Virtual Reality using Gesture Recognition for Deck Operation Training

Girts Strazdins[†], Birger S. Pedersen[‡], Houxiang Zhang[‡]*
[†]Department of ICT and Natural Sciences,
Faculty of Information Technology and Electrical Engineering
[‡]Department of Ocean Operations and Civil Engineering
Faculty of Engineering
NTNU, Norwegian University of Science and Technology,
Trondheim, Norway
{gist, birgersp, hozh}@ntnu.no

Pierre Major
Offshore Simulator Centre
Aalesund, Norway
pierre@offsim.no

***Abstract*—Operation training in simulator environment is an important part of maritime personnel competence building. Offshore simulators provide realistic visualizations which allow the users to immerse within the scenario. However, currently joysticks and keyboards are used as input devices for deck operation training. This approach limits the user experience - the trainees do not practice the gestures that they should be giving to the crane operators. Conversations with operation experts reveal that trying and experiencing the gestures is an important step of the practical training. To address this problem, we are building a gesture recognition system that allows the training participants to use natural gestures: move their body and hands as they would during a real operation. The movement is analyzed and gestures are detected using Microsoft Kinect sensor. We have implemented a prototype of a gesture recognition system, and have recorded data set of 15 people performing the gestures. Currently we are in the process of improving the system by training the recognition algorithms with recorded data. We believe, this is an important step towards high-quality training of maritime deck operations in immersive simulator environment.**

***Index Terms*—Virtual reality, Gesture recognition, Marine operation.**

I. Introduction

Marine operations are getting increasingly demanding, complex and integrated between different parties. There is an increasing trend of deck operations in recent years, which increases the need for the ships crew to be completely familiarized with the precautions and preparations necessary during deck operations.

Training programs have had success in reducing risk and improving efficiency of marine operations by training in simulators to improve overall understanding of operations to be performed. Illustrative techniques to visualize procedures, and thereby further improve (shared) situational awareness, are lacking, though. At NTNU Aalesund Campus and Offshore Simulator Centre AS (OSC), it is possible to verify and train for unique deck operations involving several ships and a rig before the operation takes place in reality, as shown in Figure 1. The centre in Aalesund is currently training 1200 professionals in the maritime industry in simulators and will be an important asset in the research. However, some aspects must be improved. First, recent feedback from personnel on deck operation training in Aalesund has raised concerns for major accidents to take place. Typical areas of concern are complex design and systems handled by less qualified crew, unclear roles and responsibilities, misunderstandings in communication, language and cultural differences, as well as increasingly complex procedures and check lists. The other technical issue is the reliability and credibility of deck operation training. As seen in the Figure 1, the staff use joysticks to control the virtual operators in the simulator. Although the visualization is realistic, the interaction is different from the real on-board deck operation and far away from reality. The training procedure is more like playing a game, rather than operating on offshore deck.

All these issues have caused a call for actions addressing training to a much higher extent in order to sustain and improve the safety performance of the industry.

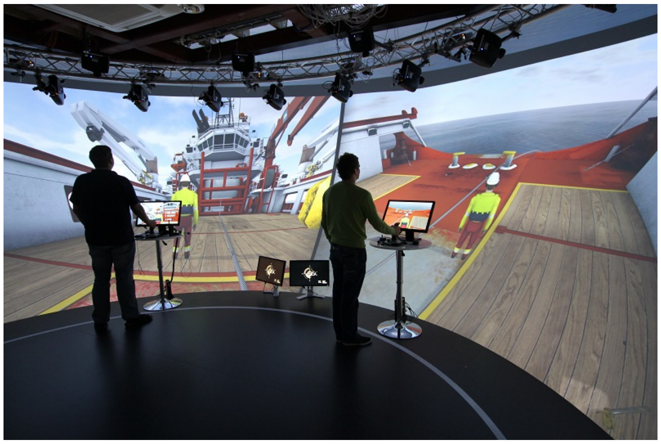


Fig. 1. Deck operation training stations at the Offshore Simulator Centre

To establish and maintain effective working relationships with all deck operations in different stations, it is beneficial to develop intelligent system to simulate the crew behavior

