Controlling virtual ROVs using gestures

Stian Surén
Department of Product Design
Norwegian University of Science and Technology

ABSTRACT

Virtual reality and head mounted displays (HMDs) give pilots of ROVs new possibilities and challenges when planning and executing underwater operations. One challenge is that pilots cannot see their physical controls when wearing an HMD. This review article has analyzed current literature to find out if there are enough studies to support the development of a gesture based interface as a viable solution for this challenge.

The main difference of opinion within the academic world of gesture interfaces is whether speech should be included or not. This article proposes an interface without speech to accompany gestures. Also discussed is how such an interface can be learned effectively, how to avoid fatigue, if gestures can control secondary functions as well, and the role of physical devices in relation to virtual reality immersion.

KEYWORDS: Gestures, HCI, Interface, Design, ROV, virtual reality

1. INTRODUCTION

Virtual reality coupled with head-mounted displays (HMD) such as Oculus Rift open up new possibilities for pilots of underwater remotely operated vehicles (ROV). At subsea fields where the landscape and man-made structures are known, virtual representations of the underwater environment can be constructed. Consequently, ROV missions can be planned and executed by controlling an ROV in the virtual reality and then saving the waypoints for an automated mission in the real world. The use of HMDs can replace the clutter of monitors for the pilot and create a more intrinsically understandable environment.

On the other hand, a user of an HMD will not be able to see his own hands or the physical interface normally used to control ROVs. Therefore, the primary goal of this article is to analyze the current state of literature to determine if it gives any basis for the development of a gesture based interface as a viable solution to this challenge.

Gesture based interfaces have been featured in science fiction movies for decades, but only recently have they appeared in real life products. Touch screens have made 2D gesture interfaces an everyday commodity, while the gaming industry with their Xbox Kinect, PlayStation Camera and Nintendo Wii have introduced 3D gestures into our homes.

The common denominator for these products is that the gestures are input to control graphical user interfaces and digital games. They are not output to control physical products or systems. For that we have wheels, joysticks, controllers and so on. However, with the emergence of new and reliable gesture sensing devices, such as MYO [1] and Leap Motion [2], it is not difficult to imagine gesture interfaces controlling advanced vehicles and robots such as cars, planes, submarines and ROVs (Remotely Operated Vehicles).

There is no other research out there that covers this subject in its entirety. Thus, to reach the goal of this article, research has focused on the different parts that make up a gesture interface for virtual ROVs: ROVs, gestures and interfaces. Following the introduction, the article will continue with this structure: Section 2 is dedicated to understanding how ROVs move in 3D space and how they are controlled in real life environments today. In section 3, previous gesture research is summarized and compared to understand the subject of gestures more thoroughly. The article will then focus on how to implement gestures into a usable interface in section 4. The discussion in section 5 will explore how all of this can be combined into a gesture interface for controlling virtual ROVs, and to find areas of uncertainty where more research is needed.

2. CONTROLLING A VIRTUAL ROV

This section aims to provide an overview on how ROVs are controlled today, and thus lay a foundation for the continuing research on adapting the control to a gesture interface.

Unless otherwise cited, the information in this section is gathered through interviews with professors, ROV-pilots and professionals.

2.1 What is an ROV?

An ROV is a Remotely Operated Vehicle, and is commonly a tethered underwater robot controlled from a boat or on land. The majority of ROVs have a floating pack made of syntactic foam that gives the ROV slight positive buoyancy, enabling the vehicle to float slowly to the surface if electrical systems break down. ROVs are normally used to observe underwater operations, explore underwater terrain or to perform certain tasks under water [3].

2.2 The ROV in relation to 3D space

ROVs, as all rigid bodies moving in 3D space, have six degrees of freedom (DOF), three translational and three rotational. The translational components define the trajectory, and the rotational components define the behavior of the ROV along the trajectory [5]. This makes for a pretty complicated system to control [6] (See figure 2).

However, ROVs are designed to only make use of 4 DOF (x, y, z, yaw), while the other two, roll and pitch, are considered intrinsically stable [6]. This implies that the ROV is always horizontally

Figure 1: A typical ROV with an aluminum frame and a floating pack on top [4].

