# KINECT-BASED SYSTEMS FOR MARITIME OPERATION SIMULATORS?

Girts Strazdins[1], Sashidharan Komandur[1], and Arne Styve[2]
[1] Maritime Human Factors Lab, Aalesund University College, Larsgardsvegen 2, 6009 Aalesund, Norway, Email: gist@hials.no, sash.kom@hials.no
[2] Offshore Simulator Centre AS, Borgundvegen 340, 6009 Aalesund, Norway, Email: asty@offsim.no

## KEYWORDS

Kinect, Natural User Interface, Maritime Operations, Simulator

## ABSTRACT

Maritime operations involve complex tasks that must be taught through practical training. Using modern technologies offshore simulators can serve as realistic environment for training. The degree of immersiveness can be increased by replacing conventional input devices (keyboards, joysticks) with natural user interface (NUI) devices. This paper surveys different NUI technologies and advocates that vision-based systems, such as Microsoft Kinect, can provide gesture recognition with high fidelity. The unobtrusiveness of vision-based technologies is highly important to avoid user resistance and preserve realism of the simulation.

## INTRODUCTION

Maritime field is an important part of industry, providing global supply of fish and oil. To increase safety and efficiency in offshore operations it is critical to provide training for the personnel. Modern technology offers wide range of interactive training options, and simulators are an important part of it. A significant objective of simulator designers is to create truly immersive simulation environment and realistic experience for the users, that must threat the simulation as real work instead of a simple game. One direction of realism improvement is to replace conventional keyboards and joysticks with natural user input (NUI) technologies, such as natural gesture, posture and movement recognition.

On the other hand, maritime personnel is a rather skeptic professional community, preferring well-proven methods, and criticizes new technologies thoroughly. Therefore it is important to select NUI technologies that offer advanced features while not disturbing natural environment and not requiring users to learn completely new concepts. New technology introduction in maritime field is an incremental process, an evolution, not revolution.

This paper reviews NUI technologies for offshore operation simulators intended for maritime personnel training. The main stress is on actual applicability of technologies: availability - existence of commercial hardware and software components (instead of pilot stage research studies); and unobtrusiveness - to ensure acceptance among personnel.

Proceedings 27th European Conference on Modelling and Simulation ©ECMS Webjørn Rekdalsbakken, Robin T. Bye, Houxiang Zhang (Editors)
ISBN: 978-0-9564944-6-7 / ISBN: 978-0-9564944-7-4 (CD)

## NUI TECHNOLOGY OVERVIEW

Natural user interface is a term used to describe technologies for human-machine interaction that are:

- fully invisible
- based on natural and widely used gestures and notions, not requiring users to learn new paradigms and concepts

Invisibility is the ideal goal of NUI, that is pursued incrementally by technological advancement. Integration of natural postures, gestures, movements and concepts into user interface assures faster technology adaptation and less resistance of the users. It allows control of machines to user groups that are neither experts in IT nor willing to become ones.

In the maritime simulator context NUI technological solutions can be divided into three classes: vision based, wearable and tangible systems.

### *Vision based NUI systems*

Vision based NUI use external camera for graphical scene record and analysis. The camera may capture light in different spectra, including visible and infrared range. Active systems may be used, where specific patterns of light (or other type of electromagnetic waves) are transmitted into the scene and then the reflected waves are analyzed. A system is called passive, if it only records image without transmitting any waves into the scene.

Vision based NUI system advantages:

- The solution is completely unobtrusive for the users. No need to wear or hold any items, no wires, clear space of interaction.
- Users can perform the gestures exactly as they would in real life, not requiring to mimic scaled gestures (for example, performing a hand gesture with a finger). From the perspective of motion, this approach is the most "natural".
- Installation is a one time activity, if the simulation space is static. Some calibration may be required for each user, yet it can be automated.
- It is easy to switch between users interacting with the system. No need to change clothes or change on-body equipment, to test sensor alignment.
- The acquired data stream contains a lot of information - usually consisting of color and depth images, sometimes also sound.
- The intelligence of processing is offloaded to a computer, that can perform complex image processing and storage operations.
- Large existing library, framework and example code base

can be reused and adapted for the solution. Many processing, simulation and analysis tools can be used.

