SRI Final Summary Report

Virtual Reality Applications in Maritime Domain Awareness

Written for:
MSC Summer Research Institute

Instructors:
Alley Butler, PhD
Blaise Linn

Written by:
Michael Alecci, Victor Carreon, Nicholas Duca,
Juan Elizondo, Jared Hobbie, Vincent Lee,
Daniel Odell, Gabrielle Padriga

Written and presented with the support of the Maritime Security Center, A Department of Homeland Security Science and Technology Center of Excellence.

Submitted on: July 27, 2017
ABSTRACT

This paper discusses the applications of Virtual Reality (VR) in improving Maritime Domain Awareness (MDA). The Unity game engine and the C# scripting language were used to create an environment for the display and inspection of 3D models of ships and hulls generated by sonar scans. By creating a user friendly environment, the quality and efficiency of ship inspection can be enhanced, increasing the number of ship inspections done and improving Maritime Security in a cost effective manner.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>iii</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 The Literature</td>
<td>2</td>
</tr>
<tr>
<td>3 Methodology</td>
<td>2</td>
</tr>
<tr>
<td>4 Discussion</td>
<td>2</td>
</tr>
<tr>
<td>4.1 Environment</td>
<td>2</td>
</tr>
<tr>
<td>4.2 Teleport Pads</td>
<td>3</td>
</tr>
<tr>
<td>4.3 Tooltips</td>
<td>4</td>
</tr>
<tr>
<td>4.4 Movement</td>
<td>5</td>
</tr>
<tr>
<td>4.5 Teleportation</td>
<td>6</td>
</tr>
<tr>
<td>4.6 Zoom</td>
<td>6</td>
</tr>
<tr>
<td>4.7 Dropdown Menu</td>
<td>7</td>
</tr>
<tr>
<td>4.8 Save</td>
<td>8</td>
</tr>
<tr>
<td>4.9 Screenshots</td>
<td>9</td>
</tr>
<tr>
<td>4.10 Quit</td>
<td>10</td>
</tr>
<tr>
<td>4.11 Scanned Models</td>
<td>10</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>11</td>
</tr>
<tr>
<td>References</td>
<td>v</td>
</tr>
<tr>
<td>Appendices</td>
<td>x</td>
</tr>
<tr>
<td>Appendix I – Literature Review</td>
<td>x</td>
</tr>
<tr>
<td>Appendix II – HTC Vive Controllers Layout</td>
<td>xix</td>
</tr>
</tbody>
</table>

# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Left Touchpad</td>
</tr>
<tr>
<td>2</td>
<td>Left Menu</td>
</tr>
<tr>
<td>3</td>
<td>Right Touchpad</td>
</tr>
<tr>
<td>4</td>
<td>Left Trigger</td>
</tr>
<tr>
<td>Page</td>
<td>Section</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
</tr>
<tr>
<td>5</td>
<td>Zoom</td>
</tr>
<tr>
<td>6</td>
<td>Right Menu</td>
</tr>
<tr>
<td>7</td>
<td>Displayed a scanned pier using MeshLab. Mesh conversion of pcd to object file.</td>
</tr>
<tr>
<td>8</td>
<td>HTC Vive controllers layout</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

Drug cartels and other criminal groups, can place waterproof crates filled with illicit materials on ship hulls. These concealments, better know as parasitic devices are attached to the hull beneath the waters surface by divers and removed at the next port. If not intercepted, the contraband housed in these crates can be smuggled into the country. The simplest way to find parasitic devices is through hull inspections, but current inspection techniques are time and manpower intensive.

Today, some ship hull inspections are accomplished using Remotely Operated Vehicles (ROVs). These small underwater vehicles are equipped with both a camera and sonar scanners. Typically, the sonar is primarily used for navigation and the camera is the main tool for inspection. This causes challenges in the turbid waters of ports where the visibility can be reduced to a few feet, dramatically slowing the inspection process. The sonar collects data in 2D but these data can be combined with positional information from the ROV to form a 3D image of the environment.

The console for controlling the ROV is also inefficient. The screen is susceptible to glare while using it outside, which can make it difficult for operators to see the ROV video feed, even if the water is clear. Beyond that, operating the ROV without focusing entirely on navigation can cause damage to the equipment. The tether that keeps it connected to the console can get knotted, or the vehicle itself can get pulled into a propeller or water jet on larger ships. This issue is only exacerbated by the need to get up close to the ship to actually inspect it with the attached camera. Additionally, operators need to record comments about the ship they are inspecting. Using the current system, they would need to put down the controller in order to write notes on anything of interest.

The objective was to fix these issues that arose from using ROVs for the entire process of ship hull inspection. First, a program was needed to make a greater use of the ROVs sonar, as current technology can make a digital 3D model of the hull. Repeated sonar scans made into a 3D hull can be more effectively examined than the real hull in turbid water through use of the camera. Some 2D sonars have the ability to filter out unneeded points in a point cloud. This cloud can then be put through another process in order to create a 3D model of the original object being scanned. Secondly, the team wanted to create an environment for inspection that solves the obstruction of the camera by murky water and glare. Lastly, since these inspections give no option for annotation or note-taking, the final aim was to give users the ability to annotate the ship hull in real time, and save those notes.

The best solution to these problems comes in the form of a headset and two controllers- the HTC Vive. The HTC Vive is a Virtual Reality (VR) Headset that gives users full immersion into 3D worlds. Through a two-person team, where one operator pilots the ROV and gets the 3D ship hull model, and the other operator uses the headset to inspect the hull up close, the team hopes to increase the effectiveness and efficiency of hull inspections, which would increase the probability of detection of parasitic devices.
2 The Literature

An examination of the open literature was undertaken by one professor and two students on the research team. This involved a search of the open literature using the Compendex or Engineering Index Database, Google Scholar, and the open internet. Search terms included virtual or virtual environment combined with (AND function) sonar, or AUV, or ROV, or maritime, or port security. Additionally, the terms sonar and 3D were searched. Only English language readable papers were collected in Portable Document Format (PDF). The resulting 84 papers were at minimum read by both one student (two student team) and one professor. The first draft of the literature survey is included in Appendix 1 due to the length of this literature review.