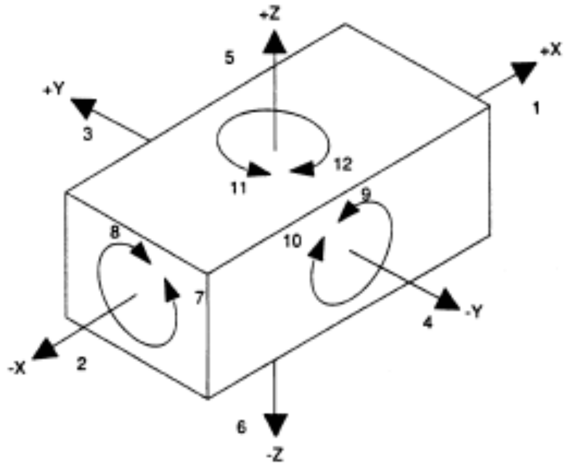


Figure 2: A rigid body showing the six degrees of freedom, and consequently the 12 ways in which it can move [7].

aligned. Thus, the ROV operator has a simpler job of keeping track of the orientation of the vehicle.

Given that an ROV is operating underwater, the vehicle will experience much more friction than vehicles operating on land or in the air. This means the thrusters must be powered continuously to move in any direction. Also, as ROVs are almost neutrally buoyant, gravity will not affect the ROV.

2.3 How real ROVs are controlled today

To control an ROV today, a pilot sits on the surface and communicates with the vehicle using a Surface Control Unit (SCU). Some specific tasks are autonomous, such as position tracking, dynamic positioning, auto-heading and auto-depth control [6]. The Surface Control Unit consists usually of several screens showing a live camera feed from cameras mounted on the ROV itself. The screens are usually not arranged in any particular order according to the orientation on the ROV itself. All maps are also 2D and from a birds eye view, like any other normal map.

The SCU is also accompanied by a user interface consisting of mostly buttons for functions and arms, and two joysticks for controlling the motion of the ROV. One joystick is used to control back/forth and yaw, and the other is for left/right and up/down.

Another way to control arms is by having a master arm in the SCU controlled by a pilot manipulating its movement with a handle. A slave arm on the ROV itself will then mimic the motion of the master arm. Force feedback is given to the pilot so that he or she is sensitive to the force applied to objects.

One issue for ROVs today is speed versus visibility. When navigating in water using cameras, high speeds will mean less visibility due to the debris in the water. This means that the pilot must pay extra attention when driving fast.

Even though the camera feed only has a delay of a couple milliseconds, the position of the ROV in the water is not updated as frequently. Depending on the technology used, the pilot will get position updates every 1 to 5 seconds.

3. GESTURES

Gestures are a part of kinesics, the study of ‘body language’ [8]. They are at the core of human communication and are used everyday as a part of human-human interaction . They are, simply put, a motion of the body that contains information [9]. They can be used to communicate as a tool on it’s own or to give a verbal message more meaning. Symbolic in nature, gestures can never be fully explained in purely kinesic terms and the speaker freely constructs their meaning [10].

Researching articles from different fields such as design, computer science and psychology give a wide set of varying explanations as to what gestures truly are.

Golod et al [11] claim gestures to be a new kind of sign language and that new systems should use the same signs as others to perform similar tasks. Conversely, McNeill [10] argues that the speaker creates gestures at the moment of speaking, and even though two people try to convey the same message, they will create unique gestures.

Cassell [12] only uses “gestures” to specifically address hand gestures accompanied by vocal speech and says: ‘I don’t believe that everyday human users have any more experience with, or natural affinity for, a ‘gestural language’ than they have with DOS commands.’ She believes the fact that gestures are not code, is what really makes it interesting. According to McNeill [10] 90 percent of gestures occur in the context of speech. Also, gestures are processed in the same parts of the brain as spoken language [13].

There are several ways to define and group different types of gestures. Cassell [12] suggest three types of gestures found in human-human interaction:

- Emblematic: Socially dependent gestures that are consciously produced. An example is ‘Thumbs up’.
- Propositional: Conscious, it embodies gestures such as ‘Put that there’.
- Spontaneous: Divided into four subgroups, they embody unconscious gestures used to aid verbal communication. E.g. saying ‘Continue talking’ while giving the hand a rolling motion.