Vision based NUI system drawbacks:

- There is no tactile feedback for the user, no feeling of boundaries. Visual feedback must be present for the user to understand, what part of the human body can be seen by the cameras, what is recognized, and what are the boundaries of gestures.
- Vision based systems have higher price, compared to inertial or tangible systems, as there are higher requirements for sensor quality and the included know-how has high cost. However, there are consumer level vision based NUI devices, such as Microsoft Kinect [Microsoft, 2013] or PlayStation Move [Sony, 2013] with price less than $300.
- There are lighting and clothing constraints for the systems to detect persons in the scene correctly. Mirrors or other light sources may interfere with NUI systems, and people may not be detected correctly, when they wear clothes hiding natural human silhouette.
- The interaction space is limited, depending on camera resolution. There are higher-end systems with interaction space up to 50x50 meters [WorldViz, 2013], yet consumer-grade devices have distance of interaction less than 5 meters.
- The system may not be able to detect face and body in some strange positions, for example, a person crouching.
- Image processing requires high performance computer, it is not suited for energy-efficient embedded systems.
- If acquired data is recorded for offline processing, it requires large amounts of storage space, compared to other raw signals, such as accelerometer or gyroscope data used in inertial wearable systems.

The main advantage of vision based systems is complete unobtrusiveness of the system. It allows the persons to act, as if there was no system installed. The users can move freely in the simulation space, and are not frightened by wearable devices. This aspect makes vision based systems superior compared to wearable and tangible technologies. Tangible interfaces might be a good addition to vision based and in some simulation contexts they might be preferred, for example realistic simulation of box lifting or door opening. But for gesture recognition in offshore simulators vision based systems are recommended due to their higher acceptance among simulator users. None of the vision based system constraints is critical for maritime simulator environment.

### *Wearable NUI systems*

NUI systems are called wearable if some of the components are attached to human body. Usually wearable systems consist of multiple nodes with inertial measurement unit (IMU). These are attached to various parts of human body and sense relative movement and orientation in space of these points. More sophisticated wearable nodes may contain video capture and processing. In addition to sensing, wearable nodes can have actuators that give tactile or audio feedback to the wearer.

Examples of wearable systems include:

- Xsens MVN [Xsens, 2013] - full body, camera-less inertial motion capture solution. Consists of 17 inertial sensors, with 6 degrees of freedom, attached to users body by a lycra suit or straps. Sensors have wired connection to a central hub node that transmits data to computer wirelessly. Solution contains software toolchain for body position capture in 3D and export to popular 3D animation software, such as 3DSMax and Maya.
- CyberGlove products [CyberGlove Systems, 2013] for hand, finger and arm tracking and interaction with virtual reality. Bending, rotation and force sensors are used. CyberForce products also provide tactile feedback.

In addition to commercially available products, pilot research results have been shown, including hand tracking wristband by Microsoft research [Kim et al., 2012], that uses infrared laser, camera and IMU. Gloves and other forms of wearable devices, such as Peregrine wired glove [Peregrine, 2010] and Genius Ring mouse [Genius, 2011] are also popular as replacements for conventional keyboard and mouse (not actually NUI devices).

Wearable NUI system advantages:

- Wearable systems usually have lower price, compared to vision based systems, as the hardware is simpler, including sensors. However, if multiple nodes are required to track the whole human body, the price may increase.
- Inertial sensors may achieve higher accuracy, compared to vision based systems, especially for hand and finger tracking.
- Sensor nodes can be implemented, using low-power microcontrollers and sensors, therefore no large batteries or mains power is required.
- The installation requirements are transferred from environment to users - wearable systems can function in virtually any environment without re-installation.
- No strict requirements for the environment: indoors, outdoors, virtually any environment is acceptable. An exception: magnetometer sensors may not function correctly in presence of large metallic objects or strong magnetic fields. No strict requirement for user clothing and accessories.
- In addition to sensing, feedback can be given to users, including tactile signals.

Wearable NUI system drawbacks:

- More complex and time consuming to change system users.
- May disturb the activities of users and introduce psychological resistance against the technology.
- Nodes use either wires that are cumbersome, or wireless transmission that limits data transmission bandwidth.
- In most cases only a part of the body is tracked, for example, human hands.
- Devices that must be moved and changed from person to person tend to brake more often and have a shorter life cycle.
- Wearable devices may have problems with wide variance of user height, weight and girt.

