-BASED TOUCHLESS INTERACTION IN COMPUTER GENERATED HEART MODEL USING A PRE-DEFINE OFFSET SCREEN DISTANCE (POSD) METHOD

Ing Kye NG[#1], Abdullah BADE[*2], Asdalifah TALIBE[*3]

[#]*Faculty of Science and Environment, University Malaysia Sabah*

[1]klyexl@yahoo.com

[2]abade08@yahoo.com

[3]asdalifah@gmail.com

Abstract—**This paper aims to develop a real-time vision-based touchless interaction system for computer generated heart visualization using an integration of motion gesture approaches. The proposed method is to overcome the limitation of physical interaction such as hygienic problems. This project includes gesture algorithms such as Gesture Estimation, Feedback and Gesture Recognition. A Predefine Offset Screen Distance for Gesture Recognition is proposed. The flow of the algorithm can be divided into four steps. The first step is the calculation of the offset line. The second step is the declaration of Recognition Threshold to reduce the noise of the motion. The third step is the range in which it is defined as left or right to be used for gesture recognition. The fourth step is the execution of the basic transformations such as rotation, translation and scale if a data motion lies within the range. Four experiments were carried out to evaluate the prototype system. The experiments are Gesture Test, Lighting Test, Distance Test and Memory test. Besides, the proposed Predefine Offset Screen Distance is further analysed by measuring the time complexity using Big-O Notation. The results showed that the proposed Predefine Offset Screen Distance is not very efficient and it has higher computational cost in time complexity with O(n^2). In this study, a prototype of Augmented Reality system that implements gesture motion has been developed.**

***Keywords*— Motion Gesture, Augmented Reality, medical imaging, solid modelling, Predefined Offset Screen Distance.**

I. INTRODUCTION

There are many diseases contribute to high mortality rate in human such as Cancer. Surgery is one of the oldest and most requisite treatments to treat cancer. The rapid advancement of technology has enhanced the surgery process by preventing complication and minimising the operation time. Besides, a surgeon can provide remote surgery (telesurgery) in a long distance [1]. However, the physical interaction between the surgeon and machine will lead to hygienic problems. The sterility restriction makes touchless interaction an interesting and efficient solution for surgeons to interact with medical system directly.

This project begins with the study of touchless interaction based on gesture technology in medical imaging during the performance of image-based surgery. Furthermore, this research focuses on 2D motion gesture. In Image-Guided Surgery (IGS) practice, surgeon will rely on the computer-generated image that superimposes into the real world as a guide to conduct the operation. Hence, Augmented Reality (AR) will be used as our platform for this research as the user is partially immersed into the AR world. To allow real and virtual object to be aligned properly with each other, AR must be studied carefully by considering these three AR system characteristics:

i. Procedure to combine real and virtual
ii. Interactivity
iii. Procedure to register the 3D model[2]

Touchless interaction algorithm which is known as the motion gesture focuses on three main steps: motion estimation, user feedback and gesture recognition in stages. Generally, the purpose of this project is to develop a real-time system which is implemented with Predefine Offset Screen Distance of motion recognition using vision-based for medical purpose.

II. PROBLEM STATEMENT

A surgeon needs to interact frequently with the medical system before or during surgery in order to review medical image and records. The physical contact between the computer and their peripheral devices such as keyboard, mouse and touch screen is difficult to be sterilised. This is because the devices have higher chance to transfer contaminated material such as bacteria. Thus, direct and touchless interactions are being proposed because they are more intuitive and also provide natural interactivity compared to contact-based input devices.

III. RELATED WORK

There are two approaches in touchless interaction implementation. The two approaches are gesture and vocal commands system. However, vocal commands are sensitive to the environment which is not suitable to be implemented for the system. The reason is that there are many people inside the surgical room such as a surgeon and his assistants during surgery [3].