Hummels and Stappers [14] on the other hand call gestures accompanied by speech gesticulations, and pose the question if it is desirable to communicate with our computer in the same way we communicate with people every day. Walker says: ‘when you’re interacting with the computer, you are not conversing with another person. You are exploring another world.’ [14]

In an observation of designers, only 10% wanted to use speech during 3D gestural design. Therefore, Hummels and Stappers [14] give a categorization of gestures that is not based on gesticulations alone, but includes a reference to space, tempo, symbols, emotion, context and task. These categories are space, pathic information, symbols and affect (see figure 3).

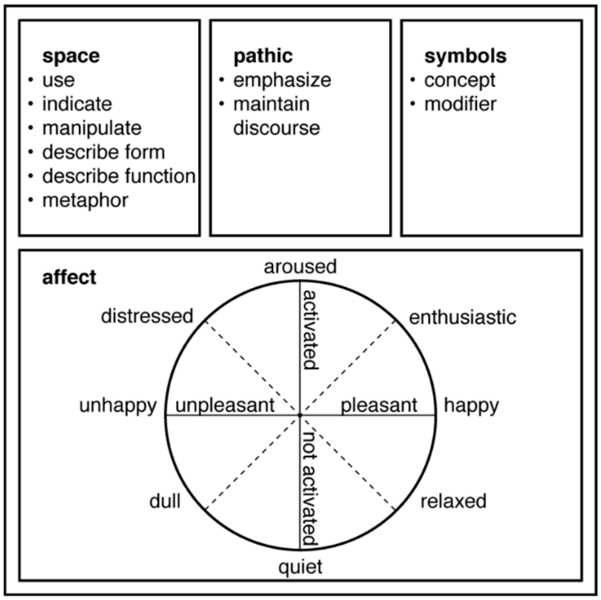


Figure 3: Categorisation of gestures made by Hummels and Stappers [14].

4. USABILITY IN GESTURE INTERFACES

4.1 What is a good gesture interface?

As gestures are getting more common in human-computer interaction, gesture based interfaces are recognized as being very different from the traditional point-and-click GUI [12, 15-17]. Gesture interfaces have had a tendency previously to concentrate on gestures as a language, rather than gestures as a part of a multi-modal communicative event [12]. One exception is ‘Put-That-There’, where the system uses gestures to fill in the blanks when it is unable to fully understand verbal communication [18].

Current gesture interfaces are indeed based on a set of preprogrammed gestures users have to learn, which is hardly intuitive to a normal user [14].

Inspired by Norman’s seven stages of execution with respect to system interaction [19], Bellotti et al. [17] introduce five issues to aid in gesture interface development.

- **Address** – Directing communication to the system
- **Attention** – Establishing that the system is attending
- **Action** – Defining what is to be done with the system
- **Alignment** – Monitoring system response
- **Accident** – Avoid and recover from errors and misunderstandings

This article will only focus on Action and Alignment, assuming that the system is already aware of and attending to the controller of the ROV. In the field of ubiquitous computing, there might be several possible objects to interact with at the same time and the context for the interface is everywhere, meaning that the user also uses gestures for human-human interaction and interaction with physical objects. However, for the interface described in this article, the user has one object to interact with, the ROV, and the interaction is continuous over a longer period of time (see table 1). The address and attention points are therefore not relevant to create a functional gesture interface for virtual ROVs compared to action and alignment.

Also, when controlling an ROV, a wrong turn will only be solved by countering the previous action or by continuing on the offset trajectory. Therefore, the accident point is also not considered in this article.

If gesture interfaces are implemented for virtual reality applications where it is essential that the user feels immersed into another 'world', it is important that the user is able to move freely in 3D space without wearing or handling a physical device such as gloves, trackers, pointing devices, game controllers etc. [9].