3 Methodology

The literature describes very limited use of virtual reality with Head Mounted Displays (HMDs) and efforts to include the specific research described in this report were not found. Therefore, utilizing the Unity 3D game engine in conjunction with the C# programming language, the teams developed a program that allows users to efficiently traverse a ship hull model and record concerning elements.

Teams used a state of the art custom desktop computer, that met and exceeded all the minimum requirements of running the HTC Vive VR headset, which allowed an efficient work flow, as no program or demo took up large amounts of time to compile or run. Such a setup meant the team was able to avoid framerate crashes. Robot Operating System (ROS) enabled laptop running Ubuntu facilitated the conversion of .ot files to .obj files.

4 Discussion

4.1 Environment

The environment that was created uses standard Unity environmental assets, a height map of the straits of Gibraltar, and a model of the WWI British naval vessel, the HMS Ajax. The terrain model was generated using built in Unity features and was textured using the Terrain Toolkit asset. Additionally, a skybox depicting mountains and a water floor was obtained from the asset store. To create a sense of realism, an instance of the standard advanced water plane asset was placed above the terrain.
4.2 Teleport Pads

One of the main features that have been implemented in the scene is the ability to create and manipulate teleport pads. All the script involving the teleport pads are located in GenerateCapsules1 and is attached to the left controller.

![Figure 1: Left Touchpad](image)

The first feature involving the teleport pads is the ability to create them. As the user flies around inspecting the ship, they may see something that they want to take a closer look at. The user can push up on the left controllers trackpad and create a teleport pad at their current location. The teleport pads will be placed in the scene with a number right above them. The number on the pads correspond to the drop down options in the menu. The script works by creating a gameobject and adjusting its size and position to match the current position of the camera rig. A canvas will also be created and placed above the gameobject to give it the proper number. The pads are stored in an arraylist so that each pad can be referenced later.

The ability to create pads is an important feature because when the user is inspecting the ship they may see something suspicious. After inspecting, they will want to make a reference point to come back to it at a later time. They can do this by placing down a pad and they will have quick and easy access to it.

Another feature with the teleport pads is the ability to change their color. The user interacts with this feature by moving or teleporting to the teleport pad and pressing down on the left side of the left controllers trackpad. The script for this feature works by using an index to cycle through the arraylist of six colors and changing the color of the pad to match the color in the list.

This feature is important because it helps mark something found on the ship. If something found is potentially dangerous, the color of the pad can be changed to red. This will help the user remember why they left a pad at that specific spot and what they should be looking for.
The third feature involving the teleport pads is the ability to teleport to them by cycling through the pads. When the user pushes down on the right side of the left controller’s trackpad, they will teleport to one of the pads. When using this function the first time the user will teleport to the first pad they made and will teleport in order until they reach the last pad created. The script works by moving the index of the list of pads and setting the camera rig position to the position of that specific pad.

The ability to cycle and teleport to the pads created is important because it allows the user to get to important spots quickly and efficiently. This allows the user to return to a point of interest without having to fly around searching for it.

The final feature involving the teleport pads is the ability to delete them from the scene. The user cycles to the teleport pad they want and pushes down on the left controller’s trackpad. This will delete the pad from the scene. The user can also move to the exact location of the pad and press down to delete it. The script works in two ways. One, by comparing the current location to the location of the touchpad, and then destroying that game object. Second, by taking a variable used when cycling through the teleport pads and using that to specify which one to delete when cycling through the pads.

This feature is important because it is important to remove unnecessary pads from the scene. This will make cycling through pads more efficient because the user won’t teleport to unnecessary pads. It will also clear the drop down menu of the pads no longer in the scene.

4.3 Tooltips

In order to assist new users in the environment, a tooltip feature has been added. The tooltip is a guide attached to the controllers that tells the user what each button does. The user can turn this feature on and off by pressing the menu button on the left controller. Once the tooltip is on, the user can look down at the controller and see it displayed. The script is located in the toolTip script and is attached to the left controller. The script works by first determining if the tooltip is on or off. If it is off, it turns the tooltips on for both controllers. In Unity the pictures of the tooltips are attached to a plane. Those planes are then attached to both controllers. This is done so that no matter where the user is at, the planes will be located on the controllers.
This feature was implemented because it is an important guide to users first using the program. There are many different functions for each controller so it is a reference point to help the user learn as he is using the environment.

4.4 MOVEMENT

In order for a user to inspect a ship a movement feature needed to be implemented. The movement feature in the scene is a flying locomotion. The user is able to fly around so that they can efficiently navigate the ship while inspecting. The user has the ability to speed up and slow down the rate that they fly. The user moves around by touching the trackpad on the right controller in the direction they want to go. In order for the user to change the speed that they move, they will push all the way up or down on the right trackpad. The script is located on the right controller and is named navigationbasicthrust.

The script works by instantiating the controller and establishing the axis on the trackpad. When the user presses on the axis, the code will add force to the vector that corresponds to the direction of the trackpad.
that is pressed. When the user pushes down on the trackpad, the forces that determine movement speed in the code will be incremented. When the user stops pressing on the trackpad, the force is immediately set to zero and the user stops their movement. In unity the camera rig is attached to Navi Base and is the object in the code that is being moved.