The motion gesture interaction has been focused on the mobile devices as the devices contain motion sensor. Jaime Ruiz and Yang Li proposed a doubleflip motion gesture for mobile motion-based interaction [4]. The result shows that the proposed method is more resistant to false positive conditions and high recognition rate. Another research team presented a gesture design tool named MAGIC [5]. MAGIC is a motion gesture control in game domain. The presented prototype system was then tested by numbers of participants. The users provided positive feedback.

IV. METHODOLOGY

The architecture of the system design for developing a vision-based real-time touchless interaction application can be divided into five stages: display system, input/output devices, rendering of 3D heart model data, device's API and gesture interaction procedures. Gesture interaction procedures are the main loop of the system. The architecture begins with the input/output stage and ended with display.

Camera is the most important input device to be used in this system. The overall process of motion gesture interaction requires a camera to capture the frames from a time series. Besides, devices such as mouse and marker tool are used for the purpose of interaction between the user and the system. Motion gesture interaction then transits to motion gestures engine which is the most important stage in the system. The gesture engine ensures that the performance of gesture sign by the user could trigger the transformation of the virtual heart such as translation, rotation and scaling.

There is a pre-data storage where the heart model data is stored into the system before the run time. Rendering of 3D heart model begins with the modelling of 3D virtual heart. The data is then stored into the system to visualise the model.

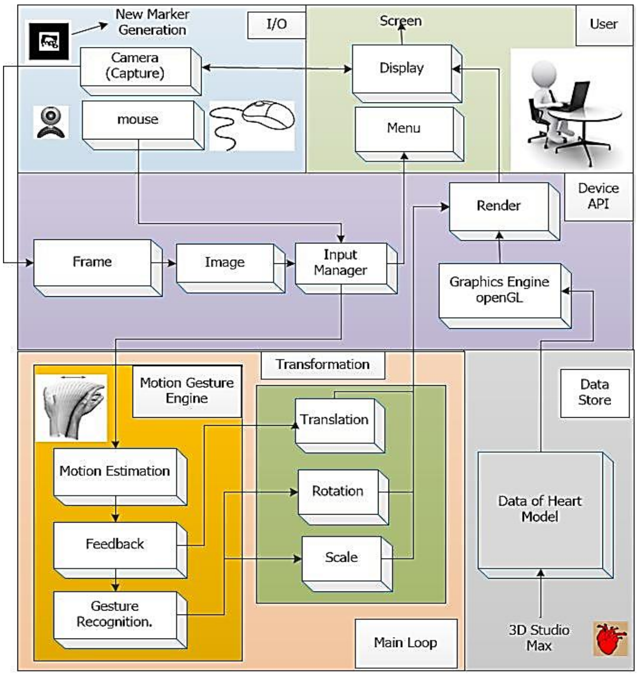

Fig 1 System Architecture of Project

i) Motion Gesture Creation

The gesture engine proposed can be divided into three constituents: Motion Estimation, Feedback and Gesture Recognition. Motion Estimation is the calculation of the motion value by comparing the previous and current frame. Then, the retrieval of data will be brought to the Feedback as a response from the user. A "live" graphical trace motion data is provided when a gesture is performed to ensure the user performs the gesture correctly. Next, the motion data is determined by the Gesture Recognition algorithm in deciding which gesture has been performed.

a. Motion Estimation

Block matching algorithm is the algorithm that is used. The last two frames capture by the camera is compared by the properties of the pixels in each frame. The comparison in term of value between colour and Euclidean Distance Formula is made in this project. Colour contains three variables which are the RGB values (R=red, G=green, B=blue). The equation of calculating the difference, d is shown below.