Nielsen et al. [16] provide following principles to build a good gesture interface:

1. Avoid outer positions
2. Relax muscles
3. Relaxed neutral position is in the middle between outer positions
4. Avoid repetition
5. Avoid staying in static position
6. Avoid internal and external force on joints that may stop body fluids

In eye-gazing interfaces, there is a phenomenon called the Midas Touch problem [18], or immersion syndrome by Golod et al [11], and it relates to the fact that the eyes, when looking at an interface, are never off. The same principle relates to gesture interfaces as well; the hands are never off in the same way that a mouse can be left in place.

There are several proposed ideas on how to solve this problem. Kendon's definition [20] of gestures say that gestures are merely deliberate movements with sharp onsets and offsets [12]. Golod et al [11] take this a bit further and say that gestures consist of segments and phrases, much like verbal speech. This suggests that it is important to take these different rhythms into account when designing a gesture interface.

An interesting problem that may occur is that even though gestures in 3D control an interface, humans actually perceive 2D elements faster than 3D elements [21]. Therefore, the GUI and other elements on screen should be 2D.

4.2 Effectively learning a gesture interface

This part gives a brief introduction to relevant themes on understanding how a user can improve their intuitive skill as fast as possible when using a gesture interface.

There are two basic conditions for acquiring a skill [22]:

1. An environment that is sufficiently regular to be predictable
2. An opportunity to learn these regularities through prolonged practice

Games, such as football or chess, provide a regular environment, while long-term forecasters in the stock market operate in a zero-validity environment [22].

Feedback is essential when developing any kind of intuitive skill. The quality and speed of feedback is just as important as being given the sufficient opportunity to practice. Cars are an ideal example: when pressing the gas pedal, brake or turn, the pilot will get immediate feedback through the behavior of the vehicle. In the case of breaking too hard, the user will experience slight discomfort [22]. However, the feedback should not interrupt the work being executed as this might interrupt the user's state of flow [23] and will slow down progress.

The reason feedback is important is because humans are programmed to see patterns [21, 22, 24]. If a user only connects the user interfaces with negative feedback, it will make the user discouraged [25]. In the economy of action, effort is a cost and the acquisition of skill is driven by the balance of benefits and costs [22]. Hence, if the costs are large (receiving negative feedback), the user will abandon the task. Weinschenk [23] adds that feedback should be precise and objective. Praise or negative feedback steals attention from what the users needs to do to improve.

In addition, it is important that the feedback is coupled with the correct action. In so-called 'wicked' environments, a user will view the feedback as the result of one specific action, while it is actually the result of something else [22].

Feedforward is therefore interesting as an alternative to, or to complement, feedback [11]. In cognitive science, feedforward is described as 'images of adaptive future behavior, hitherto not mastered' [26]. In other words, by predicting the future behavior of the user and the outcome of those actions, the system can aid users in achieving their goals and learn without the need to fail first.

To successfully include feedforward, one should not give the user information about something they need to do 'far' into the future as such information will only stay in short term memory for 18 seconds [24]. Also, it is easier for users to learn a skill when the action and concurrent reaction is linked in time [22].

4.3 Domain expert users

Costabile et al [27] define domain expert users as 'experts in a specific domain, not necessarily in computer science, who use computer environments to perform their daily tasks'. Alas, domain expert users are professionals who have had the time to develop intuitive expertise and knowledge in one area [22].

It is important to recognize that even domain expert users of the same community are different in many ways, [27]. In fact, expertise is not a single skill, but a collection of skills [22]. As a result, two experts in the same field may use different skills or strategies to solve the same task.

This should be reflected in the system by offering the user different materializations of information, such as text, visuals, sound, haptic feedback etc. [27].