This feature was implemented because it’s extremely important for the user to move around the scene freely. If the user couldn’t move around they wouldn’t be able to inspect the ship. It gives them the ability to conduct a very thorough inspection of the entire ship or area that they are looking at.

4.5 TELEPORTATION

The teleportation feature allows the user to quickly teleport to a location they point to with a laser. By pulling and holding the trigger on the left controller, a parabolic laser beam will appear. By hovering the laser over the point they want to travel and releasing the trigger, the user will teleport. Once the trigger is released they will be instantly teleported to the specified location. All the scripts used for the teleporting came from the Virtual Reality Toolkit (VRTK) asset. In unity the scripts that had to be attached were the VRTK Controller Event, Pointer, and Bezier Pointer. Doing this gives users a fully functioning teleportation system.

![L Trigger](image)

**Figure 4: Left Trigger**

The teleport feature was added because it allows users to travel the scene very quickly. The user may want to go from one end of the ship to the other. Teleportation allows the user to do it with one click of the trigger, as opposed to flying. The Bezier pointer is more effective in teleporting in the scene because it allows the user to teleport to objects of any height. A straight line pointer would be more effective going across a flat plane.

4.6 ZOOM

The zoom feature in the scene allows the user to scale themselves bigger or smaller compared to the world they are in. The user grabs both controllers by the grips and moves the controllers further out or closer to each other. As the user is zooming, a cube will appear with a picture that will let the user know
their current scale. If the user wants to reset their scale back to normal, they can pull the right controller’s trigger. The script is called Zoom and is located in the camera rig. The script works by taking the value of the distance between the two controllers. It first gets the initial position and then gets the position as it moves further away or closer together. Depending on which way the user is zooming, the code will alter the local scale of the world and the cube. In order to avoid any complications with the scale and menu, the script gets deactivated when the menu is up. In unity the user needs to attach the controllers and the pictures to the script.

Figure 5: Zoom

This feature was added to assist the user in two ways. First, they can scale themselves larger in order to move around the scene quicker and to get an overall picture of how the scene looks. Secondly, the user can make themselves smaller in order to get a more focused look of the ship hull. When they are smaller, they will move slowly along the ship and the specific spot will look bigger.

4.7 DROPDOWN MENU

The drop down menu is a built in user interface attached to the menu that allows the user to see a list of teleport pads. It can be brought up by pressing the menu button on the right controller. When the user clicks on one of the options, they are teleported to that specific pad. While a cycling method of traversing the teleport pads already exists, this menu allows the user to quickly see all the pads at once and then choose one to travel to. When many teleport pads have been created cycling becomes less efficient. The drop down menu solves this problem by making it easier to find the location the user wants.
The drop down menu is a Unity UI element that was created in the scene and attached to the users camera. The menu is centered in the user’s view and is only visible upon pressing a menu button to toggle it on and off. Along with turning on and off, the menu is updated in real time as the user creates and deletes teleport pads. The script for the drop down menu functions is located in the GenerateCapsules1.cs script, which is attached to the left controller. The Unity UI dropdown already has built in functionality, allowing the user to use a scrollbar to see all possible teleport pad options if the list is extensive.

The drop down methods in the script are realized through three methods that allow teleport pads to be added, deleted, and to teleport to the option in the list. The code for adding and deleting pads to the menu is called from the update method, and is located in the same spot as creating and deleting the pads. This allows the menu to be altered during runtime by the code. When the user selects an option, the drop down menu in unity runs the teleport menu method. The code in this method sets the users camera position to the the position of the teleport pad they have selected.

The script for the laser pointer is located on the right controller and is named ViveUILaserPointer. The user has a laser pointer that comes out of the right controller in order to select an option. They can point to a selection and then pull the trigger in order to confirm their choice.

This specific function sees greater value with a greater number of teleport pads used. As a user creates more pads, they will not want to cycle through them to find the one they need. This menu alleviated that problem by letting them go to the specific pad theyd like to visit, rather than having to hit the button multiple times.

4.8 SAVE

The save feature in the scene allows the user to save their current scene along with the teleport pads they have placed. When they return to the scene, all of their teleport pads and their colors will be in the
exact same spot as when they left the scene. When the user wants to save the scene, they will bring up the menu and click the save button. The script for this feature is inside GenerateCapsules1. The script works by opening the reader and reading a file at the start of the scene. The file has either four or five elements. The first four elements refer to the teleport pad and their coordinates in the scene. The fifth element is a check to see if the teleport pad should be loaded. When a teleport pad gets deleted from a scene, a fifth element is added to ensure that it is not loaded. When the user creates a pad, the pad information is added into another file. When the save button is clicked, all the information from that pad is copied over into the original file that is loaded when the scene starts. In unity, the user will have to put the path to the save file and the temporary file for saving.

This feature was added so that a user can come back into a scene for further inspection without losing all the work they had done. Another user can also continue on with an inspection without repeating the work that a previous user has done.

4.9 Screenshots

The screenshot feature takes a picture of what the user is currently looking at through their headset. It is located on a button labeled "making memories" on the menu. The user brings up the menu and pushes the button with the control pad. A screenshot will be taken and saved in the designated directory.

The code is located in screenshot which is attached to the camera eye in unity. The scripts uses the built in unity function Application.CaptureScreenshot. This function tells unity to take a picture of the current scene or game view and save it as a png file. The code will also hide the menu and so that it is not in the picture. The last thing the code does is it saves the current number of the picture the user has taken and stores them. It will save and load this number every time the scene is entered. This is to ensure that the pictures dont get overwritten or conflict with previous pictures. This can be reset with the reset index option on the script.

In the Unity Editor the user can change multiple options. They can adjust the resolution of the pictures taken and also name the pictures. They have the ability to choose where they want the pictures to be saved on their computer. If the path already exists it will save it there and, if not, the code will create it.