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + (q_3 - p_3)^2}$$
$$= \sqrt{\sum_{i=0}^{3} (q_n - p_n)^2}$$
$$where\ p, q \in r, g, b$$

Among the pixels, the difference of the RGB value between two frames which is lower than a Motion Threshold is then delivered as the motion vector. The Motion Threshold will then determine the motion detection level. For instance, the increasing value of Motion Threshold will lead the user to perform faster motion gesture so that the system is able to detect and calculate the motion value. The value of Motion Threshold is directly proportional to the frame rate of the system as the motion detection is based on the difference of colour pixels between frames,

$$Motion\ Threshold = kf_r$$
$$where\ f_r\ is\ the\ frame\ rate\ per\ second$$
$$k\ is\ a\ constant\ with\ range\ of\ [0.5\ , 1]$$

b. Feedback

The rendering of the transformation value on screen window or menu acts as a feedback to the user. This is an important stage as it provides user with the information regarding to the motion detection on the screen coordinate system. In order to do this, the coordinate of the detected motion which is the motion data that is calculated from motion estimation step will be tracked and returned to be displayed. Besides, this step will also guide the user to perform motion gesture on their purpose for the transformation. For translation, the return motion data is used directly for translation method to translate the virtual object. In contrast, rotation and scaling are needed to undergo gesture recognition to identify a specific operation to perform.

c. Gesture Recognition: Predefine Offset Screen Distance

The gesture recognition algorithm is essential to recognise the motion detected to perform certain operation. In this project, a Predefine Offset Screen Distance is implemented. The gesture algorithm is implemented by specifying a certain predefined distance offsets on the screen. The offset line divides the screen into two fragments. The mathematical solution to get the offset line is illustrated below.

$$offset\ line = \frac{window\ screen\ width}{2}$$

Next, a Recognition Threshold value needs to be declared to set a range at the centre of the screen. The setting of the range can isolate the noise because the motion estimation algorithm is highly sensitive to light factor. The Recognition Threshold cannot be too small (not effectively to reduce noise) or too big (limited recognition range). Therefore, the best range for Recognition Threshold is at 10% of the screen width size which is allocated at the centre of the screen. The formula of Recognition Threshold is shown below.

$$Recognition\ Threshold = 5\% * screen\ width$$

A simple picture of the Predefine Offset Screen Distance and its flow are illustrated in Fig. 2 and Fig. 3. Subsequently, a range is defined from the following steps in Predefine Offset Screen Distance. Thus, if the motion data is allocated at the left screen, a specific operation will be performed. Motion data at another screen will have different operations.

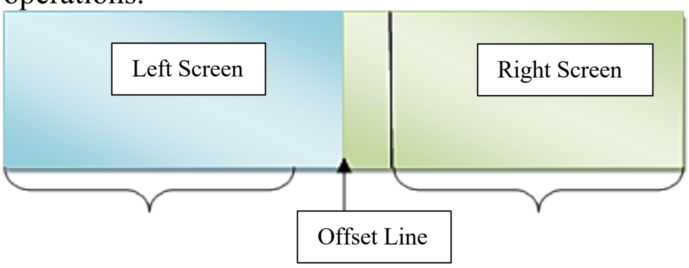

Fig 2 The proposed Predefined Offset Screen Distance

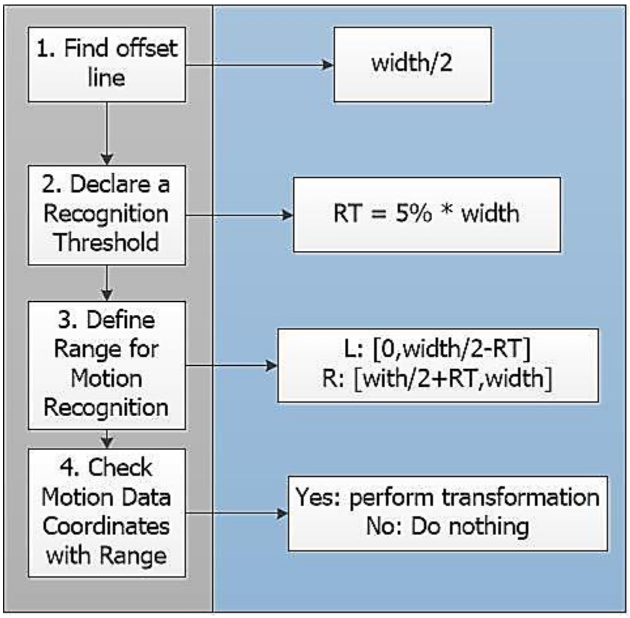

Fig 3 Flow Chart of Predefine Offset Screen Distance

V. SYSTEM DESIGN AND ANALYSIS

The prototype system contains one main window which capture images from the web camera and two sub windows.