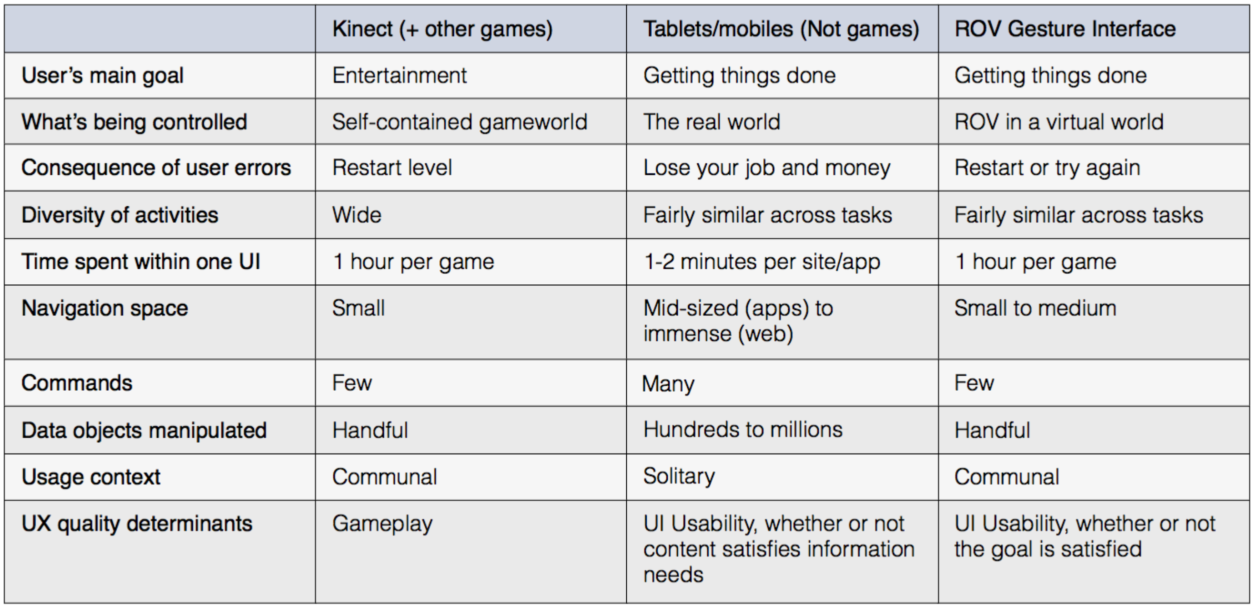
Another interesting aspect is that 'using the system changes the user, and as they change they will use the system in new ways' [28]. The co-evolution of users and systems has its origin from two main points [27]:

1. Creativity; the user finds novel ways to fulfill needs that are not accounted for in the design.
2. Acquired habits; the user follows an interaction strategy they have become accustomed to.

4.4 Gestures in gaming vs. apps and web

The Xbox Kinect and Playstation Camera deliver immersive experiences that rely on tracking of hands and body parts to control their games and interfaces, without any physical devices. They are successful, with the Kinect selling 4 million units its first six weeks [29].

Even though gesture interfaces have been quite common on gaming platforms for a few years, they have not come into everyday interfaces like computers, mobiles or tablets. Table 1 uses research by Nielsen [29] to highlight the difference between the two different platforms. This makes it is easier to decipher why gestures only work with gaming environments at the moment.



	Kinect (+ other games)	Tablets/mobiles (Not games)	ROV Gesture Interface
User's main goal	Entertainment	Getting things done	Getting things done
What's being controlled	Self-contained gameworld	The real world	ROV in a virtual world
Consequence of user errors	Restart level	Lose your job and money	Restart or try again
Diversity of activities	Wide	Fairly similar across tasks	Fairly similar across tasks
Time spent within one UI	1 hour per game	1-2 minutes per site/app	1 hour per game
Navigation space	Small	Mid-sized (apps) to immense (web)	Small to medium
Commands	Few	Many	Few
Data objects manipulated	Handful	Hundreds to millions	Handful
Usage context	Communal	Solitary	Communal
UX quality determinants	Gameplay	UI Usability, whether or not content satisfies information needs	UI Usability, whether or not the goal is satisfied

Table 1: An overview of the difference between games and other platforms in terms of usability needs. [29] Proposed attributes for the virtual ROV gesture interface is in the right column for comparison.

5. DISCUSSION

From research, it is evident that there are many different views on gestures and how they should be used to create interfaces. While Hummels and Stappers [14] and Golod [11] are opponents to the use of gestures as an addition to speech, Cassell [12] and McNeill [10] are proponents.

Table 1 compares the use of Kinect for games with the use of tablets and mobile phones for non-gaming. A gesture interface controlling a virtual ROV will not solely have elements from only one of the columns in the table. The last column in table 1 proposes how this interface can be put in the same context.