This feature is important because it allows the user to save a picture of the scene on their hard drive. The user can then look at it closer or even send it somewhere else for a second look. It also allows the user to archive anything they find to use as reference for later use.
4.10 QUIT

The quit feature allows the user to leave the current scene and return to the editor. The script is called quit and is located on the camera eye. The user interacts with this function by opening the dropdown menu and clicking the quit button. The code utilizes the EditorApplication.isPlaying and sets it to false. This tells Unity that when the button is pushed, the scene has to be stopped. This feature was implemented so that the user can easily exit the scene without needing another person at the computer.

4.11 SCANNED MODELS

The ROV 2D sonar scan is used to create a 3D visual as an octomap. In order to incorporate this 3D visual in the Unity scene a common file type such as .3ds or .obj is required. By using the Robot Operating System (ROS), pcl and octomap libraries, ubuntu version 14.04, and c++, octomap files (.ot) were able to be converted to point cloud data (.pcd). Using the same methods as described above, the point cloud data files could be converted straight to object files. However, by using MeshLab to attach a mesh rendering to points in the object a smoother object file has been created (Figure 7). Being able to import sonar scanned 3D models will allow users to be able to inspect real world objects.

Figure 7: Displayed a scanned pier using MeshLab. Mesh conversion of pcd to object file.
5 Conclusion

Current methods of ship hull inspection are costly and inefficient. Utilizing virtual reality to inspect ships has proven to be a more feasible and effective method than the existing technology. The virtual environment integrates a multifaceted program designed with user interaction as the paramount priority. Users have the ability to perform an in depth inspection of an existing ship. The ship inspection software allows the user to easily traverse a ship, annotate, and scale arbitrarily. The incorporation of this functionality into one program creates the optimal ship inspection tool.

With current criminal activity there is no debate over the necessity for ship inspections. This inspection tool has the best potential to counter parasitic devices being placed on ship hulls, ensuring a safer Maritime Transportation System. The user friendly interaction will directly translate into more accurate results than previous methods. Overall this inspection software supersedes all current standards and will set the bar as the new paradigm for Maritime Security.
REFERENCES


This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2014-ST-061-ML0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.


This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2014-ST-061-ML0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.


Appendix I – Literature Review

Virtual Reality and its Applications for Ship Hull and Port Inspection

Written for:
Maritime Security Center, Summer Research Institute

Instructor:
Alley Butler, PhD

Written by:
Alley Butler, Victor Carreon, and Juan Elizondo

Written and presented with the support of the Maritime Security Center, A Department of Homeland Security Science and Technology Center of Excellence.

Submitted on: July 27, 2017
1 Introduction

With the rise of low cost technology and our inclination towards an electronic based world, virtual and augmented reality should soon become a common and practical way of conducting business, research, and many things. This change would cause certain methods to be adopted resulting in more accommodating computer systems. For example, completing inspections, such as port and ship hull inspections, would not be as time consuming and have more reliable results.

Current ship hull inspections require experienced divers to physically search and scan for any defects or parasitic devices, which is very dangerous. The danger arises when divers get near the ship propellers, as they can get sucked into them causing severe injuries or death. Not only is the operation dangerous, but it is also very time consuming. In order to inspect all of the ship hull with detail, many diving hours are needed. As mentioned above, there is a high risk when performing ship hull inspections in this manner, therefore the cost of hiring a diver is very expensive.

In order to protect our ports in a more efficient way, a remote ship hull inspection has been implemented. This method requires less time and was more cost effective. Current methods employ Remotely Operated Vehicles (ROVs). Their use in ship hull inspections facilitates the process and allows for faster processing of data. However, even though ROVs are a good way of inspecting the ships hull, there are limitations to their inspection capabilities, especially when operating in turbid waters. Although ROVs have water proof cameras that can capture what was seen underwater, but this proved ineffective, in areas with poor water clarity. With the high turbidity in certain areas, acoustic cameras or sonars were introduced. This allowed for underwater scenes to be captured with no problems because of water clarity. Although sonar improved the capabilities of remote ship hull inspections, a well trained operator is still required in order to operate the ROV and interpret the images, which is still rather time consuming.

In hopes of facilitating the inspection process, virtual reality implementation of ship hull inspections is developed at the prototype level by the extended research team. Integration with the virtual reality interface should provide a means of efficient presentation and documentation of ship hull sonar scans. This literature review was conducted in order to ensure current knowledge of the state of the art regarding virtual reality system implementation for inspection with sonar scans. It has been noted that there exist various applications that are beginning to implement this technology to aid their work; however, the research team was unable to find any examples of the use of stereoscopic virtual reality for the inspection of ships and port facilities.

2 Applications

Different virtual and augmented reality applications for ROVs were found, as a result of a two plus month review. One application is a human robot interface based on mixed reality (Domingues et al., 2012). Domingues et al. create a digitized seafloor site in 3D imagery with the help of ROVs, and use the information gathered to edit interactive, virtually animated environments. He and his collaborators employ an Oculus Rift brand Head Mounted Display (HMD) in a system reported under development. The robots that are controlled are semi autonomous. Domingues et al. advocate a cylindrical control room, and they migrate point cloud data to an octomap. Unlike the present study, where the focus is inspection, the paper by Domingues et al. is on a human in loop control process.

Another use of virtual reality with an HMD is reported by Candeloro et al. (2015). Their work improves...
the telepresence experience. This helps the situational awareness during underwater operations, and to actively control a ROV. In the effort reported by Candelero et al. the HMD used is an Oculus Rift brand for active control of the ROV where the dynamics of the ROV are considered. Again, the focus of this paper is on human in the loop control and not underwater inspection.