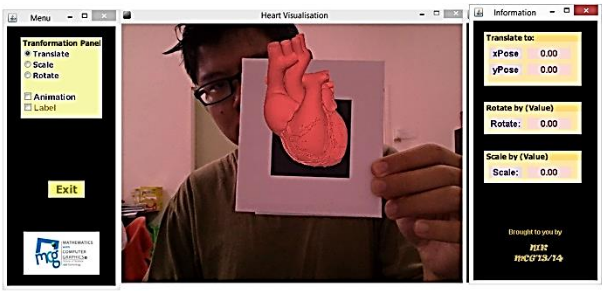

Fig 4 System Interface of Prototype

The flow of the system begins with the function of a web camera. The web camera will capture and process the image throughout the program. An external data which consist of the data of heart model is then imported into the system. Once a marker is detected, a virtual heart will be display on the screen.

Motion gesture algorithm starts with motion estimation where data motion is collected. The motion data is checked with a threshold value. If the motion data is higher than the threshold, transformation will occur. For translation, the return value from feedback is taken directly to translate the virtual heart. Meanwhile, the rotation and scale on this system need to undergo a gesture recognition algorithm named Predefine Offset System Distance. Different motion on different screen provides different orientation of transformation.

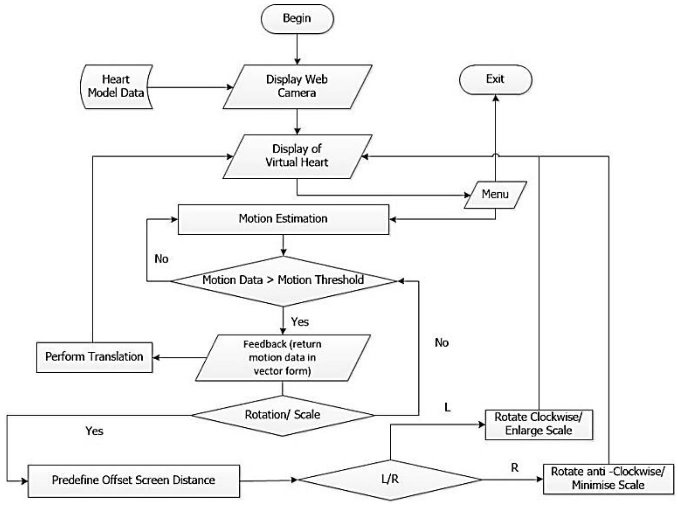

Fig 5 Flow Chart of the System

The computation of the complexity and Big-O notation is based on the Predefine Offset Screen Distance. According to Fig. 3, the first step yields the notation of $O(1)$ as a constant, $\frac{width}{2}$ is assigned to the variable offset line. The second step gives another notation of $O(1)$ from the constant value of Recognition Threshold. The third step operation also yields $O(1)$ because of the declaration of constant. In fourth step, the calculation refers to the Motion Estimation. The inner and outer loop executes according to the number of pixels in the screen. This gives a notation of O(n). In the inner loop, there are 6 variables r_1, g_1, b_1, r_2, g_2, b_2 being assigned to the constant which results in Big-O notation$(O(1) + O(1) + O(1) + O(1) + O(1) + O(1))$. Next, the square root shows the Big-O notation of $O(\sqrt{n})$ and $O(n^2)$ as it is a quadratic equation. Fig. 6 shows the computation of Big-O notation in details:

$$O(3) + O(n) * [(O(1) + O(1) + O(1) + O(1) + O(1) + O(1)) + O(\sqrt{n}) * O(n^2)]$$

$$= O(3) + O(6n) + O(n\sqrt{n} * n^2]$$

$$= O(3) + O(6n^2) + O(n^2)$$

$$= O(7n^2 + 3)$$

$$= O(n^2)\ time\ complexity$$

Big-O notation is calculated as $O(n^2)$. This indicates that the computational cost for the algorithm is costly.

