Waving at the Heart: Implementation of a Kinect-Based Real-Time Interactive Control System for Viewing Cineangiogram Loops during Cardiac Catheterization Procedures

Bart Suelze[1,2], Robin Agten[1,2], Philippe B Bertrand[3,4], Thijs Vandenryt[1], Ronald Thoelen[1], Pieter Vandervoort[4,5], Lars Grieten[4,5]

[1]Faculty of Engineering Technology, Hasselt University, Diepenbeek, Belgium
[2]Faculty of Engineering Technology, KU Leuven, Diepenbeek, Belgium
[3]Faculty of Medicine, Hasselt University, Agoralaan, Diepenbeek, Belgium
[4]Department of Cardiology, Ziekenhuis Oost Limburg, Genk, Belgium
[5]Mobile Health Unit, Hasselt University, Diepenbeek, Belgium

Abstract

The quest for a good natural user interface (NUI) between healthcare provider and technology is extremely important and will determine by enlarge its acceptance in clinical practice. Combining human gestures and voice commands may provide an interesting solution for human interaction with high tech clinical equipment. As proof of concept we focused on the setting of the cardiac catheterization laboratory where many interactions are required in reviewing the fluoroscopic cine-recordings. The new Kinect from Windows was used to develop a protocol allowing real-time interaction between the interventional cardiologist and the imaging software of the fluoroscopic image viewer. These interactions were based on single arm gestures to limit interference with invasive procedures. In situations where the cardiologist is unable to free up his hand, voice commands could be activated to execute the required actions. This natural user interface platform improved the efficiency in the catheterization lab.

1. Introduction

Technological advances in healthcare have taken a rapid pace the last few years. The quest for a good natural user interface (NUI) between healthcare provider and technology is extremely important and will determine by enlarge its acceptance in clinical practice. Combining human gestures and voice commands may provide an interesting solution for human interaction with high tech clinical equipment. Such an interface could be established using the Windows Kinect as a gesture acquisition platform. By developing a processing layer it could be possible to perform gesture recognition that could trigger required actions. The use of Kinect in health care is already found in a variety of clinical applications such as 3D mapping of bodies, diagnosing dyslexia [1], observing and analyzing posture and gait [2], home monitoring of edema [3] or manipulating medical images without a controller [4, 5]. However interfacing the Kinect in real life applications require certain conditions such as, ease of use, single arm movements to control basic function, voice recognition, good skeletal capture in low light conditions, even with hidden body. As proof of concept, we focused on the setting of the cardiac catheterization laboratory (Figure 1). During an invasive cardiac procedure there are many interactions required to scroll between subsequent fluoroscopic cine-recordings and between frames within one recording in order to properly visualize and locate coronary artery lesions, accurately position coronary stents and evaluate final results of the intervention (Figure 1). As the cardiologist is scrubbed and draped with sterile clothing, he cannot touch the control panel and has to rely on a technologist or assisting nurse to review the recorded cineloops.

2. Objectives

The aim of this study was to build an ease of use and low threshold interface allowing the cardiologist to view and review during the procedure all recorded fluoroscopic cineloops without the need for an assisting person.

3. Materials and methods

In our setup we developed a software application that consists of three different layers. A layer for signal acquisition and image capture, a layer for gesture processing and finally a layer that could execute various software commands independent of manufacturer.

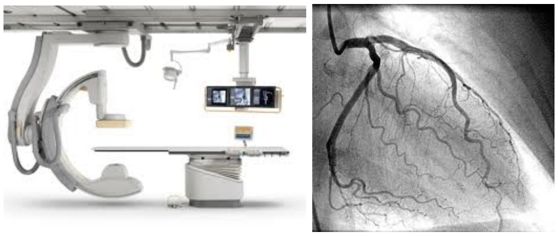

Figure 1. (top) is a representation of the catheterization lab with the X-ray tube used for making cineloop movies and the imaging software that allows real time or reviewing of the movies. (bottom) conventional angiogram of coronary arteries of the heart (colored black)

3.1. Signal acquisition using the Kinect from Windows

The input device for our application is a Kinect from Windows (Figure 2). It is equipped with a laser-based infra red (IR) projector and monochrome CMOS sensor. The IR projector is used to send a fixed speckle pattern towards the focused area, which is detected by the CMOS sensor and used to calculate depth data by triangulation against a hardwired pattern. The Kinect also embeds a tilt motor control for optimal positioning, a microphone array consisting of 4 microphones for sound capture and an RGB camera. The Windows Kinect has a near mode allowing accurate movement analysis from 40 cm up to 3 m and does not require full body recognition (seated mode). The software development kit v1.6 and the supported drivers allow application development. Since there was no gesture toolkit available the required gestures were algorithmically programmed.