Also, Lin and Kuo (1999) created a synthetic subsea scenario to assist the navigation of a ROV in and around an offshore installation. Lin and Kuo use CAD information to construct the representation of the underwater geometry. They employ a ROV Safety Domain as a bounding box that they hope can help avoid ROV collisions with the environment. There is no mention of the use of a HMD and their paper does not report actual usage. Essentially, the research reported by Lin and Kuo does not emphasize sensor feedback to the operator/inspector, and is, therefore, distinctly different from the focus of this effort. Although there have been several reports of the use of virtual environments, especially HMD based, the authors have found no report of the use of HMDs for sensor based underwater inspections.

Other applications involve using a ROV to explore and reconstruct deep underwater archaeological sites. Drap et al. (2007) uses divers and ROVs to collect photographs for orientation and as a way to measure amphorae with photogrammetry using archaeological knowledge. They develop a virtual reality and augmented reality for the visualization of an underwater site with the collection of the data described above. This method provides archaeologists with an improved insight into the data and the general public with simulated dives to the site. Once again, the article by Drap et al. does not use a virtual reality environment for inspection in real time.

Similarly, an interactive Large Scale Visualization Environment (LSVE) is used to detect and classify sonar contacts (Encarnacao et al., 2001). Encarnacao and his team of researchers use the LSVE to rapidly detect a contact in a low signal to noise environment and perform the task of classification. The prototype for this application uses a semi-immersive 3D display system, with multiple and mixed modalities of interaction. In another project, augmented reality is relied upon by the Performance Analysis and Training Tool (PATT) to assess the Mine Counter Measures (MCM) capabilities, which involve the use of an autonomous underwater vehicle (AUV) in conjunction with Computer Aided Detection/ Computer Aided Classification (CAD/CAC) (Mignotte et al., 2009). Mignotte and his colleagues use this program to effectively clear a survey region using augmented reality. The most important feature is to automate mission planning, perform risk analysis and Q-route planning.

Some augmented and virtual environments include, the processing of 3D sonar images to augmented reality models for objects buried in the seafloor (Palmese and Trucco, 2008). Palmese and Trucco describe a processing chain that starts within a 3D acoustic image of the object to be examined, which then ends with an augmented reality model. Originally, the chain included blocks devoted to a statistical 3D segmentation, semi-automatic surface fitting, extraction of measurements, and ends with a virtual reality modeling language (VRML) representation.

In another approach, the Oculus Rift brand HMD is also used with a system of systems approach to present the concept of virtual reality dynamic user interface, for the teleoperation of heterogeneous robots (Mortimer et al., 2016). Mortimer and his team use octomap to process the point cloud data. This method is used to create a voxelized representation of the 3D scanned environment. Once this is complete, the octomap is displayed as a 3D point cloud using the Oculus Rift brand HMD.

These articles mainly focus on the creation of 3D virtual or augmented reality environments for the assistance in ROV navigation, and telepresence. As mentioned, the articles do not use virtual reality for real
time inspection. The authors argue that the inspection process can be faster and more effective, if the environments are fully captured by the sensors and implemented into the virtual/augmented reality environment as a 3D world.

3 Imaging

For the creation of a 3D environment, images gathered from sonar need to be processed and rendered. Currently, sonars create a 2D image that is compiled to create a 3D image. This method does not create the best results. This outcome is because of all the noisy data the sonar captures. That is why the data is processed; things like filters and algorithms to clean the data are used to create better images. There are different types of sonars that were investigated, which include acoustic imaging, and a combination of sonars and cameras. Also, there are different sonar types that are available such as the side scan and the forward scan.

In order to process the data acquired from a acoustical camera, a technique for the segmentation of 3D images is presented (Murino et al., 1998). This technique identifies the most reliable image, likely corresponding to a man made object. Then, it determines the points belonging to the same surface to create the 3D image. These steps are fundamental to supporting object recognition and reconstruction of acoustical 3D images.

There are two different sonar types available. The first to be discussed is the forward scan sonar, which is great for navigation. When performing harbor surveillance and ship hull inspection, drift-free navigation is a concern that needs to be resolved in order to produce better results (Johannsson et al., 2010). Using forward scan sonar, large-scale mosaics of underwater imagery can be registered through Fourier-based methods (Hurtós et al., 2012). A high-resolution sonar was used to develop a snapshot in turbid water where video is ineffective (Rosenblum and Kamgar-Parsi, 1992). The development and assessment of shape recognition algorithms and 3D image formation rely heavily on high-resolution 3D sonar measurements (Lorenson and Kraus, 2009). There are several applications of using dual-frequency identification sonar (DIDSON), like observing aquatic plants, because of its high resolution in dark, turbid waters (Xu et al., 2011). DIDSON provides a bridge from the gap between optical systems and typical sonars (Belcher et al., 2002). Also, methodologies that are able to retrieve local relative geometry of an object through estimation of missing elevations have been observed (Brahim et al., 2011). The issue of reconstruction of 3D points from two sonar views has also been investigated (Negahdaripour, 2010). Multiple viewpoints are associated through algorithms in order to take advantage of the sonar imaging process (Guernev and Petillon, 2015). Special editors are able to analyze data from spatial scenes and synthesize a virtual environment (Zakharevitch et al., 1999). Other applications of forward scan sonar include the reconstruction of scour around bridge piles, which is important in maintaining the stability of bridge foundations (Fadool et al., 2012).