### *Tangible NUI systems*

NUI system is called tangible if users interact with digital environment using physical objects.

One tangible NUI examples is Nintento Wii Remote (Wimote) - the controller for Nintendo Wii gaming console [Nintendo, 2013]. It has a 3-axis accelerometer and an optical sensor, used to detect device location relative to the consoles Sensor Bar, which is places above or below TV and transmits infrared light in predefined pattern. Wii Remote can be used as pointing device as well as a tool for hand gesture detection (without finger separation). An additional module, called Wii MotionPlus, can be plugged in the Wii Remote to increase tracking accuracy by adding gyroscope sensors.

Tangible NUI system advantages:

- Physical objects are not worn or attached to human body, rather simply touched, grasped and carried. This leads to simpler setup and switching between users.
- Interaction between objects is more natural and intuitive.
- More realistic sensory feedback, compared to vision based systems.

The main drawback of tangible NUI systems: basically interaction with hands only, no ability to capture position of the whole skeleton.

### *Hybrid systems*

Vision can be merged with wearable or tangible approaches to increase accuracy. Examples:

- PlayStation Eye Move [Sony, 2013] mixes vision technology by PlayStation Eye camera and wand with inertial sensors (3-axis accelerometer, 3-axis angular rate sensor, magnetometer) and a sphere illuminated with visible light of specific spectra acting as a marker tracked by PlayStation Eye camera.
- Microsoft wearable hand tracking wristband [Kim et al., 2012], that uses infrared laser and camera, worn around persons wrist to detect finger gestures.
- A prototype glove, uses specific color segmentation and video processing for hand tracking [Wang and Popović, 2009].
- WorldViz PPT - long range, high quality cameras, used to recognize and track specific markers, attached to objects [WorldViz, 2013].

Hybrid systems may combine the advantages of multiple technologies. However, hybrid approach may also have a mixture of constraints imposed by all the used technologies. For example, wearable and vision based approach combination may increase accuracy, while requiring users to wear some objects and setting limitations on room lighting and person clothing.

## MICROSOFT KINECT

Kinect is a motion sensing device by Microsoft, that was released on November 2010 for the XBox 360 video game console. It is a vision-based motion and gesture recognition system that uses infrared active scanning technology to detect distance to players and objects in the field of view. The technology was widely accepted in the gamer community and Kinect sensor holds the Guiness World Record of being the “fastest selling gaming peripheral” - 8 million units sold in the first 60 days [Guinness World Records, 2011]. Shortly after the release of Kinect for XBox, open source community created a Linux driver for the device, and *hacking Kinect* became popular [Xia et al., 2011], [Frati and Prattichizzo, 2011] . Microsoft realized the potential of the technology beyond gaming and released official development kit in February 2011, containing a software development kit (SDK) and also a sensor device, called “Kinect for Windows”, optimized to be used with PC computers and Windows 7 environment. Numerous interesting Kinect applications have been demonstrated, including virtual dressing room [Zhou et al., 2012], room scanner [Manctl, 2013] and virtual furniture fitting [NCONNEX, 2012].

### *Technolgy and hardware*

Kinect sensor provides the following raw data:

- Depth image stream, resolution 640x480 pixels, 30 frames/sec. Distances from 1.2m up to 3.5m are suggested, while theoretically distances from 0.8m to 4.0m are available. Near mode tracking is also available, where person tracking in distances from 0.4m is available (different tracking technology is used, with lower accuracy). Depth map is calculated using active infrared scanning. Infrared light is transmitted in a predefined pattern in the space and time of flight of light waves is calculated by infrared camera.
- Color image stream, resolution 680x480 at 30 frames/sec or 1280x960 pixel resolution at 12 frames/sec. Different image formats are available: RGB, YUV or Bayer. Color camera sensor is used to capture image stream.
- Audio stream, captured by a four-element linear microphone array.
- Sensor orientation relative to the ground, captured from 3D accelerometer sensor.

In addition, Kinect tilt can be detected and adjusted in 54 degree range by sending commands to the built-in motor.

### *Software and preprocessed data*

Microsoft Kinect is not only accurately selected hardware components. The true power lies in the SDK containing data preprocessing algorithms that incorporate knowledge and results of extensive research work [Shotton et al., 2013].