The contents of table 1 are important for analyzing whether speech will be used to accompany the gestures. Cassell claims speech will give the communication greater depth and the user more freedom, while it is instantly intuitive [12]. However, in an interface with a small navigation space, few commands, and only a handful of objects to manipulate, depth and freedom is not important. Then it is more natural to follow the sign language approach by Golod et al, or Hummels and Stappers who say that human-computer interaction should be different from human-human interaction.

Since few commands are easy to memorize, the most important aspects of such an interface is that commands are driven by rules, meaning that they give the same output every time, and vice versa: that one output is driven by one command. An example is simply that one unique command, such as raising a hand, controls one aspect of the ROV, like movement up and down. This will ensure repeatable operations, and simplicity over time.

This poses a new question the author has not found any studies of and which has implications for how gesture interfaces can be designed: if a gesture interface consists of pre-programmed gestures, will they still be processed by the same part of the brain as spoken language, or will they become a motor skill, procedural memory, like playing guitar or throwing a ball? If the latter is true, then gestures in the context of interfaces end up being analog input, whereas the true potential is that they can also carry meaning. [10]

The space gestures seem like the most fitting for the control of an ROV. They consist of gestures for manipulating objects, like pushing them up or down, which is intuitively what you would do with an ROV. A supportive group of gestures can be the Symbols, which include the 'V' sign for victory, who can serve as complimentary options. The propositional gestures Cassell suggests could be applied to an interface with speech recognition, as they are important in certain task-oriented communication [12, 30].

Controlling an ROV is an example of a skill, and not a particular function. It is therefore not as crucial for the user to intuitively master the skill at first try, unlike calling someone from a phone. The crucial part is that the system evolves with the user, learning the variations in how the user creates certain gestures and the rhythm of the communication. While applying machine learning to know how the user behaved in the past, the system should also predict how the user's interaction strategy would evolve in the future. That way, the system will be tailored to each unique user, counteracting the fact that all users, even domain experts in the same field, are different [27].

The communal aspect of the interface, that pilots can teach each other, can be reduced if the system evolves uniquely with each user. However, as most users will be power users, an approach that favors flexibility and adaptability to accommodate the needs and creativity of the user is better than aiming only for simplicity.

The most important points from Nielsen et al [16] in section 4.1 will in this case be to avoid static positions and repetition. Stressed joints will negatively affect concentration and performance. Hence, the interface should promote frequent periods of rest, varying tasks and activation of most joint and limbs without being tiresome. Also, the gesture 'amplitude' should not be too big or too small, at least not for a prolonged period of time.

To control ROVs, it is apparent that a high level of accuracy must be obtained. A proposition is to give the user the possibility to vary the sensitivity of response. This will allow the user to continue to apply movements that are within the amplitude levels given by Nielsen's gesture interface rules, while either making very small or very big adjustments to the vehicle itself.

Fatigue is important to avoid in order to maintain a high level of accuracy. To give the user sufficient rest between gestures, a cruise-control function can be applied. With a set speed and direction, the user only needs to interact with the system when changing these settings.

Assuming the user already knows where to go, another possibility is to divide the trip into several stages by having the user create waypoints in 3D space, and then have the ROV follow these waypoints automatically afterwards. This will take advantage of the superiority gestures have in interacting with 3D environments.

Neutrality of the system is an issue with gestures compared to regular joysticks. Joysticks will always return to a neutral position in the center, but for gestures it is difficult to define a center in 3D space where the vehicle will have no direction. In this case, it could be even better to have a gesture signaling a neutral position, relying on automation of the system to level the vehicle in space. On the other hand,

Another solution is proposed by 'Imaginary Interfaces' [31]. They enable users to create the origin of a coordinate system by making an 'L'-shaped gesture with the non-dominate hand. Hence, all gestures made by the dominant hand are made according to the reference hand, and the benefit is that the user will have a visual representation of the origin in relation to the control.

As discovered, feedback is essential to learn a new interface. Visual feedback is pretty easy in this case: the user will quickly see if the ROV behaves differently from what he expects.

Though, in a virtual world, conditions can be perfect. This means that there is no debris rushing past the screen to indicate speed, or no difference in light showing depth. Other ways of feedback must be tested in order to find intuitive ways of reading the behavior of the ROV and the environment.