The second method is the side scan sonar, which is very useful when looking for high resolution images. One method to estimate the 3D aspects of the seafloor is using a sequence of sonar images and combining GPS positioning (Sun et al., 2008). Another way of generating high resolution 3D images of the objects in the seabad is combining synthetic aperture sonar with bathymetric processing (Griffiths et al., 1997). The side scan sonar data can be analyzed in systems, such as the Quester Tangent’s QTC MULTIVIEW system for statistical seabed classification, that assign data points to classes by clustering (Preston et al., 2001). Being able to analyze image sequences can lead to long term registration issues. Leblond et al. (2005) developed techniques for determining the displacement between images that have been mapped at different times.
Both sonars and cameras have limitations. Therefore, to achieve a higher resolution some researchers decided to combine the two. A multi sensor registration for the automatic integration of 3D data acquired from a stereovision system and a 3D acoustic camera in close range acquisition was developed by Lagudi et al. (2016). The approach was to integrate an acoustic and an optical system. This approach was to improve the limitations created when using the two individually. Langudi et al created a method for aligning a single camera with a multibeam sonar which is placed on an ROV. Another approach is to merge sonar and monocular images to perform large scale mapping of shallow areas from an autonomous surface vessel, reducing the mission time, cost and risk (Iscar and Johnson-Roberson, 2015). Iscar and Johnson-Roberson use a multibeam sonar data to generate a mesh of the seafloor. Optical images are blended and projected onto the mesh. This is done after a color correction process which increases contrast and overall image quality. Some researchers believe that opti-acoustic stereo imaging is promising, with the registration of the images from both cameras, it provides valuable scene information that cannot be readily recovered from each sensor alone (Negahdaripour, 2007). Negahdaripour states that in order to create an effective inspection strategy, the deployment of both optical and sonar cameras is critical to enable target imaging in a range of turbid conditions. Also, an approach for sparse 3D sonar scans and fused stereo images was created using a limited sensor load, and small GoPro Hero2 cameras (Nelson et al., 2014). Nelson and his team start the reconstruction process by making an evidence grid which is created by mosaicing horizontal and vertical sonar scans. Using these scans, a volumetric representation is constructed using a level set method. In order to get the fine details from a scene, stereo cameras are used and transformed into point clouds and projected into the volume. With these different techniques being offered, there are multiple approaches from which to choose when conducting an inspection with an ROV.

4 Automated Systems and Path Planning

Although the use of ROVs keep a human in the loop for planning and control, many look forward to the elimination of the tether and the human operator. This gives rise to the Autonomous Underwater Vehicle (AUV) or sometime called the Unmanned Underwater Vehicle (UUV). To accomplish this improvement, the machine has to be able to plan and control the execution of the mission. Research in this area includes important contributions that should be examined for the use of ROV ship and port inspection.

Key among the researchers focused on the use of marine underwater vehicles for autonomous inspection is the work of Englot and his collaborators. For example, Englot and Hover (2013) investigated inspection using a Hovering Automated Underwater Vehicle (HAUV). They introduced new techniques for coverage path planning over complex 3D structures. To accomplish this goal, Englot and Hover prove five theorems and provide multiple definitions with regard to roadmap construction using combinatorial optimization. Results are provided for an inspection of the USCGC Seneca. In Hover et al. (2012) the issue of Simultaneous Localization and Mapping (SLAM) is examined for the same HAUV. In this paper, there is a process that includes state estimation, sonar registration, and feature recognition. Results are produced on the SS Curtis and the USCGC Seneca. In another paper, Hollinger et al. (2013) consider Bayesian active learning to provide a robust, non-adaptive planning algorithm that is computationally efficient when compared to adaptive algorithms.

Other work by Englot and his colleagues focuses on path planning in non-marine environments, including multi-objective robot path planning for city like environments for aircraft (Ding et al., 2014), multi-objective, sampling based, minimum risk path planning with obstacles (Shan and Englot, 2015), environments with GPS denial (Bopardikar et al., 2015), and an unspecified land robot (Englot et al., 2016). An information theoretic approach was taken in two other papers (Bai et al., 2016); (Bai et al., 2015). In addi-
tional work, Wang and Englot propose a nested Bayesian committee machine which updates the map with greatly reduced complexity (Wang and Englot, 2016), and Bopardikar et al. (2016) use a novel bound on the maximum eigenvalue of the estimation error to the covariance matrix as the cost function for belief space planning. In (Doherty et al., 2016), Englot and his collaborators fuse local overlapping Hilbert maps where the probabilistic output of the classifier is treated as a sensor, employing sensor fusion to merge local maps. This body of work is theoretically advanced, provides efficient computational algorithms, and is proven by real world examples or computer simulation.

Clearly, Englot’s work is important to the field of automation of robotic function and in marine robot inspection. It should be noted this his work includes the migration of sensor data into an octree data structure or octomap as an efficient information storage configuration.

5 Octomaps

In order to create real time rendering, the reduction of computational time is a key factor. In the creation of maps, the log odds ratio is used, which updates the map by creating a fine grid, each square in the grid is binary. This means that it is either defined by a zero or one, zero meaning the space is empty, and one it is occupied. By scanning multiple the probability of generating a real map of the environment increases. After every scan the map is updated depending on the probability of the space being occupied or empty.

