Touch-free gesture based control of medical devices and software based on the Leap Motion Controller

Stanislas MAUSER[a,1] and Oliver BURGERT[b]
Reutlingen University, Faculty for Informatics, Germany [a,b],
Stanislas_Christophe_Yves.Mauser@student.reutlingen-university.de [a]
Oliver.Burgert@reutlingen-university.de [b]

Abstract. There are several intra-operative use cases which require the surgeon to interact with medical devices. We used the Leap Motion Controller as input device and implemented two use-cases: 2D-Interaction (e.g. advancing EPR data) and selection of a value (e.g. room illumination brightness). The gesture detection was successful and we mapped its output to several devices and systems.

Keywords. gesture recognition, touch-less interaction, human machine interface, medical device control

Introduction

Goals

During a surgical intervention, the surgeon is dependent on the availability of patient information such as risk areas of tumor margins, and he hast to control surgical devices like endoscope, x-ray, drill etc. Nowadays, this control is either performed by using dedicated sterile equipment, foot pedals or by directing a surgical assistant or nurse. This impedes the execution of the surgery, increases the risk of infection for the patient and is generally associated with a higher expenditure of time (see [1], [2]).

For a subset of the use cases described above, a contact free, gesture based controller, which is used by the surgeon might be helpful. Such a touch-less approach should be in reach for the surgeon while operating and without the need for special gloves or additional tools.

In [1], several surgeons were interviewed whether they want to instruct other persons to acquire information or get it themselves by using e.g. new gesture recognition based system technologies. Over 80% of the surveyed surgeons preferred to search for information themselves and not by instructing another person. Additionally, the availability of touch-based systems (like smartphones and tablets) and pose tracking systems like Kinect pave the way towards additional intuitive interaction devices.

[1] Corresponding Author.

State of the art

There exist multiple examples for contact free input devices: In [3], a sterile solution to navigate the planning system MeVis with a Wii Remote Controller is described. This allows for gesture recognition, but the controller needs to be touched. Gesture recognition can be performed based on tracked surgical instruments. In [4], such an approach is used to detect surgical gestures performed by a Polaris® navigation system pointer. A Microsoft Kinect is used for contact free interaction with the InVesalius program. Here the surgeon may contact-free interact with the program with one hand, to visualize 2D and 3D images [5].

A commercially available solution is the Mi-Report developed by Fraunhofer HHI for Karl Storz which allows for presentation of patient data with contact free gestures control, specifically for use in the sterile medical field [6]. The camera system is called HHI Handtracker and is based on an infrared camera system, which allows a fast and robust detection and tracking of fingers and gestures. The cameras are mounted to the hand orthogonal to the ceiling. By using infrared-based cameras, the system is also less insensitive to unwanted light of other devices or the sun. Several fingers are recognized and processed in real time (50 Hz). The 3D coordinates are then passed to the application.

The available systems have several drawbacks: Either they require a dedicated input device for gesture recognition (navigation system, Wii controller), or the gestures recognition is relatively coarse. Furthermore, large gestures result in physical fatigue.

1. Methods & Materials

We used the new Leap Motion Controller: A small device, not bigger as two fingers, but with the capability to track “ten fingers up to a precision of 1/100th of a millimeter” [7]. The tracking volume of the Leap Motion Controller is about 20 to 600 mm semicircular around the device. We used a development kit including a Leap Motion Controller revision 0.6.5., C# as programming language, and the Visual Studio 2012 IDE.

Development hardware included a Apple Mac Pro running two Intel Xeon Westmere CPUs, 24GB RAM, two ATI 5700 HD graphic cards and Windows 8 in a virtual machine. For evaluation and presentation, a Lenovo W520 running a Intel i7 CPU, 8GB RAM and a nVidia Quadro 2000M graphic card was used.

1.1. Contact free gesture solution prototype for medical devices

The high precision of the Leap Motion offers much more possibilities then similar devices like the Microsoft Kinect. But basically, the problem of finding a gesture which is easy to learn and intuitively to use, is the same as with other devices. There are commonly accepted gestures for 3D contact-free input devices, yet. Best practices gestures from touch-based systems, like Apple iPhone or similar Point-Gestures can be implemented with the Leap Motion, but must be seen as starting point for the development of truly intuitive 3D input gestures.

In the medical context, we identified different classes of interaction: binary interaction (e.g. a simulated mouse click), simple 2D interaction, like advancing to the next picture or patient record entry, entering a value (e.g. adjusting OR table height or control a light source), point and click for selection of objects, and complex 3Dinteraction, e.g. rotating a volumetric patient reconstruction. For our first prototype, we implemented the “simple 2D interaction” and “entering a value” use cases.

1.2. 2D Interaction

2D interaction is realized by performing a swipe gesture to the right or to the left, orthogonal over the Leap Motion Controller. To detect a swipe, we implemented a two-staged process: First, we used the build in Leap SDK swipe detection. On top of that, we defined three threshold states. The first threshold starts the swipe process and defines whether it is a left or right swipe. The next two thresholds and the speed of the finger or hand ensure that the gesture was really intended. The gesture detected can be mapped to a key-press event (e.g. cursor left, cursor right), which allows for control of applications which were not developed with contact-free interaction in mind. We implemented a mock-up of an electronic patient record viewer to show the feasibility of the approach.