* Corresponding author

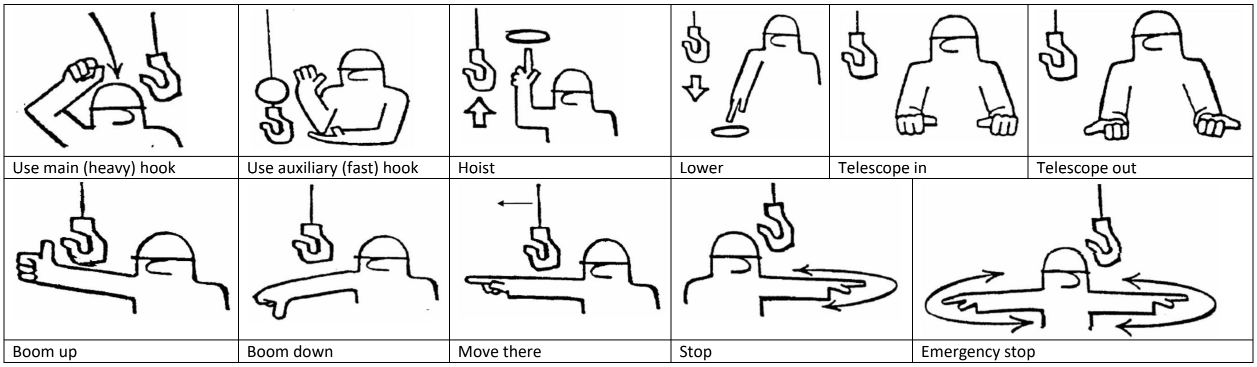


Fig. 2. Gestures used during lifting operations

and recognize their gestures, for example, to help crane operator to recognize international crane signals, shown in Figure 2. Virtual reality (VR) technologies have accelerated rapidly in the recent years. Currently available sensor devices such as Microsoft Kinect allow to accurately track the user's position, orientation and motion. Such possibilities were not available ten years ago. Creating a VR-based simulator for behavior recognition during deck operation is of great interest in this domain. However, there is no relevant research and applications implemented for deck operation. The joint project between Mechatronics lab at NTNU Aalesund and OSC aims to develop a prototype of VR simulator that can provide behavior recognition during deck operations. We are developing a case study on behavior recognition of Able bodied seaman (AB) giving signals to crane operators. The simulator uses Microsoft Kinect sensor to recognize all the crane operation gestures shown in Figure 2. A Graphical User Interface (GUI) shows the recognized behavior.

II. Related work

Numerous gesture detection systems have been proposed previously, including Hidden Markov Models [1], Finite State Machines [2], Dynamic Time Warping [3], fuzzy logic [4], and specific approaches, for example, encoding of trajectories as characters, and application of string matching algorithms [5].

This paper does not consider development of a new gesture detection algorithms or methods. Rather, we combine existing tools with and develop a solutions for detection of a specific gesture set: crane operation gestures. Gestures defined in the NORSOK standard [6] have been chosen as a basis for this project due to multiple reasons. First, it is a standard widely used in Norway, where significant amount of offshore operations are performed. Secondly, most of hand signal in NORSOK are common among multiple standards [7].

Different commercial vision-based solutions are available, including Microsoft Kinect [8] and PlayStation Move [9]. High-end systems are available for more specific scenarios, such as WorldViz for interaction space up to 50x50 meters [10].

The Microsoft Kinect was chosen to use here because it is a well-proven commercial technology with high accuracy and rich software development kit (SDK). It is a vision-based system, and completely unobtrusive. The users do not have to wear anything and operate naturally. Non-wearable sensors are also strongly recommended by professionals and DNV-GL. However, one of the main drawbacks of vision-based systems is the lack of tactile feedback, yet that is not important in our interaction scenario.

Although not part of this paper, Kinect can be combined with wearable solutions, such as CyberGlove products [11] for arm and finger tracking. Full-body wearable systems, such as XSens MVN [12] are too obtrusive for natural training scenarios.

There is a significant amount of previous research on gesture recognition and benchmarking. Different data sets with recorded gestures are available, such as NATOPS gesture database [13], [14] and Chalearn gesture challenge [15]. However, none of the previously recorded data sets were directly usable in our scenario. Therefore, we had to record our own database of gestures.

The contribution of this paper is development and evaluation of a crane operation gesture recognition system used for deck operation training. To the best of our knowledge, no other systems have been presented in the literature before. The paper will be organized as follows. Section III will give an introduction to our gesture recognition structure design including performance evaluation and the system setup. After that, the user study is presented in section IV. The experiment information, accuracy evaluation and use feedback will be explained in details. The tests confirmed project idea and show the VR technology could enhance the effectiveness of marine training performance. The conclusion and future work are given in the end.