The official Kinect SDK by Microsoft allows development of Kinect-enabled applications for Windows, using Visual Studio and .Net environment, in either C#, C++ or Visual Basic. The SDK contains simple yet rich API. In addition, Microsoft provides well organized documentation and developer toolkit, containing extensible application examples. Drivers for the Kinect sensor are also included.

The Kinect SDK can detect up to 6 people from the depth images, and skeleton joint positions can be tracked for up to 2 from them. All the image processing is done by the Kinect SDK. Joint positions in 3D space relative to Kinect sensor position are exposed.

The SDK can record audio signal from the microphone array. In addition, it can detect direction of the audio source and filter background noise.

Each new version of the SDK adds new feature support, such as relative bone rotations, seated mode (only upper part of human body visible), improved image quality and performance.

Additional features can be used by bridging Kinect with other Microsoft SDKs:

- Speech recognition by using Microsoft SDK and Kinect for Windows Language packs. A total of 12 languages are supported at the moment (Language pack v11.0).
- Face recognition and tracking by using Microsoft Face Tracking SDK (included in Kinect SDK since v1.5)

***Third party drivers and SDKs***

In addition to official Microsoft SDK, third party drivers and software libraries have been developed by open source communities:

• OpenKinect [OpenKinect, 2012] - open source Kinect libraries, C language. Provides only basic access to raw data streams, no player detection or skeleton tracking available. Its usability for gesture recognition is limited, as player joint tracking implementation is a rather complex task. OpenKinect can be used in computer vision research projects, where custom detection and tracking algorithms must be evaluated.

• OpenNI framework - an open source SDK used for the development of 3D sensing middleware libraries and applications [PrimeSense, 2013c], originally in C++ language, yet wrappers to other languages are available, including Java and Python. The community promotes interoperability between different NUI devices by creating multi-layer framework, that abstracts hardware and provides unified interface for higher level middleware, such as PrimeSense NITE [PrimeSense, 2013b] - a computer vision framework, implementing skeleton tracking, hand tracking and gesture detection algorithms.

In contrast to Microsoft SDK, these open source solutions are designed to be platform independent: Windows, Linux, MacOS. OpenNI is even device-independent. OpenKinect and OpenNI have open source (NITE middleware is not), however, the generality of OpenNI framework in first versions (v1.x) makes its source code and documentation hard to understand. New version of OpenNI (v2.x) has been recently released (December 18th, 2012). It has been rewritten from scratch, with simplified API. While it lacks some of the later features (no Java and Python wrappers are available at the moment), the OpenNI seems to be a promising alternative in future.

Both open source solutions support Kinect for XBox and are having issues with support of Kinect for Windows device. There are bridging solutions available which use official Microsoft Kinect drivers and SDK, and implementing a hardware abstraction layer for OpenNI [Washio, 2012].

Author experience shows, that the existing versions of open source solutions are unstable for Kinect for Windows device. The situation may change in coming years, if the communities put enough effort to develop the OpenNI framework, yet at the moment official Microsoft SDK is significantly more stable, easier to use and understand and should be preferred for NUI applications.

***Kinect in Java***

One of the most popular programming languages is Java. However, official SDK supports only .Net environment and development in Microsoft Visual Studio. The most convenient approach for Kinect programming in Microsoft SDK is to use C# language, as it has a high-level API with simple workflow. However, wrapping C# to Java is not directly possible.

Multiple alternatives do exist to provide access to Kinect for Java programmers:

• Manually create a two-layer C# to Java wrapper, using procedural C as an intermediate layer. Java Native Access (JNA) [JNA, 2013] or Java Native Interface (JNI) [Oracle, 2011] can be used for wrapping. Official SDK also provides C++ interface. Wrapping is a bit easier in this case, yet it adds complexity of handling garbage collection manually.

• Use automated C# to Java wrapping solutions, such as JNBridge Pro [JNBridge LLC, 2013]. Java interface is created from a .Net DLL automatically. Yet some specifics of data passing between .Net and Java have to be considered to get maximum performance. And some of the tools may require purchase of a license.

• Create your own forwarder of Kinect data streams, approach used in JNect - an Eclipse plugin for Kinect [Aumann, 2012].