Evaluation of the detection capabilities and error rate was performed by controlling the PowerPoint presentation of a 90 minutes lecture on computer science (Informatik II) at Reutlingen University. The contact-free 2D interaction was the only input device for slide advancement. The long duration of 90 minutes and the real lecture situation ensured that the teacher was not concentrating on using the interaction device correctly but on the teaching situation. The professor had no training on the device and started after a 20 seconds test. Afterwards, a video recording of 60 minutes of the lecture was analyzed to verify how many unwanted gestures were performed and how many gestures failed.

1.3. Entering a value

Since there is no commonly accepted interaction scenario for entering a value, yet, we implemented a control interface which allows for mapping different types of gestures to a method which finally passes the calculated value to the target application.

The gestures for entering a value take the best practice gestures from touch devices into account. We used the built-in gestures of the Leap Motion and additional values like pitch, yawl or the y-position of the hand, which can be read out from the Leap Motion. Gestures like “drag and swipe” seem to be intuitive, but detection of start and stop position is difficult. Along with this, the problem of how the given minimum, maximum and current value shall be represented occurs.

1.3.1. Counting fingers:

From a technical and logical viewpoint, this is the simplest gesture. Analog to the sign language, the surgeon can modify the value by showing up to five fingers orthogonal over the leap motion Controller. This can be mapped to the values 0-5 or to a percentage of the value range. E.g. for setting of 20% of the maximum value, a surgeon shows only one finger, etc. The problem of this solution is the limitation on only five levels. The termination of the gesture must be defined properly to prevent misinterpretations: After the desired value has been set and the hand is removed from the detection area, the fingers successively disappear and the value would decrease to one or zero.

1.3.2. Mapping the palm orientation:

A rolling gesture of the palm, analogue to an airplane's wing that rolls down to the right or to the left for navigation, was implemented as an alternative. The vertical palm position, relates to 50% of the maximum value. By rolling the palm to the right, the value increases and when rolling to the left, the value decreases.

Like in the counting fingers scenario, unwanted values sometimes occurred when removing the hand out of the tracking area. This effect happened much less frequently than in the number of fingers gesture, but often enough that it hinders productivity. Therefore, the number of detected fingers on the hand that performs the roll gesture was added as an additional condition. For example, if two fingers are set as the restriction modification, the values will only change when at least two fingers are recognized by the controller. So if the surgeon has set the desired value, he can simply make a fist and remove the hand without concerns out of the detection area.

1.3.3. Mapping the vertical hand position:

The third gesture approach is based on the vertical movement of the palm position. In this scenario, the distance between the location of the controller and the measured center of the palm will modify the value. In the prototype, 250 sampling stages were implemented, starting from about 15 cm above the controller and ending at about 40 cm. In the graphical prototype this gesture is mapped to a vertical slider.

In most cases the values rest untouched, when the hand is removed slowly to the side out of the detection area. But in few cases, errors still occurred and moreover, for the use case during the OR course, it can´t be a requirement to remove the hand slowly. To compensate this, the technique of restriction by a certain number of fingers, as described above, was introduced.

1.3.4. Lock and unlock gesture:

In analogy to the Mi-Report [6], a lock-unlock gesture was implemented for this gesture concept. To reduce the possibility of unintended gesture input during the intervention, a defined gesture for locking and unlocking the detection of gestures was implemented. For unlocking the gesture detection, the surgeon has to hold his palm orthogonal over the controller with all fingers extended and as straight as possible for a defined duration. In the prototype these conditions can be adjusted at run time, since it’s not entirely clear how precise the check of the gesture must be. Adjustable variables can be for example the length of time, the gesture has to be executed without interruption, the minimum number of fingers it has to recognize, and the maximum pitch and roll valued of the palm.

This gesture toggles the locking and unlocking condition, so that the same gesture can be used for both. But additionally the gesture detection is automatically locked after a defined amount of time, when no changes to the light value occur. The current default value in the prototype is three seconds.

2. Results

2.1. 2D interaction study results

During 60 minutes, 99 gestures were performed: 78 intentional gestures and 21 unwanted gestures. Unwanted gestures resulted from arm movement, putting down chalk, etc. 37 of the intentional gestures were performed to correct unwanted gestures. 58 of the 78 intentional gestures were recognized correctly and triggered a slide change. Thus, ~25% of the intended gestures were not fully recognized because they were not performed in the working space or the gesture was not performed properly.

In an interview, the performing teacher stated that the interaction was simple and fun, but the false positive rate was too high for routine use.

2.2. MDeviceControl Gesture Prototype

In the prototype, the gestures approaches were all mapped to horizontal or vertical sliders to show potential testers the effect of their gesture movements. Both gestures, the palm roll gesture and the vertical movement gestures are implemented and visualized in the prototype with and without the additional condition of n fingers, for restriction of unwanted value modification during the gesture process.