III. Gesture recognition system

The gesture recognition system architecture is shown in Figure 3. Microsoft Kinect sensor is selected for user body tracking. Several software frameworks are supporting the Kinect sensor, including Kinect SDK, and Open NI [17]. Based on available technology research, we chose to use the

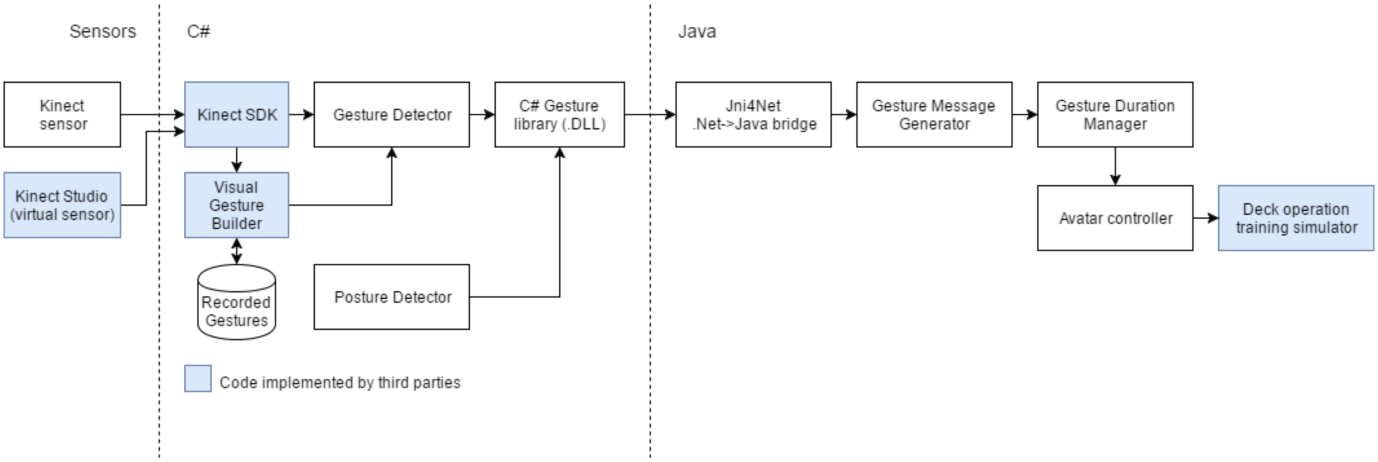

Sensors
C#
Java
Kinect sensor
Kinect Studio (virtual sensor)
Kinect SDK
Visual Gesture Builder
Recorded Gestures
Gesture Detector
Posture Detector
C# Gesture library (.DLL)
Jni4Net .Net->Java bridge
Gesture Message Generator
Gesture Duration Manager
Avatar controller
Deck operation training simulator
Code implemented by third parties

Fig. 3. Gesture detection system architecture

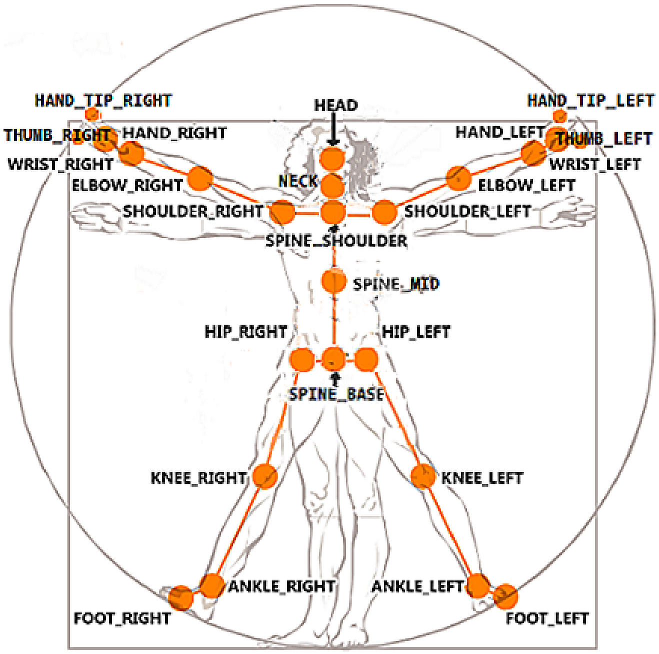


Fig. 4. Joints recognized by second version of Microsoft Kinect sensor: three joints per palm, 25 joints in total. Image courtesy of [16]