• Java bindings are supported in OpenNI v1.x framework, and can be used, if the OpenNI framework is suitable for a particular application. OpenNI version 2.0 does not have Java bindings, yet the implementation should come in near future.

***Advantages***

There are several advantages of using Kinect as NUI technology in maritime operation simulators:

• Unobtrusiveness. This is the most important advantage: the technology is fully unobtrusive from user perspective - no additional devices, joysticks, sensors, or markers are required to be held or worn by the users to be recognized.

• Accuracy. Research studies show, Kinect provides accuracy with error below 4cm [Khoshelham and Elberink, 2012]. While there may be difficulties of distinguishing finger gestures in larger distances, this accuracy is more than satisfactory for skeleton tracking needs. In addition, Kinect does not have error accumulation effect that is present for inertial systems.

• High value know-how. Extensive image processing research work has been done to create raw data preprocessing in the SDK and provide high level skeleton tracking API [Kohli and Shotton, 2013].

• Simplicity. Easy to use API is provided, supplemented by rich example set and documentation.

• Rich sensor set. The device contains sensors and provides data streams that can be useful in a variety of scenarios: depth map, color video stream, audio capture, gravity and tilt sensing.

• Integration. Kinect SDK integrates with speech recognition and face detection SDKs by Microsoft.

***Constraints***

To use Kinect technology, developers must take the following constraints into account:

• Clear space is required, where players are distinguishable from the room.

• While Kinect is able to work in full darkness, direct sunlight or other bright lightning conditions are not desired.

• Player clothing should not be reflective or hiding the human silhouette. Specialized costumes, such as used by clowns or cheerleaders may interfere with skeleton tracking.

• The range or operation is limited to a small room - distance range 0.4-3.5 meters, and only a couple of meters in horizontal and vertical directions, depending on the distance to Kinect.

• Kinect may have problems in environments, where other infrared light sources are present. The same applies to using multiple Kinect devices simultaneously.
• The official Microsoft SDK supports Visual Studio and .Net on Windows only. Open source alternatives, with their advantages and drawbacks, were mentioned above.

***Unintended gesture avoidance***

One issue to solve for Kinect application in maritime operation simulators is false gesture rejection. The operator performs series of other actions during training in addition to hand signals. In his book David Catuhe suggests to use the following methods to separate between hand signal and ambient motion contexts [Catuhe, 2012]:
• The user is ready to give hand signals when the body is in a stable position. It can be detected by comparing recent positions of one or several joints of the skeleton, for example, hip center.
• The user should be facing the sensor. It can be detected by comparing distance of both shoulders to Kinect or detecting orientation of the head, using Microsoft Face Tracking SDK (included in Kinect SDK since v1.5)

## KINECT EXTENSIONS

Kinect technology can be extended to get improved accuracy, additional features or better user experience. Examples of extensions include:
• Using multiple Kinect sensors to track human body in full 360 degree range. While a single Kinect is only able to analyze the surface in front of the sensor, multiple devices can be places in different angles and may capture the whole body, even when the person has turned back or side to the first Kinect. Examples include iPi Soft[iPi Soft, 2013] - a software to capture 3D video, using two Microsoft Kinect, Sony PlayStation Eye [Sony, 2013] or Asus Xtion Pro [ASUS, 2013] camera sensors.
• Omni direction treadmill floor - to allow movement of the character while keeping the person in place [Inition, 2013].
• Microsoft Kinect official SDK provides skeleton tracking, without hand and finger separation. Hand tracking can be added by using third party software, such as NITE middleware [PrimeSense, 2013b] or SigmaNIL framework [SigmaNIL, 2012]. In short range interfacing, devices, such as Leap Motion [Leap Motion, 2013], can be used for finger separation in gestures.

## KINECT ALTERNATIVES

Kinect is only one of 3D user tracking devices. It includes hardware developed by PrimeSense company. The same hardware is included in Asus Xtion Pro Live [ASUS, 2013] and PrimeSense Carmine [PrimeSense, 2013a] 3D depth sensors. All the devices with PrimeSense hardware are supported by OpenNI open-source NUI framework [PrimeSense, 2013c] and NITE middleware [PrimeSense, 2013b]. On one hand, OpenNI offers platform-independency and source open for extension. On the other hand, as experience of our research group shows, OpenNI is unstable for *Kinect for Windows* sensor device. Unstable operation was encountered on Windows 7, as well as on Mac OSX 10.7 Lion operating systems. Although it may change in near future, the existing version of official Microsoft SDK (v1.6, October 2012) is more stable compared to current OpenNI framework versions (v2.0 and v1.5.4.0).