In the prototype, as can be seen in figure 1, the lock- and unlock gesture conditions can be adjusted in the 3rd box. In the second box the detected gesture is visualized in values. These both boxes are just for the prototype and not for the productive working solution implemented. On the bottom left, the pitch and roll state of the hand is visualized by arrows, which is more intuitive as plain numbers.

First tests with untrained subjects showed that with four or more fingers for three seconds and a maximum pitch/roll-deviation of 15° it is quite sure that this gesture is not unwanted performed during the OR, but can still be carried out quickly and easily.

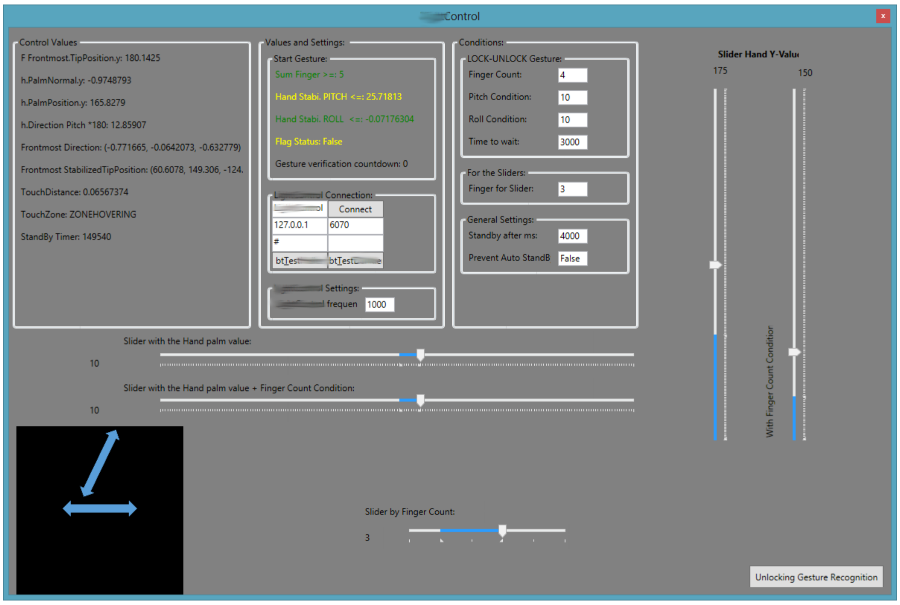

Figure 1. Screenshot of MDeviceControl Gesture Prototype

3. Conclusions & Discussion

The Leap Motion Controller is a stable 3D gesture input device which has potential applications in intra-operative situations. The use-cases have to be carefully selected and the best gestures for each application have to be detected and implemented. The developed MDeviceControl prototype is a valuable tool for user testing and rapid development.

Based on the studies and tests performed, it is clear, that a lock- and unlock gesture is mandatory, especially if precise and intuitive tool control is crucial.

Still not evaluated are the optimal light conditions for the Leap Motion Controller, since unwanted light and the impairment of its emitted IR light could result in reduced recognition rate. We did not detect interferences with surgical navigation systems using IR light, yet. The next step will be to simulate lighting conditions found in operating theaters with or without daylight access and OR illumination.

References

[1] C. Dressler, T. Neumuth, M. Fischer, O. Abri, G. Strauss: *Intraoperative Bedienung einer elektronischen Patientenakte durch den Operateur*, In: HNO, vol. 59 issue 9, Universität Leipzig, September 2011, DOI: 10.1007/s00106-011-2331-y, 900-907.

[2] P. Chojecki, U. Leiner: *Berührungslose Gestik-Interaktion im Operationssaal*, i-com, Vol. 8, Issue 1, Mensch-Computer-Interaktion im Operationssaal, DOI 10.1524/icom.2009.0003, 13-18.

[3] F. Ritter, C. Hansen, K. Wilkens, A. Köhn, H. Peitgen: *Benutzungsschnittstellen für den direkten Zugriff auf 3D-Planungsdaten im OP*, i-com, vol. 8 issue 1, Mensch-Computer-Interaktion im Operationssaal, DOI: 10.1524/icom.2009.0005, 24-31.

[4] S. Franke: *Integration intraoperativer Patientendaten in ein TIMMS Patientenmodell am Beispiel der Lokalisation des Sulcus Centralis*, Diploma Thesis, Universität Leipzig, 2010.

[5] G. Ruppert, L. Reis, P. Amorim, T. Moraes, J. Silva: *Touchless gesture user interface for interactive image visualization in urological surgery*, World Journal of Urology, Ausgabe 5, 2012, DOI: 10.1007/s00345-012-0879-0, 687–691.

[6] S. Schrader, U. Leiner: *Neuartiges System zur Visualisierung von Patientendaten mit berührungsloser Handgestik-Steuerung*, In: Workshop-Proceedings der Tagung Mensch & Computer, Berlin, 2009, ISBN: 978-3-8325-2181-3, 302-304.

[7] Leap Motion official website at https://www.leapmotion.com/ (last access: 14.06.2013).