VI. Experiments and Results

In this project, four experiments were carried out. The experiments were gesture test, distance test, lighting test and memory test.

i) Gesture Test

Gesture test is a test to check the efficiency of the proposed gesture algorithm. This test is conducted in an empty room with sufficient lighting condition. The distance between the camera and the marker is adjusted to 45cm. The test is divided into three sections: translation, rotation and scaling. The tests recognition is conducted by making motion that should be recognised as one of the classes in gesture creation. In translation test, tester tries to translate the virtual heart to a coordinate which is set to (100, 100). The result is then observed from the information menu which leads to the record of the time in completing the transformation from the origin (0,0) to the respective coordinate.. Rotation and scaling test are conducted by performing rotation and scaling based on Predefined Offset Screen Distance Recognition. The time taken to obtain the rotation of [0, 5] and scale factor of [0, 2] are also recorded. Each rotation and scaling test is run 10 times for each sample to determine the efficiency time of the gesture recognition. Table 1,2,3 shows the result of the gesture test with respect of time.

Experiment	Time Taken (seconds)
1	6
2	10
3	11
4	8
5	9
6	5
7	5
8	6
9	7
10	5

Table 1 The result of the gesture test on translation

Experiment	Time Taken (seconds)
1	6
2	7
3	6
4	6
5	5
6	6
7	5
8	7
9	6
10	6

Table 2 The result of the gesture test on rotation

Experiment	Time Taken (seconds)
1	13
2	14
3	12
4	12
5	14
6	13
7	15
8	14
9	14
10	13

Table 3 The result of the gesture test on scale

In gesture test, the result of the translation showed that the motion recognition algorithm did not constantly run smoothly. The inconsistency of the program is due to the instability of the formula of Euclidean which checks the pixel colour difference between two frames. The elimination of the noise in the motion detection did not occur eliminate as the devices used could not withstand the withdrawal of the noise. The noise is generated by reflecting of light or blurring of web camera resolution. In contrast, the rotation and scaling gesture recognition yield a better result where the time taken for rotating the virtual heart from 0 to 5 is averagely 6 seconds while scaling of the virtual heart from 0 to 2 is averagely 14 seconds. Hence, this test concludes that the motion recognition for rotation and scaling of the virtual heart is very efficient.

ii) Memory Test

Distance test determines the efficiency and accuracy of camera to detect marker and motion gestures based on distance. This test is important to test the accuracy and robust detection of marker in Processing 2.1.2. The experiment is set at a moderate lighting condition in a room. The apparatus used in this experiment are cardboard, Sensonic web camera with resolution of 1280x720, a 60x60mm pattern marker and a long ruler. A cardboard is used as a platform for the marker whereas the web camera is placed opposite the marker. A ruler is used to calculate the distance between the marker and the web camera. Distance parameters between marker and camera is the manipulation variable during the experiment. There are 5 tests conducted with different distances, $d =$

25cm, 35cm, 45cm, 55cm and 65cm. The visibility of the heart determines the suitable distance between the camera and marker for marker detection in Processing2. The result of distance test is illustrated in the following Fig. 6 captured by the web camera and snipping tool application.

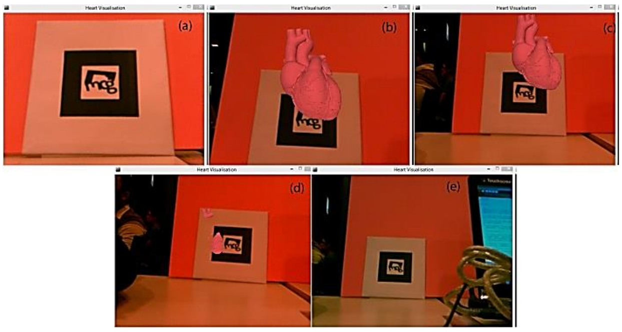

Fig 5 Result of distance experiment at d = (a) 25cm (b) 35cm (c) 45cm (d)55cm (e) 65cm