3.2. Gesture programming and signal processing

For our application 5 different gestures were required that could execute 8 primary commands in the imaging software (Table 1). These were programmed in a .NET environment in C# language. The movements were continuously recorded using the RGB and IR camera and gesture recognition was done on pattern recognition without an initiating trigger. As an example, the Wipe Left Gesture (WLG) is used to go to the next movie, and in case the movie is in still-mode, it will scroll to the next frame.

The swipe movement has 2 different states, no action or the swipe action declared as:

```
private enum WavePosition
{ None = 0,
Swipe = 1,}
```

To initiate the motion, boundary boxes are defined where the right hand should start (Figure 3 a). These imaginary boxes are variable depending on the operator, i.e. they are calculated depending on the height of the person. The wireframe is constructed from the RGB image and the accuracy of skeletal deduction is variable between certain sensitivity intervals. The WLG is only activated by placing the hand in the left boundary box and moving it towards the left side of the body facing the back of the hand towards the Kinect. If a gesture is performed this takes 'n' frames (F_n) and a maximum time (T_n), thus the gesture could result in a success, failure or still in progress:

```
private enum WaveGestureState

{ None = 0,
Success = 1,
Failure = 2,
InProgress = 3 }
```

For each frame the Kinect decides if the gesture is still busy it will increase the IterationCount:

```
if (position == WavePosition.Swipe)

{ if (State != WipeGestureState.InProgress) {
State = WipeGestureState.InProgress;
IterationCount = 0;
StartPosition = position; }
IterationCount++; }
```

During the gesture, the arm has traveled a certain distance, and the algorithm recognizes the movement by comparing the new X and Y-position of the hand with the X and Y-position in the previous frames. Since the Wipe gesture induces no change in the Y coordinate of the hand, only the X-coordinate is variable. Upon moving the hand to the end position, the distance between the hand and the end-position becomes smaller.

```
if (hand.Position.Y >= elbow.Position.Y - Y_MARGIN
&& hand.Position.Y <= elbow.Position.Y + Y_MARGIN)
{ if (currentX ¡ lastX < X_MARGIN)
{ lastX = currentX;
tracker.UpdatePosition( WavePosition.Swipe); }
else
{ tracker.UpdatePosition(WavePosition.None); } }
```

Since the frame rate of the Kinect is fixed at 30 fps, the current frame is compared 30 times per second with the previous frame, and also to the WavePosition to validate if the gesture is still ongoing. Erroneous skeletal captures are possible resulting in incorrect movements (glitches), by building in a limitation on maximum distance traveled