official Microsoft Kinect SDK as the most feature-rich and stable framework. It allows us to detect up to six people in the scene, and track joint locations of two users. This project used Kinect version 2 sensor due to two significant advantages over the first version: more detailed palm tracking (three joints per palm allowing basic finger tracking, see Figure 4) and higher accuracy. These aspects are important in our scenario, where several gestures differ almost exclusively by finger configuration (such as *Boom Up* and *Boom Down*, see Figure 2).

There is one limitation of choosing the Kinect SDK. It runs only on Windows platform and requires programming in C# or C++ with the .Net framework. Our project had necessity for integration with other software solutions written in Java. Therefore we use Jni4Net bridge solution [18] to translates method calls between C# and Java environments. Evaluation shows that performance of this bridging solution is effective to deliver the joint information at 30 frames per second - the rate at which Kinect sensor is operating.

We have used Visual Gesture Builder (VGB) to record gesture patterns. It is a tool provided by Microsoft as part of their Natural User Interface Tools package [19]. The tool set allows to define, record gestures and later perform live detection of recorded gestures. Our contribution was recording and tuning of the gestures according to the NORSOK standard, and integration of the detection events with the rest of the training system software.

Our system is extensible to other gesture sensors and other types of input devices. For example, we could add a keyboard as one mechanism to signal gestures manually. We use the Observer Pattern to disseminate events in the system. Gesture detection components generate events, any component in the system can subscribe to receive these events. The Gesture Message Generator component translates device-specific interface to system-wide general messaging interface. As long as a new sensor has a software driver that can generate these generic event messages, a new sensor can be added to the system.

The gestures have also the temporal dimension. Each gesture is valid for a period of time. A decay mechanism has been implemented. When the Gesture Message Controller raises an event, it is detected by Gesture Duration Manager. This component manages the current state of all the gestures: probabilities that each gesture is detected. It adds a decay for each probability. Currently we use a simple mechanism: periodic events are generated every 100ms and at time moment t the probabilities p_i of each gesture are decayed using formula $p_{it} = p_{it-1} * \alpha$, where α is a constant coefficient. We use value 0.9 for α.

The detected events are converted to a message which is sent to the Offshore Simulator Centre's software components. Here the message is interpreted and the avatar in simulation shows a gesture accordingly, see Figure 6. The simulator software is closed, and this research project did not have any modification to the simulator. Our software sends messages equal to those sent by joysticks in the current simulations.

For testing purposes we have built a simple Graphical User Interface (GUI), see Figure 5. It shows the detected person, location of all detected joints. It also shows detected gestures and confidence of detection.

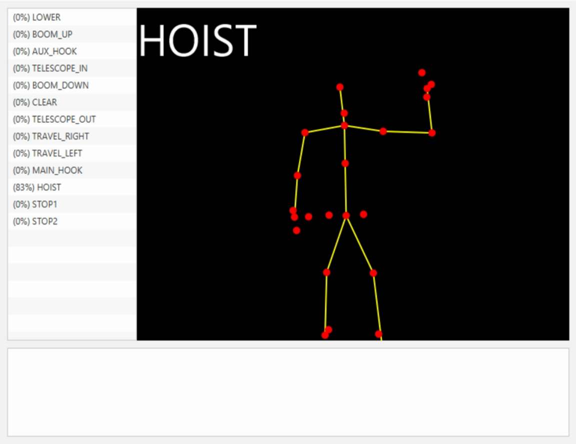


Fig. 5. Graphical User Interface (GUI) for testing purposes

The software uses a Kinect for Windows 2.0 sensor. To utilize the sensor, the computer running the software must have a 64-bit (x64) processor with two or more (physical) cores of 3.1GHz or more, and 4GB or more RAM. The computer must also have a USB3.0 connection ports, and a graphics card (GPU) that supports DirectX 11. The software must be run in either the Windows 8 or 8.1, Windows Embedded 8 or Windows 10 operating system. The application is built for Java 8 (or newer). Thus, a Java Runtime Environement (JRE) of this version must be installed on the system.