Additionally, there is secondary functionality, such as controlling arms or tools of any sort. Many ROVs have robotic arms similar to human arms that perform difficult operations. By using gestures, the robotic arms can mimic the movement of the human arms, creating a one to one relationship between input and output. To give force feedback here is challenge without any physical device. Testing must be performed to learn if immersion is important, and, if so, if a physical device will affect immersion.

For long-term use, gesture interfaces using physical devices are more expensive due to durability, maintenance and security [9]. Also to consider, is the possibility of discomfort and added fatigue when using such devices.

With HMDs such as Oculus Rift, which has embedded accelerometers and head tracking, the additional two degrees of freedom can be given to the user himself, by allowing the user to tilt his head up/down and left/right without actually rotating the vehicle itself. The extra dimension of reality in the virtual will immerse the user even more, hopefully making gestures more natural. Another idea is to display a virtual model of the body within the HMD, so the user can se the gestures he is performing.

The fact that ROV-pilots use 2D maps for navigation can create a cognitive load when the pilot uses 3D gestures to control the ROV and has to perform mental rotations on the go. A solution might be to lock the vertical position when navigating on the 2D map. Another solution is to have a 2D map over the YZ axis instead of the classic XY, or, in other words, a 2D representation of what is coming ahead of the ROV.

6. CONCLUSION

This article has performed a literature review to analyze if a gesture interface can be applied to control ROVs in a virtual environment. Research has focused on understanding the problem space, which is 'controlling ROVs today', and learning about the proposed solution, which is 'using gestures to control the ROVs'.

Even though there is a lot of research out there and real life implementations such as the Kinect, many of the issues need more work for a gesture interface to work better than current solutions using joysticks, e.g.:

- Feedback from a device versus immersion into the virtual world.
- Simplicity for beginners or flexibility for experts.
- Neutral positions of the system
- Accuracy
- Fatigue

However, if there is one aspect that needs more research, it is in the development of a general norm for gesture interfaces. Continued research and, perhaps more importantly, practical experience in the field is key to figure out which approach is the best fit for ROV gesture interfaces.

REFERENCES

[1] (2015, 04.03.15). *Myo - Gesture Control Armband*. Available: https://http://www.thalmic.com/en/myo/

[2] (2015, 04.03.15). *Leap Motion Inc*. Available: https://http://www.leapmotion.com/

[3] (2015, 05/03/15). *Remotely operated underwater vehicle*. Available: http://en.wikipedia.org/wiki/Remotely_operated_underwater_vehicle

[4] (2015, 05.03.15). *USA: Forum Inks Five-Year Supply Agreement with Canyon Offshore*. Available: http://www.dredgingtoday.com/2011/11/15/usa-forum-inks-five-year-supply-agreement-with-canyon-offshore/

[5] L. Hale, "Principles and techniques for designing precision machines," Ph.D., Massachusetts Institute of Technology. Dept. of Mechanical Engineering, Massachusetts Institute of Technology, 1997.

[6] L. G. García-Valdovinos, T. Salgado-Jiménez, M. Bandala-Sánchez, L. Nava-Balanzar, R. Hernández-Alvarado, and J. A. Cruz-Ledesma, "Modelling, Design and Robust Control of a Remotely Operated Underwater Vehicle," *International Journal of Advanced Robotic Systems*, vol. 11, 2014.

[7] (2015, 05.03.15). *6 Degrees of Freedom*. Available: http://www.carrlane.com/siteData/webRefs/Images/JigFix_03-01.gif

[8] M. Danesi, "Kinesics," in *Encyclopedia of Language & Linguistics*, ed, 2006, pp. 207-213.

[9] F. Sen, L. Diaz, and T. Horttana, "A novel gesture-based interface for a VR simulation: Re-discovering Vrouw Maria," in *Virtual Systems and Multimedia (VSMM), 2012 18th International Conference on*, 2012, pp. 323-330.

[10] D. McNeill, *Hand and Mind - What Gestures Reveal About Thought*: University of Chicago Press, 1992.