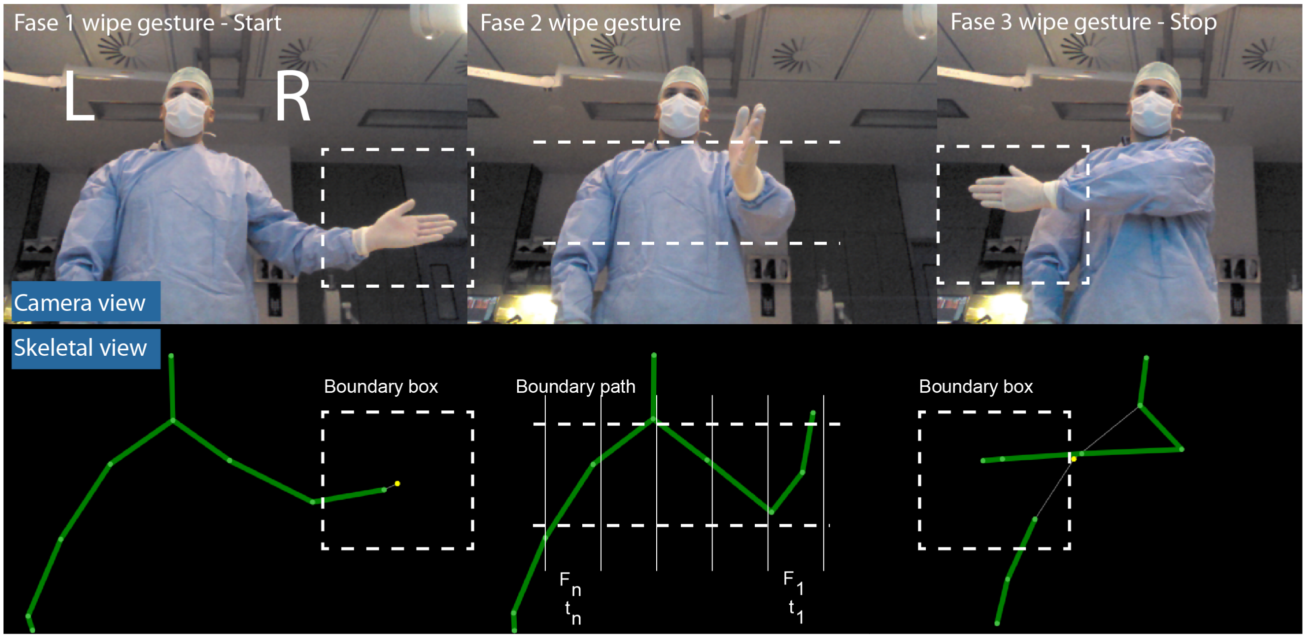

Figure 2. Step-wise demonstration of a wipe gesture. The gesture exists of 2 different phases, 1 static and 1 dynamic. The camera images are represented for the user but the skeletal capture is used for movement analysis. Boundary boxes are used to analyze and quantify the action.

per frame this is circumvented. The number of frames required to complete a gesture is important since every frame the algorithm decides whether the gesture is still busy, and will increase the iteration count of tracking.

```
if (tracker.State!= WaveGestureState.Success&&
tracker.IterationCount == REQUIRED_ITERATIONS)
{ tracker.UpdateState(WaveGestureState.Success);
GestureDetected(this, new EventArgs());
} else
{ tracker.IterationCount = 0; }
```

If the gesture is completed and successfully recognized de Wavegesture state changes to 1 and a trigger is given to execute the preprogrammed command.

3.3. Executing commands

Cineloop images and movies are stored in an Agfa Healthcare IMPAX database and recalled by a proprietary DICOM viewer for real-time analysis. For our application we developed an interfacing layer that uses shortcut- and preprogrammed control commands since many software tools are proprietary. If the algorithm detects a successful completed gesture the WaveGesture state is changed to 1 and a preprogrammed command is executed. In our setup these executing commands were: 'play', 'stop', 'next frame/movie', 'previous frame/movie' and adaptive scrolling.

Table 1. List of gesture and voice commands for the different imaging controls

Gesture	Voice	Action
Push	*start/stop*	play/pauze cineloops
Wipe Left	*next*	Next movie/frame
Wipe Right	*Back*	Previous movie/frame
Hold hand in end position	-	Adaptive scrolling between frames

3.4. Voice control

Voice control was programmed via the speech recognition engine that contains various language packages. During initialization of the program the grammar is created which describes all the words that need to be recognized (Table 1). The speech recognition mode is continuously listening and when a pre-programmed word is recognized within a certain confidence interval, an event is triggered.