On one of the computers used for development (which uses a quad-core 3.6GHz CPU), only 1% of the CPU is in use. At the same time, approximately 110MB of RAM is in use. It is safe to say the application uses very little of the computers resources to run. The image-processing algorithms, to generate body joint locations from the depth images [20] as well as gesture recognition, are executed on the Kinect device itself and does not affect the runtime resource-use of the application.

For development of the system, the following additional software is required:

- Microsoft Visual Studio 2015 (or newer)
- Apace Maven 3.0 (or newer)
- Java Development Kit (JDK) 1.8 (or newer)
- Microsoft Kinect for Windows SDK 2.0

IV. User study

We have built a prototype system and are currently in the stage of improving it. A user study with a twofold motivation has been performed. First, the research team wanted to get feedback of the prototype system accuracy. Second, we recorded videos of several persons performing the gestures, to train the gesture detection algorithm. The gathered data serves as a valuable resource in system improvement.

A. Experiment description

In total 15 test subjects participated: six master students, six bachelor students and 3 researchers. The average age of participants was 27 years, 14 were male, one female. The experiment was performed over two days: 9 participants on the first day and 6 participants on the second day.

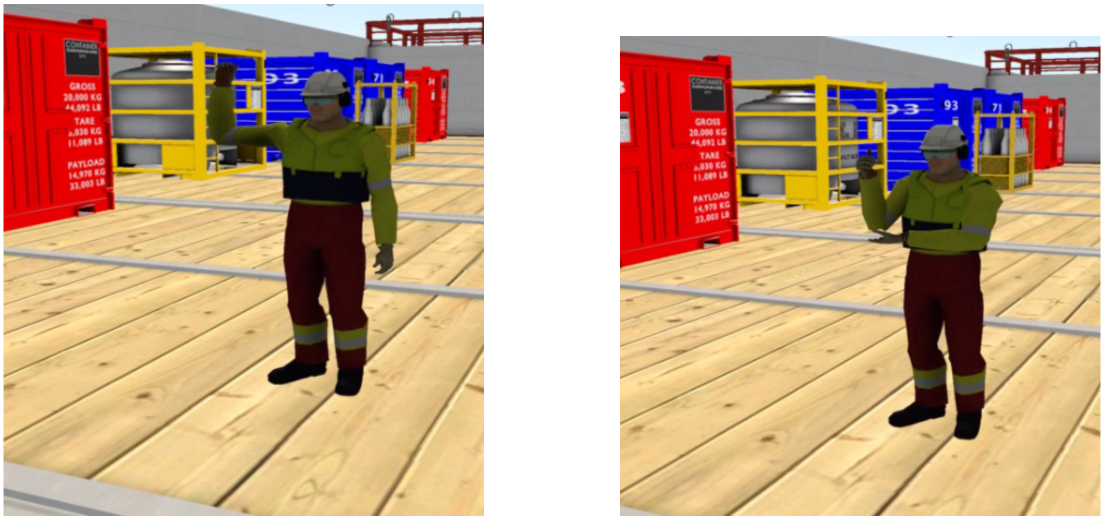
The experiment consisted of three phases. In the first phase, the participant was shown a video containing all the 11 gestures that our system can recognize so far. In addition, they got an explanation in person and could ask any questions in case of misunderstanding. Our definition of gestures was based on NORSOK standard and an interview with crane operation training expert in Aalesund. In the second phase, the same video was shown and the participant had to repeat the gestures shown in the video. Kinect sensor recorded all their movements using Kinect Studio tool. The gesture recording was done twice: once focusing on right-handed gestures and once - left handed. In the third phase participants were asked to fill out a questionnaire describing their subjective perception and related background, age and gender.

During the tests we recorded infrared image stream from the Kinect sensor. The total amount of recorded data reaches 75 Gigabytes. The recordings allow us to replay the user movements - we can re-run our gesture detection software as if the users are still present. The data can be used both to train and test the gesture recognition system.

During the first experiment day, it was discovered that the demo video was inaccurate and hard to comprehend for the participants. It showed an avatar performing the gestures. Therefore, for the second day experiments we recorded a new video with one of our team members showing the gestures accurately and clearly.