A 3D user tracking device, without PrimeSense hardware, is provided by Panasonic: D-Imager (EKL3105) [Panasonic, 2012]. Panasonic promises ranges up to 9m, 4cm accuracy, and provides software for depth map extraction only, no skeletal tracking. Theoretically it can be used with OpenNI, a third party developer driver with unknown state is available [danielpq, 2012].

Assuming the stability of drivers and skeleton tracking routines, authors suggest to choose Kinect for Windows device with official Microsoft SDK. Microsoft Research team at Cambridge has made a great effort to incorporate accurate and fast state-of-art pose detection algorithms in the Kinect sensor [Shotton et al., 2013]. Alternative solutions may become more rich and stable in the future, but are incomplete at the moment.

## DISCUSSION

Maritime operation training is a simulator use case, that poses specific requirements on the technology and environment. The trainees are adults from the maritime field. As opposed to teenage or preschool children, these are mature people that prefer training atmosphere to maintain a certain level of seriousness. While some entertainment and gaming aspects may seem attractive in public demonstrations, is not particularly welcome in real training sessions. Advanced technologies, that might increase engagement in youth and video game fans in particular, may interfere with trainee experience and decrease the degree of immersiveness. Thus, the technologies must be selected carefully to ensure maximum realism, that lets the users perceive the simulation as a reality and feel the same level of responsibility while performing the operations.

In 2005, Offshore Simulator Centre was performing pilot studies on using a virtual reality helmet (head-mounted display) in the training simulation. The technology was discarded for two basic reasons: the quality of visuals was too low to provide realistic interface; and the users did not accept devices that interfere with their natural movements.

While the former argument might be outdated today, the latter still applies: adult simulator users, and maritime operation experts in particular, are resistant to wearable technologies. Traditional tools are preferred as much as possible, and any *high-tech* device makes the maritime experts feel as fictional movie characters, and it certainly is against the basic principle of simulators - to provide realistic experience.

Existing offshore operation simulator confirm the preference of using vision based NUI technologies:
• Kongsberg Offshore Vessel Simulator for Seismic Streamer Operations Training - uses Kinect for avatar movement control [Kongsberg Maritime, 2012].
• Maersk Offshore Helicopter Landing Officer (HLO) training simulator incorporates experimental use of Kinect for item selection in the graphical user interface and gesture recognition for helicopter landing operations as part of offshore operations [Maersk Training, 2011].

To conclude the discussion - the unobtrusiveness of NUI technologies is very important to provide realistic simulation environment that is treated as a serious real-life experience by the users in operation.

## FUTURE WORK

Our existing research is focused on incorporating NUI technologies for more natural and immersive user experience in maritime operation simulators. A gesture recognition framework is being developed, targeted primarily for crane operator training. Objective gesture recognition accuracy evaluation and subjective training experience improvement assessment is planned as part of future work.

## CONCLUSION

This paper analyzes natural user interface technology applicability to maritime operation simulators. The Microsoft Kinect vision-based 3D user skeleton tracking system is suggested as optimal commercial solution currently available. The preference is substantiated by providing arguments from maritime training experts who advocate on the superiority of fully unobtrusive technologies that are required to maintain the realism of immersive simulations.

## ACKNOWLEDGMENT

The authors would like to thank Qi Xu for contribution to NUI device survey, Robert Rylander and Hans Petter Hildre for providing feedback and access to expert knowledge in maritime operation training.

This study is supported by Norwegian Research Council (NFR) Grant in Marine Technology, Nr. 210797.(2011-2013).

## REFERENCES

[ASUS, 2013] ASUS (2013). Xtion PRO LIVE. Available from: http://www.asus.com/Multimedia/Xtion_PRO_LIVE [cited 2013-04-05].