4. Results and discussion

In our application we developed a tool that enables the interventional cardiologist, who is scrubbed and draped sterile, to interface with the already installed cineloop

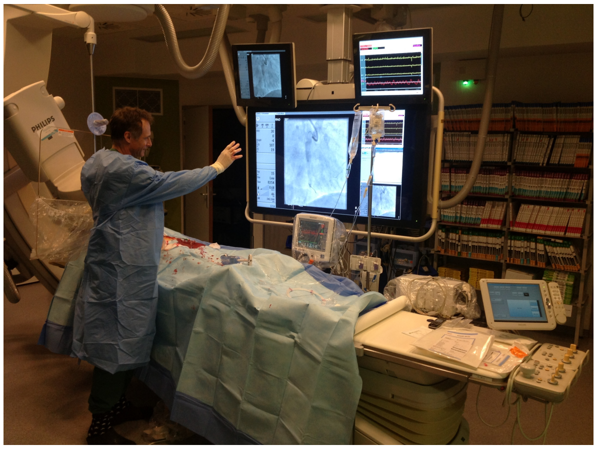

Figure 3. Real-life demonstration of the Waving at the Heart application during a catheterization procedure. The interventional cardiologist is able to manipulate the cineloops without assistance.

viewer in the catheterization lab, in our case the IMPAX database from Agfa Healthcare. This enables the cardiologist to manipulate cineloops with basic commands during the procedure with intuitive gestures. Since imaging software is often proprietary and different software packages are installed within and between various hospitals our solution requires versatility and out of the box applicability that could operate software applications in any graphic user interface, independent of manufacturer. Since no gesture engine is available, and the programmable gestures should remain simple to minimally interfere with the invasive procedures, all commands were algorithmically programmed focusing only on right arm gestures. Programmed gesture-software interactions were play, stop, next/previous image, next/previous frame and when movies are in still mode, a function for adaptive speed scrolling between the still-frames can be activated by holding the hand in the end fase position of the wipe. The near and seated mode of the Windows Kinect enables a good skeletal capture with minimal erroneous captures. In situations where the cardiologist is unable to free up his hand, voice commands could be activated to execute the required actions. Since the Cathlab is a noisy environment the broad confidence intervals were used allowing easier detection of the word-commands: start/stop and next/back. Since the application is straight forward in usage, does not require software or hardware upgrades and allows basic control of the imaging software in an intuitive way, this application resulted in a successful integration and technology adoption by the interventional cardiologist. This enabler improves the workflow efficiency by empowering physicians to navigate themselves to the correct movie or frame to properly visualize and locate coronary artery lesions, accurately position coronary stents and evaluate final results of the intervention (Figure 3).

5. Conclusion

The application of a Kinect-based control system offers a straightforward way for establishing a NUI between interventional cardiologists and cineloop viewing software during invasive cardiac procedures. This application improved procedural efficiency, is flexible for implementation in other interventional or surgical settings and requires no additional upgrading of hardware or software.

Acknowledgements

This study is part of the Limburg Clinical Research Program (LCRP) UHasselt-ZOL-Jessa, supported by the foundation Limburg Sterk Merk, Hasselt University, Ziekenhuis Oost-Limburg and Jessa Hospital. Also special thanks to Agfa Healtcare and the Microsoft Innovation Center Flanders for supporting this project.

References

[1] de Albuquerque A, Moura E, Vasconcelos T, Mendes L, Nagem D. Kinect Sensor used as a Support Tool In Clinical Analysis. Journal of Biomechanics 2012;45:304.

[2] Clark RA, Pua YH, Fortin K, Ritchie C, Webster KE, Denehy L, Bryant AL. Validity of the microsoft kinect for assessment of postural control. Gait Posture 2012;36(3):372–377.

[3] Hayn D, Raschhofer S, Falgenhauer M, Modre-Osprian R, Fruhwald F, Schreier G. Edema detection for heart failure patients in home monitoring scenarios. Computing in Cardiology 2012;39:93–96.

[4] Ebert L, Hatch G, Thali M, Ross S. Invisible touchControl of a DICOM viewer with finger gestures using the kinect depth camera. Journal of Forensic Radiology and Imaging 2013; 1(1):10–14.

[5] Gallo L, Placitelli AP, Ciampi M. Controller-free exploration of medical image data: Experiencing the kinect. In 24th International Symposium on Computer-Based Medical Systems (CBMS). IEEE, 2011; 1–6.

Address for correspondence:

Lars Grieten
Schiepse Bos 6, 3600 Genk, Belgium
Lars.Grieten@ZOL.be