[11] I. Golod, F. Heidrich, C. Möllering, and M. Ziefle, "Design principles of hand gesture interfaces for microinteractions," presented at the Proceedings of the 6th International Conference on Designing Pleasurable Products and Interfaces, Newcastle upon Tyne, United Kingdom, 2013.

[12] J. Cassell, "A framework for gesture generation and interpretation," in *Computer Vision in Human-Machine Interaction*, R.Cipolla and A. Pentland, Eds., ed: Cambridge University Press, 1998.

[13] J. Xu, P. J. Gannon, K. Emmorey, J. F. Smith, and A. R. Braun, "Symbolic gestures and spoken language are processed by a common neural system," *Proceedings of the National Academy of Sciences,* vol. 106, pp. 20664-20669, December 8, 2009 2009.

[14] C. Hummels and P. J. Stappers, "Meaningful Gestures for Human Computer Interaction: Beyond Hand Postures," presented at the Proceedings of the 3rd. International Conference on Face & Gesture Recognition, 1998.

[15] M. C. Cabral, C. H. Morimoto, and M. K. Zuffo, "On the usability of gesture interfaces in virtual reality environments," presented at the Proceedings of the 2005 Latin American conference on Human-computer interaction, Cuernavaca, Mexico, 2005.

[16] M. Nielsen, M. Störring, T. Moeslund, and E. Granum, "A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI," in *Gesture-Based Communication in Human-Computer Interaction*. vol. 2915, A. Camurri and G. Volpe, Eds., ed: Springer Berlin Heidelberg, 2004, pp. 409-420.

[17] V. Bellotti, M. Back, W. K. Edwards, R. E. Grinter, A. Henderson, and C. Lopes, "Making sense of sensing systems: five questions for designers and researchers," presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, USA, 2002.

[18] A. J. Glenstrup and T. Engell-Nielsen, "Main Problems of Eye-Gaze Interfacing," Bachelor, Information Psychology, Institute of Computer Science, University of Copenhagen, 1995.

[19] D. A. Norman, *The Design of Everyday Things*. New York: Doubleday: New York, 1990.

[20] M. Lohse, R. Rothuis, J. Gallego-P, #233, rez, D. E. Karreman*, et al.*, "Robot gestures make difficult tasks easier: the impact of gestures on perceived workload and task performance," presented at the Proceedings of the 32nd annual ACM conference on

Human factors in computing systems, Toronto, Ontario, Canada, 2014.

[21] S. Weinschenk, *100 Things Every Designer Needs to Know About People*: New Riders Publishing, 2011.

[22] D. Kahneman, *Thinking, fast and slow*. New York: Farrar, Straus and Giroux, 2011.

[23] S. M. Weinschenk, *How to Get People to Do Stuff*: Peachpit, 2013.

[24] R. Revlin, *Cognition: Theory and Practice*, First ed.: Macmillan Education, 2013.

[25] C. o. D. Studies. (2015, 08.03). *Feedforward and Positive Self-Review*. Available: http://www.creating-futures.org/keystosuccess/feedforward/

[26] (2015, 08.03). *Feedforward, Behavioral and Cognitive Science*. Available: http://en.wikipedia.org/wiki/Feedforward,_Behavioral_and_Cognitive_Science

[27] M. F. Costabile, D. Fogli, C. Letondal, P. Mussio, and A. Piccinno, "Domain Expert users and their needs of software development," presented at the UAHCI Conference, 2003.

[28] J. Nielsen, *Usability Engineering*. San Diego: Academic Press, 1993.

[29] J. Nielsen. (2010, 8.5.15). Kinect Gestural UI: First Impressions. *Nielsen Norman Group*. Available: http://www.nngroup.com/articles/kinect-gestural-ui-first-impressions/

[30] R. A. Bolt, ""Put-that-there": Voice and gesture at the graphics interface," *SIGGRAPH Comput. Graph.,* vol. 14, pp. 262-270, 1980.

[31] S. Gustafson, D. Bierwirth, and P. Baudisch, "Imaginary interfaces: spatial interaction with empty hands and without visual feedback," presented at the Proceedings of the 23nd annual ACM symposium on User interface software and technology, New York, New York, USA, 2010.