From the result, the optimum range of distance to detect a marker in processing 2 is between 35cm and 55cm. Distance of 25cm and below is too near for detection because the camera resolution cannot capture the complete black square area which causes the detection failure. Meanwhile, distance of 65cm and above is too far for the marker detection. This is presumably because of the pattern of the marker becomes blur when the distance of the camera is far and the camera resolution is low. Thus, the best distance for marker detection in Processing 2 is 45cm.

iii) Lighting Test

Lighting is significant for a lighting-based. To test the efficiency of the system based on lighting factor, the apparatuses such as 28 pieces of LED light source, a Sensonic web camera, a measuring tape, a marker and a board are prepared. The experiment is carried out in a dim light room where no external light source is being supplied. The distance between the light source and the marker is manipulated and measured using the measuring tape in centimeter(cm). The power consumption of the light source is 1.5W and the resolution of the web camera is 1280x720. There are 5 tests being conducted with different distances, d = 15cm, 25cm, 35cm, 45cm, 55cm, 65cm, 75cm and 85cm. The detection of marker is observed by the appearing of virtual heart. The data is recorded and further analysed in the next section. The result of lighting test is illustrated in the following figure capture by the web camera and snipping tool application.

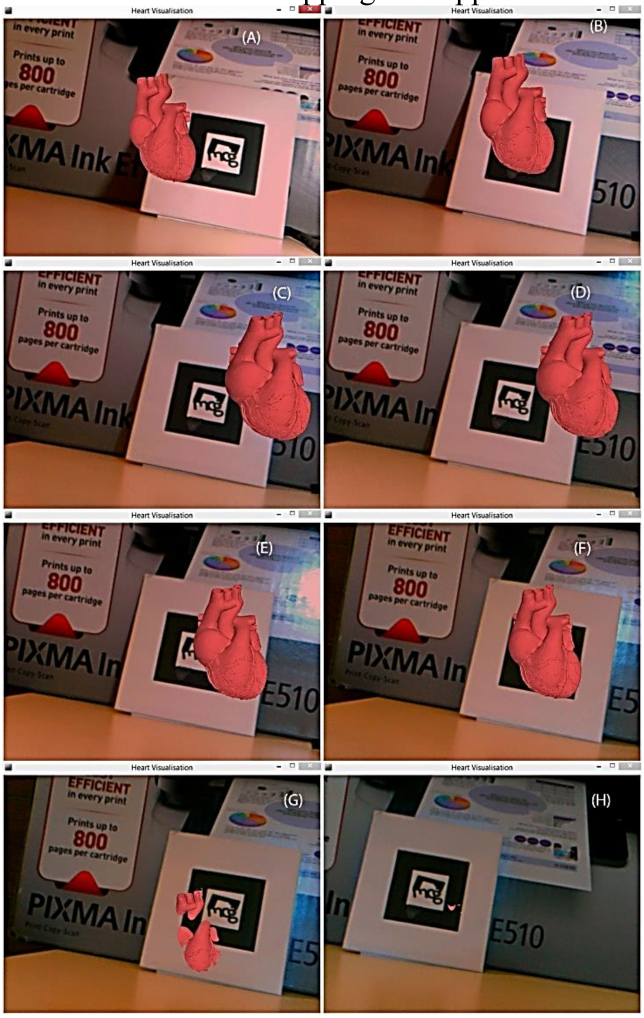

Fig 6 Result of distance experiment at d = (a) 15cm (b) 25cm (c) 35cm (d) 45cm (e) 55cm (f) 65cm (g) 75cm (h) 85cm

From the observation, the light distance ranges from 15cm to 65cm shows that the marker detection works well. However, the lighting at the distance of 75cm shows that the virtual heart loaded in AR system is not complete and lighting at the distance of 85cm yields no output. This indicates that lighting is important for a vision-based system. The insufficiency of light source will cause the camera unable to efficiently detect the pattern and edge of the marker.

iv) Memory Test

The objective of memory test is to test the memory consumption of the prototype system. When a Java program started, the Java Virtual Machine(JVM) will obtain memory from the operating system. The memory is called java heap memory. JVM will fully utilises this memory for new creation of object and operator. When objects are destroyed, the memory goes back to the Heap space. This experiment is tested on a Sony Vaio with i7-3537U Processor. The main memory on this electronic device is 8 GigaBytes(GB).The computer memory unit in Megabytes (MB) is the sole variable to be tested in this experiment. A Java Profiler is used to observe the memory usage of this program. The profiler use is VisualVM 1.3.7.