B. Accuracy evaluation

During user study preparation phase, it was discovered that several gesture definitions used in the prototype system differed significantly from the standard. The prototype system was made following the animations shown in the OSC training system. This system uses a *dialect* of the NORSOK gestures. E.g., in *telescope out* gesture the standard says that one should hold hands still above waist and below shoulders. The dialect had this gesture with hands moving upwards and downwards. The prototype system had poor recognition of several gestures, especially, *telescope out* and *telescope in*. We believe that conforming to the standard is important. Therefore, during the experiments we asked the participants to perform gestures according to the NORSOK standard. As a result, our recorded data-set is more appropriate for training accurate gesture detection, while the prototype implementation demonstrated limited precision. We did not measure exact detection rate of each gesture. Subjective evaluation showed that the numbers are too low to be considered acceptable. We are currently



(a) Gesture ”Hoist”

(b) Gesture ”Use main hook”

Fig. 6. Gestures visualized by an avatar in the OSC Simulator

TABLE I
RECORDED USER DATA ACCURACY, ACCORDING TO GESTURE DEFINITION BY A CRANE OPERATION EXPERT

	Aux Hook	Boom Down	Boom Up	Hoist	Lower	Main Hook	Stop	Emerg. Stop	Telesc. In	Telesc. Out	Direction
Avg. score:	4.31	3.34	3.10	3.41	2.59	3.59	3.97	3.62	3.17	3.14	4.10
StdDev:	0.81	1.26	1.21	0.95	1.74	1.57	1.18	0.90	1.83	1.81	1.21
Wrong:	0	2	0	0	6	2	1	0	5	5	1

developing an updated version of the system, training it using the recorded data.

One conclusion from our study is that user understanding of the gestures vary a lot and some of them did not perform the gestures accurately enough. This did not come as a surprise, considering that all the participants (with exception of one subject) did not have previous experience with crane gestures.

After the recordings we performed analysis of the recorded data. One member of our team was evaluating every gesture performed by every participant, comparing it to the standard and allowed variation range, as suggested by a crane operation expert. Each recorded gesture was ranked in the scale from 0 to 5, where 0 means ”completely wrong”, and 5 means ”perfect”. The average score, standard deviation and number of recordings considered ”completely wrong” are shown in Table I.

The table shows that gestures *Auxiliary hook*, showing direction (Left/Right) and *Stop* are the most accurate. *Lower* gesture is the least accurate: the average score is 2.59 an only 17 from 30 recordings have satisfactory quality. *Telescope in*, *Telescope out* and *Boom up* also have challenges with accurate performance among participants. Based on this analysis we are currently training the gesture detection system using only the accurate gesture recordings.

C. Qualitative user feedback

After taking the gesture recordings, the participants were asked to fill out a questionnaire to rate three aspects: how clearly they perceived the videos, how accurately they have performed the gestures and how accurately the system detecting the gestures they performed.

As described before, we improved the video quality during the second day of the experiment. It had a clear impact on the user feedback: average video quality score on the second day was 4.83 (out of 5.0) versus 3.63 on the first day.

Participants were very critical towards their performance: the average self-assessment score was only 3.57, meaning that they did not feel very competent and comfortable with what they were doing.

Attitude towards the automated gesture detection system accuracy was surprisingly high: the average score was 3.86 (out of 5), while only one person gave a score of 1.0 and only one person gave score 2.0. However, we have a hypothesis that this high score can be attributed to two facts: participants were students excited by the technology, and they were not asked to perform any real crane operation. If they would have to move a crane with the gesture detection, we would expect significantly more frustration if the system would recognize their gestures wrongly sometimes.

V. CONCLUSIONS AND FUTURE WORK

This article describes a research project in cooperation between NTNU Aalesund Campus and Offshore Simulator Centre AS. We have identified a gesture set used for communication between Able Bodied seaman and crane operator during maritime deck operations. We have designed a gesture recognition system that utilizes Microsoft Kinect sensor for tracking of the user’s movements and detects gestures automatically. The first prototype implementation revealed several inconsistencies in the recognition. However, we have collected a valuable data set of 15 different people performing the gestures. We are currently in the process of developing a new version of the system by improving the gesture recognition through extension of the training data set. The system has showed promising results so far.

Although the Kinect sensor version 2 has increased accuracy, the detection of finger configuration is still far from perfect. In particular, there are challenges in separation of *Boom Up* and *Boom Down* gestures, as well as separation between *Telescope In* and *Telescope Out* (see Figure 2). Both these examples rely on position of the thumb. We could argue if these gestures are reasonable from human perspective as well. Is the crane operator sitting far away from the AB really able to see the thumb orientation? Or should these gestures be improved in the standard itself? Potential future research direction include combination of Kinect-based gesture detection with other natural user interface (NUI) technologies. Smart gloves could be one option, and there are many more.