One of the biggest advantages of using an octree is that it allows for an initial coarse visualization which is refined when the tree level is lowered (Loke and Du Buf, 2000). Loke and Du Buf use the integration of a recently developed filter method with segmentation and surface construction methods in order to detect and visualize much smaller structures. One of the most important features, is that after the data is visualized for the first time the processing can be stopped at any point, if a new region of interest is needed. If the region is correctly selected the process continues to create a detailed map. Another method was created using probabilistic map fusion for fast incremental occupancy mapping with 3D Hilbert maps (Doherty et al., 2016). Doherty and his team state that instead of maintaining a single supervised learning model of the entire map, a new model is trained with each of the robots range scans. These types of maps are used incrementally in real world application when there are mapping scenarios with overlap between sensor observations. Another representation of the use of oct maps is an open source framework to generate a volumetric 3D environment model (Hornung et al., 2013). Hornung and his team approach is based on octrees and uses probabilistic occupancy estimation. Not only does it represent occupied space but also free and unknown areas. They use an octtree compression method that keeps the 3D model compact. Hornung et al. demonstrate the approach is able to update the representation efficiently and models the data consistently while keeping the memory requirement at a minimum. In addition to the previous work, a multiresolution visualization framework that is being optimized for dealing with huge survey areas with many gaps was investigated, taking into account both the CPU time and the user interactivity (Loke and Du Buf, 2001). Loke and Du Buf construct a quadtree that allows the elimination of gaps by interpolating available 3D data. The visualization at a high tree level allows the rapid change or adjustment of the region of interest. They also use a very efficient triangulation that allows for a fast interactivity even at the highest detail level. In other work, a process was created to produce a descriptive outline 3D occupancy map using Gaussian processes (GPs) (Wang and Englot, 2016). The GPs process has high computational complexity which has limited it application to large scale mapping and online use. This issue is resolved by using test data octrees, the use of octrees within blocks of the map to prune away nodes of the same state. This condenses the number of test data used in a regression, in addition it allows the fast data retrieval.
The use of the Octree data structure and properly designed algorithms is important to rapid system response required by real world application. And, this leads to a focus on real time processing for effective performance by the entire system under consideration.

6 Real-Time

The most efficient way to obtain data and begin analyzing it is to have systems in place that are able to complete tasks in real-time. When computers, robots, or any other machines, generate data with respect to the environments in which they observe as they are observing, a better workflow can be established. For example, near-real-time visualization is able to achieve remote monitoring and control monitoring of dynamic objects with a publish-and-subscribe mechanism (Santamaria and Opdenbosch, 2002). This means that users can subscribe to any real-time data to receive information, as well as monitor and log events of an underwater job at the same time they are being performed. Various articles exist in which numerous methods of real-time implementation are discussed.

Real-time imaging can be accomplished through several means. A method for visual and acoustical processing for 3D imaging has been presented (Lagstad and Auran, 1996). Since a single sensor is not capable of giving optimal results in all aspects, sensor fusion is often relied upon. Sensor fusion is used to obtain better data in areas where a sensor may be lacking, and real-time application of this method enhances capability for 3D AUV perception.

There exist more applications of real-time imaging, such as identifying scour and beach morphology in 3D using the Coda Echoscope (Cunningham, 2015). Cunningham states how it is critical to understand scour and its effects around subsea structures. This real-time surveying can allow for monitoring and maintaining conditions at an appropriate level. The Coda Echoscope is a powerful device that is able to provide a complete 3D view of every sonar ping in real-time (Davis and Lugsdin, 2005). Davis and Lugsdin continue to describe the device and its capabilities, one of which is its capability in imaging and navigation, making it a great technology for inspection tasks in real-time. Another application being that of a real-time framework used for sonar based autonomous 3D perception (Auran and Malvig, 1996a). Auran and Malvig describe a method for real-time clustering using basic image processing from 3D sonar data. This method could then be combined with an AUV to help with collision avoidance, path planning, and occupancy grid models. Algorithms could also allow for the segmentation of echo clusters within a dynamic 3D sonar image (Auran and Malvig, 1996b). Auran and Malvig state that real-time organization of 3D range data can be achieved by an echo management framework grouping sonar. Similarly, spatio-temporal side scan sonar echo data can be physically represented in real-time (Riordan et al., 2005). Riordan and his team present their modular simulator and the results generated when observing the seafloor. This real-time simulator can be incorporated to AUVs and ROVs for real-time evaluation of vehicle survey control strategies. Another implementation for the advancement of underwater vehicles is that of a predictor based image recognition (Yu et al., 2007). This method uses high-resolution sonar images to predict the presence of an object depending on its viewpoint. The real-time sonar image recognition presented by Yu and his team could also facilitate autonomous inspections.

Feature tracking in real-time is an important feature for underwater tasks, such as simultaneous localization and mapping (SLAM) (Folkesson et al., 2007). Folkesson and his team show that greater consistency in the feature location estimates are shown through parameterization. Mapping and perception in real-time allow for greater control of ROVs and AUVs, which lead to efficient methods for gathering data.
7 Training

Introduced above were some applications of ROVs, and how they are used for the creation of 3D images. There are different training systems that have been developed to simulate the use of different equipment in the various fields as well as in the use ROVs. Since the ROV equipment used is very expensive, training users is crucial to ensure the equipment is not damaged. Some training operations are discussed below with positive results.

One of the fields with the most information with using virtual/augmented reality is the medical field. One of the application where virtual reality was use, was to train doctors for surgery. Seymour et al. (2002) demonstrate that virtual reality training transfers technical skills to the operating room. Seymour and his colleagues created a experimental group with no difference in baseline assessments compared to a control group. The experimental group was required to dissect a gallbladder. The result examiners determined that there was a 29% increase in the correct dissection for the virtual reality trained residents. Also, the residents that were not trained were nine times more likely to fail to make progress, and five times more likely to injure the gallbladder or burn non-target tissue. Another example of how virtual reality is being used in the medical field is the development of a virtual reality based power wheelchair simulator (Sonar et al., 2005). Even though people with disabilities could use the wheelchair to be more independent, third party payers are reluctant to purchase one for these individuals until the person can demonstrate their ability to operate them independently. The use of virtual reality helps protect the person from danger when learning to operate the wheelchair. Not only is virtual reality used for training in the medical field, other applications were found. One of these applications is the use of a simulator that serves as a platform to study the dynamics of ships and ship mounted cranes under dynamic sea environments and as a training platform for ship mounted crane operators (Daqaq, 2003). This was done using a cave automated virtual environment (CAVE), using simulated sea conditions to train operator to operate the (Auxiliary Crane Ship) T-ACS 4-6.