[Aumann, 2012] Aumann, A. (2012). jnect: An Eclipse Plug-In providing a Java Adapter for the Microsoft Kinect SDK. Available from: http://code.google.com/a/eclipselabs.org/p/jnect/ [cited 2013-04-05].

[Catuhe, 2012] Catuhe, D. (2012). *Programming with the Kinect for Windows Software Development Kit*. Microsoft Press.

[CyberGlove Systems, 2013] CyberGlove Systems (2013). CyberGlove III. Available from: http://www.cyberglovesystems.com/products/cyberglove-III/overview [cited 2013-04-05].

[danielpq, 2012] danielpq (2012). OpenNI Module to TOF Panasonic DImager camera. Available from: http://opennidimagermodule.codeplex.com/ [cited 2013-04-05].

[Frati and Prattichizzo, 2011] Frati, V. and Prattichizzo, D. (2011). Using kinect for hand tracking and rendering in wearable haptics. In *IEEE World Haptics Conference (WHC)*, pages 317–321. IEEE.

[Genius, 2011] Genius (2011). Ring mouse 2. Available from: http://www.geniusnet.com/wSite/ct?xItem=56975ctNode=3619mp=1 [cited 2013-04-05].

[Guinness World Records, 2011] Guinness World Records (2011). Kinect for XBox 360: Fastest-Selling Gaming Peripheral. Available from: http://www.guinnessworldrecords.com/world-records/9000/fastest-selling-gaming-peripheral [cited 2013-04-05].

[Inition, 2013] Inition (2013). MSE Omni-directional Floor. Available from: http://inition.com/3D-Technologies/mse-omni-directional-floor [cited 2013-04-05].

[iPi Soft, 2013] iPi Soft (2013). Motion Capture for The Masses. Available from: http://www.ipisoft.com/ [cited 2013-04-05].

[JNA, 2013] JNA (2013). Java Native Access (JNA). Available from: https://github.com/twall/jna [cited 2013-04-05].

[JNBridge LLC, 2013] JNBridge LLC (2013). JNBridge Pro. Available from: http://www.jnbridge.com/ [cited 2013-04-05].

[Khoshelham and Elberink, 2012] Khoshelham, K. and Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454. Available from: http://www.mdpi.com/1424-8220/12/2/1437 [cited 2013-04-05].

[Kim et al., 2012] Kim, D., Hilliges, O., Izadi, S., Butler, A., Chen, J., Oikonomidis, I., and Olivier, P. (2012). Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 167–176. ACM.

[Kohli and Shotton, 2013] Kohli, P. and Shotton, J. (2013). Key developments in human pose estimation for kinect. In Fossati, A., Gall, J., Grabner, H., Ren, X., and Konolige, K., editors, *Consumer Depth Cameras for Computer Vision*, Advances in Computer Vision and Pattern Recognition, pages 63–70. Springer London.

[Kongsberg Maritime, 2012] Kongsberg Maritime (2012). Kinect for Windows integrated with KONGSBERG Offshore Vessel Simulator for Seismic Streamer Operations Training. Available from: http://goo.gl/bjYdp [cited 2013-04-05].

[Leap Motion, 2013] Leap Motion (2013). Individual Hand And Finger Gesture Detection Device. Available from: https://www.leapmotion.com [cited 2013-04-05].

[Maersk Training, 2011] Maersk Training (2011). User Guide HLO Simulator. Available from: http://offshoresimulator.com/hlo_guide.pdf [cited 2013-04-05].

[Manctl, 2013] Manctl (2013). Skanect: 3D Scaning Made Easy. Available from: http://skanect.manctl.com/ [cited 2013-04-05].

[Microsoft, 2013] Microsoft (2013). Kinect for Windows. Available from: http://www.microsoft.com/en-us/kinectforwindows/ [cited 2013-04-05].

[NCONNEX, 2012] NCONNEX (2012). Room Designer. Available from: http://nconnex.com/portfolio/room-designer/ [cited 2013-04-05].

[Nintendo, 2013] Nintendo (2013). Wii gaming console. Available from: http://www.nintendo.com/wii [cited 2013-04-05].

[OpenKinect, 2012] OpenKinect (2012). OpenKinect Project. Available from: http://openkinect.org [cited 2013-04-05].