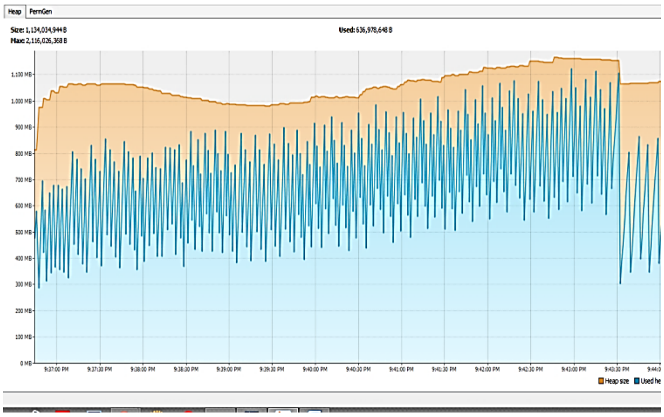

Fig 3 Result of memory consumption by Java Profiler

The heap memory is represented with beige colour while the used heap is represented as light blue colour. Based on the result above, the memory consumption for this prototype system in heap memory is between 1000MB and 1200MB. Meanwhile, the used heap shows a fluctuated line on the graph. The lowest of used heap is 300MB (when no comparison of frames) while the highest reaches to 900MB (when there is marker and gesture detection). As a consequence, this system consumes a lot of memory when the system is running, especially when gesture and marker detection occur. The reason is that the motion detection algorithm requires the checking of each pixels colour from time to time which consumes memory usage. Furthermore, the heart model that is loaded into the system is considered to be very complicated as many mesh topology is used for modelling. Apart from that, the GUI menu which is created in this system also consumes a lot of memory as many libraries are being called to build the menu.

VII. CONCLUSION

An interactive Augmented Reality System of Medical Imaging is created in this project. The interaction of this system primarily focuses on motion gestures. A 3D heart model is established by using the boundary scheme of solid modelling technique. The software that is used to construct the model is 3D Studio Max. Next, the data of the model is loaded into the system as Wavefront format (.obj). Although there is no consideration of dataset in medical imaging, the realistic of the heart model is due to the textures and lighting which are implemented on the model.

A combination of interactions makes this system a real-time application which allows the user to play with it. First of all, the user will interact with the system by performing motion gestures. The system will detect and recognise the gestures and then provides feedback to the user. As the user's goal is to translate the heart model to a certain position, correct motion gestures will yield the correct result. In addition, a sub menu was developed by using GUI Builder for mouse interaction. Users also have the alternative way to choose for the basic transformation such as translation, rotation or scaling of the model. Different transformations have different gestures recognition algorithms.

Finally, experiments are carried out to examine the accuracy of marker and motion gestures detection. The experiments carried out include memory consumption, lighting test and distance test.

REFERENCES

[1] R.A.,MischowskM.J., Zinser, J., Keeve, E & J.E, Zoeller, *AIntraperative Navigation in the Maxilofacial Area based on 3D Imaging Obtained By a Cone-beam.2007*.

[2] R. T.Azuma, *A Survey of Augmented Reality.* , 1997.

[3] HIH: Hands-free Interaction in the Hospital. 2013.

[4] Ruiz, J. & Li, Y. *DoubleFlip: A Motion Gesture Delimiter for Mobile Interaction. 2011*

[5] Ashbrook, D. & Starner, T. *MAGIC: A Motion Gesture Design* . 2010.

[6] *Layer (PHY) Specification*, IEEE Std. 802.11, 1997.