Training for underwater ROVs without virtual reality is very difficult because of the cost of the equipment. Also the terrain in unknown, which makes it easy to lose equipment. With the use of virtual reality, ROV operators enter the field having the knowledge required to perform their tasks without losing equipment. An underwater virtual world can comprehensively model all necessary functional characteristics of the real world in real time (Brutzman, 1995). Brutzman states that virtual world is designed from the perspective of the robot which enables realistic autonomous underwater vehicle (AUV) for evaluation and testing in the laboratory. This means the virtual world is designed to display what an operator will look at when using an ROV. Another application is the development of a virtual platform for a ROV to simulate operation tasks under the sea (Zhang et al., 2015). The model is a structure of a typical ROV which operates 1,000 meters below the surface. Zhang et al. use the virtual platform to train an operator to tele-operate the submerged ROV. This training is done by coordinating a pan tilt camera, the ROV main body, and the underwater manipulator.

In another training application, a program sponsored by to Office of Naval Research developed a virtual environment based training system for remote operated vehicle ROV pilots, called Training for Remote Sensing and Manipulation (TRANSoM) (Fletcher, 1997). Fletcher created a dynamic model which may be tailored based on individual vehicle characteristics. This model needed to fit very specific criteria. The simulation needed to have sufficient fidelity to provide an effective training tool in place of the real system. The simulation was compared to the performance of the Imetrix Talon ROV. Also, ROV simulators are successfully used in training sessions for operators who control vehicles when training on real objects is too expensive (Fabekovic et al., 2007). In order to create a realistic virtual environment, Fabekovic and his team developed a modular structure with the ability of receiving signals from an in use ROV. Similarly, another
prototype that was created for training is a Virtual Environment Intelligent Tutor which implements various
training aids (Fletcher and Harris, 1996). Fletcher and Harris create a way to promote the development
of both the sensorimotor and cognitive skills related to basic maneuvering tasks and situational awareness.
Using a head tracked HMD with an intelligent tutoring system, student pilot behavior is monitored, and
the system offers verbal and graphical feedback, mission review, and performance assessment (Pioch et al.,
1997). Pioch et al. noticed better performance in the categories of depth, control, and adherence to an expert
path when the training was made available.

Multiple successful applications of the use of VR systems for training have been reported, and although
not directly relevant to the topic of VR for inspections, these successes are noteworthy. Not only is the use
of VR very important for training purposes, possibilities are endless for other applications.

8 Other

Several categories have been presented throughout the course of this literature review; however, there are
some topics which cannot be classified into specific categories because they vary in nature, or are a mixture
of various methods. These works are discussed in this section because they are relevant to the focus of this
literature review, but defy further categorization.

Sensor fusion and its application to maritime surveillance system is discussed as it is important to in-
corporate various sensors to obtain optimal performance, which could result in safer maritime security (Gu-
nasekara et al., 2012). Gunasekara and his team state their use of centralized sensor fusion, where the clients
forward data to a central location and an entity at the location is responsible for correlating and fusing the
data. They mention a virtual reality application that visualizes the information fused from the sensors. Sim-
ilarly, a synthetic environment (SE) is discussed and its use in the mine warfare area (Morrison, 1998). The
SE involved the use of a UUV along with a side scan sonar and other simulations. Morrison states the data
was combined using the distributed interactive simulation (DIS), which has a number of applications such
as operator training and sonar evaluation. Some studies have been conducted on transforming a ROV to an
AUV for shallow, coastal environments (Gracanin et al., 1999). A ROV requires someone to operate and
navigate the vehicle, while an AUV is autonomous. With ever growing technological advances, the need for
AUVs should expand.

Furthermore, the approach facilitates interaction between research teams involved in the MODENA
project, as simulation results from a set of autotomized models developed by each team in parallel (Paren-
thonen and Belemaalem, 2008). An integrated development environment (IDE) along with the visual design
environment (VDE) can allow for the direct manipulation programming for ROVs (Maffei et al., 2000).
Maffei and his team state that with this method, a pilot does not have to learn the computer system, but
instead focus their attention on learning the task domain.

In addition, the classification and segmentation of sea mines on synthetic aperture sonar (SAS) side scan
images can help in determining potential objects of interest (Köhntopp et al., 2015). In this work, Köhntopp
and his team use active contours and superellipse without prior knowledge to segment the image in object,
shadow of the object, and background areas. With this information, classification can become much sim-
pler when coming into contact with a familiar object along with its shadow. Likewise, the segmentation of
acoustic 3D sparse images is presented to aid in detection (Giannitrapani et al., 1999). Giannitrapani and his
team present a method for obtaining an augmented representation of objects present in underwater scenes
in hopes of facilitating navigation and inspection of underwater environments for the human operator. Sim-
ilarly, real-time sensor processing with automatic target recognition (ATR) for forward scan sonar is used for mosaicing and the creation of live 3D reconstruction modules that may highlight possible improvised explosive devices (IEDs) (Reed et al., 2010). Reed and his co-authors propose a novel approach for improving situational awareness and help the operator localize possible threats. The proposed solutions provide a means for increased search and inspection capabilities.

9 Conclusion

There are a myriad of ways in which ROVs, sonar, and virtual reality, are being utilized as research and development continues and expands; however, articles were not found that involve the use of virtual reality to enhance the capability of ROVs for ship inspection and port security. As a result, we remain confident that the current vein of research represents a fruitful and open direction for further research efforts.

APPENDIX II – HTC Vive Controllers Layout

Figure 8: HTC Vive controllers layout