[Oracle, 2011] Oracle (2011). Java Native Interface (JNI). Available from: http://docs.oracle.com/javase/6/docs/technotes/guides/jni/ [cited 2013-04-05].

[Panasonic, 2012] Panasonic (2012). D-Imager EKL 3105. Available from: http://www2.panasonic.biz/es/densetsu/device/3DImageSensor/en/ [cited 2013-04-05].

[Peregrine, 2010] Peregrine (2010). Gaming glove. Available from: http://www.theperegrine.com/ [cited 2013-04-05].

[PrimeSense, 2013a] PrimeSense (2013a). Carmine 3d sensor. Available from: http://www.primesense.com/solutions/sensor/ [cited 2013-04-05].

[PrimeSense, 2013b] PrimeSense (2013b). NITE Middleware. Available from: http://www.primesense.com/solutions/nite-middleware/ [cited 2013-04-05].

[PrimeSense, 2013c] PrimeSense (2013c). OpenNI: The standard framework for 3D sensing. Available from: http://www.openni.org/ [cited 2013-04-05].

[Shotton et al., 2013] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Commun. ACM*, 56(1):116–124. Available from: http://doi.acm.org/10.1145/2398356.2398381 [cited 2013-04-05].

[SigmaNIL, 2012] SigmaNIL (2012). Vision Framework for Natural User Interfaces. Available from: http://www.sigmanil.com/ [cited 2013-04-05].

[Sony, 2013] Sony (2013). PlayStation Move. Available from: http://us.playstation.com/ps3/playstation-move/ [cited 2013-04-05].

[Wang and Popović, 2009] Wang, R. and Popović, J. (2009). Real-time hand-tracking with a color glove. In *ACM Transactions on Graphics (TOG)*, volume 28, page 63. ACM.

[Washio, 2012] Washio, T. (2012). kinect-mssdk-openni-bridge. Available from: http://code.google.com/p/kinect-mssdk-openni-bridge/ [cited 2013-04-05].

[WorldViz, 2013] WorldViz (2013). PPT - Precision Position Tracker. Available from: http://www.worldviz.com/products/ppt [cited 2013-04-05].

[Xia et al., 2011] Xia, L., Chen, C., and Aggarwal, J. (2011). Human detection using depth information by kinect. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 15–22. IEEE.

[Xsens, 2013] Xsens (2013). MVN - Inertial Motion Capture. Available from: http://www.xsens.com/en/general/mvn [cited 2013-04-05].

[Zhou et al., 2012] Zhou, Z., Shu, B., Zhuo, S., Deng, X., Tan, P., and Lin, S. (2012). Image-based clothes animation for virtual fitting. In *SIGGRAPH Asia 2012 Technical Briefs*, page 33. ACM.

## AUTHOR BIOGRAPHIES

**GIRTS STRAZDINS** obtained his masters degree in computer science from University of Latvia in 2008. He worked as a wireless sensor network researcher at the Institute of Electronics and Computer Science (EDI), Riga, Latvia, from 2010 to 2012, when he moved to Aalesund and joined Maritime Human Factors Lab at the Aalesund University College. He is currently working in research related to natural user interface and gesture incorporation in offshore simulators.

**SASHIDHARAN KOMANDUR** was born in India, obtained his bachelor's degree in naval architecture & ocean engineering from IIT-Madras, masters degree in ocean engineering from University of California Berkeley and Ph.D. in industrial engineering from the University of Washington-Seattle. He is leading a research group in field of human factors in marine operations at Aalesund University College. He is also the Global maritime knowledge hub (GMKH-Norway) chair for Human factors in marine operations. His main work revolves around understanding human factor issues that exist in the ship bridge environment of modern ships.

**ARNE STYVE** was born in Bergen, Norway and went to the University of Newcastle upon Tyne where he studied Microelectronics and Software Engineering and obtained his B.Eng w/Hons degree in 1991. He worked for a few years in Siemens Defense Industries (later Thales) in Oslo, where he was project manager and principal engineer developing solutions for the Norwegian Army. In 1999 he moved to Aalesund and worked as an Assistant Professor at Aalesund University College teaching Object Oriented SW Development. In 2005 he joined the project that later became the Offshore Simulator Centre AS (OSC AS), a company developing advanced offshore simulators for team based training of critical offshore operations. He is currently the manager of the R&D department.