The project will result in a new training module or complete product for current OSC deck simulator and NTNU Aalesund training module. The deck operation training quality and efficiency will be improved. Furthermore, the product will provide more possibility for training evaluation and analysis.

Acknowledgment

The project is fully supported by Mechatronics Lab, Department of Ocean Operations and Civil Engineering, NTNU Aalesund. The authors would like to thank Demy Patrick Gielesen and Malen Akernes Barstad for their contribution to the user study and work on an improved version of the gesture recognition system. Also special thanks are given to Xavier Charmasson for his contribution to the project during his internship.

References

[1] R. Liang and M. Ouhyoung, “A real-time continuous gesture recognition system for sign language,” in *IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 558–567.

[2] P. Hong, M. Turk, and T. Huang, “Gesture modeling and recognition using finite state machines,” *IEEE Conference on Automatic Face and Gesture Recognition*, no. March, pp. 410–415, 2000.

[3] D. Wilson and A. Wilson, “Gesture recognition using the xwand,” Assistive Intelligent Environments Group,Robotics Institute, Carnegie Mellon University, Tech. Rep., 2004.

[4] B. Bedregal, A. Costa, and G. Dimuro, “Fuzzy rule-based hand gesture recognition,” *Artificial Intelligence in Theory and Practice*, pp. 285–294, 2006.

[5] T. Stiefmeier, D. Roggen, and G. Troster, “Gestures are strings: efficient online gesture spotting and classification using string matching,” in *Proceedings of the Second International Conference on Body Area Networks BodyNets*. ICST, 2007.

[6] Standards Norway, “NORSOK STANDARD R-003: Safe use of lifting equipment, Rev. 2,” https://www.standard.no/en/sectors/energi-og-klima/petroleum/norsok-standard-categories/r-lifting-equipment/r-0031/, 2004, [Online]. Last visited 2017-04-06.

[7] US Department of Labour Mine Safety and Health Administration, “Hand Signals for Lifting Equipment,” https://arlweb.msha.gov/Accident_Prevention/Tips/HandSignals.pdf, 2003, [Online]. Last visited 2017-04-06.

[8] Microsoft, “Kinect for Windows,” http://www.microsoft.com/en-us/kinectforwindows/, 2017, [Online]. Last visited 2017-04-06.

[9] Sony, “PlayStation Move,” https://www.playstation.com/en-us/explore/accessories/playstation-move/, 2017, [Online]. Last visited 2017-04-06.

[10] WorldViz, “PPT - Precision Position Tracker,” http://www.worldviz.com/products/ppt, 2017, [Online]. Last visited 2017-04-06.

[11] CyberGlove Systems, “CyberGlove III,” http://www.cyberglovesystems.com/cyberglove-iii, 2017, [Online]. Last visited 2017-04-06.

[12] Xsens, “MVN - Inertial Motion Capture,” http://www.xsens.com/en/general/mvn, 2017, [Online]. Last visited 2017-04-06.

[13] Y. Song, D. Demirdjian, and R. Davis, “Tracking body and hands for gesture recognition: Natops aircraft handling signals database,” in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE, 2011, pp. 500–506.

[14] ——, “NATOPS Aircraft Handling Signals Database,” http://groups.csail.mit.edu/mug/natops/, 2011, [Online]. Last visited 2013-12-06.

[15] I. Guyon and V. Athitsos, “Chalearn gesture challenge: Design and first results,” in *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012, pp. 1–6.

[16] Microsoftt, “JointType Enumeration,” https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx, 2017, [Online]. Last visited 2017-03-23.

[17] Occipital Inc, “Open NI 2,” https://structure.io/openni, 2017, [Online]. Last visited 2017-03-23.

[18] jni4net contributors, “jni4net - bridge between Java and .NET,” http://jni4net.com/, 2017, [Online]. Last visited 2017-03-23.

[19] Microsoftt, “Natural User Interface Tools,” https://msdn.microsoft.com/en-us/library/dn799270.aspx, 2017, [Online]. Last visited 2017-03-23.

[20] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, “Real-time human pose recognition in parts from single depth images,” *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.