



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**REALISTIC REFLECTIONS FOR MARINE
ENVIRONMENTS IN AUGMENTED REALITY TRAINING
SYSTEMS**

by

Jason Nelson

September 2009

Thesis Advisor:
Second Reader:

Mathias Kölsch
John Falby

**This thesis was done at the MOVES Institute
Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE September 2009	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE Realistic Reflections for Marine Environments in Augmented Reality Training Systems			5. FUNDING NUMBERS	
6. AUTHOR(S) Jason Nelson				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) Training systems for emerging threats often require complex, realistic and flexible scenarios. Two recent studies analyzed small boat swarming attacks and found that no adequate training systems exist, particularly since live-firing at multiple targets is impractical. Augmented Reality (AR)-compositing real environments and simulated objects-can overcome this training gap as it allows replacing real ammunition and targets with virtual rounds and boats. Recent advancements in AR address the generation and display of shadows and lighting effects from the virtual objects onto the real scene. However, creating maritime AR environments bears additional difficulties due to the ocean's dynamics and its reflective surface. This thesis presents methods for creating realistic reflections of computer-generated ships on live ocean video. After mirroring the ship, a custom graphics shader is applied to the reflection to distort the reflection and to smoothly blend it with the background ocean video. A user study was conducted in which the participants had to determine the authenticity of real and automatically augmented images, yielding over 30% of augmented images to be considered authentic. improve an AR training system for shipboard personnel in small boat defense, other high-fidelity augmentations for marksmanship or convoy training improve immersion, hence training effectiveness, and could ultimately save money, ships, and even lives.				
14. SUBJECT TERMS Augmented Reality, Fragment Shader, Water Reflection			15. NUMBER OF PAGES 79	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**REALISTIC REFLECTIONS FOR MARINE ENVIRONMENT IN AUGMENTED
REALITY TRAINING SYSTEMS**

Jason A. Nelson
Lieutenant, United States Navy
B.A. University of Illinois, 2003

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND
SIMULATION (MOVES)**

from the

**NAVAL POSTGRADUATE SCHOOL
September 2009**

Author: Jason A. Nelson

Approved by: Mathias Kölsch
Thesis Advisor

John Falby
Second Reader

Mathias Kölsch, Ph.D.
Chair, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Training systems for emerging threats often require complex, realistic and flexible scenarios. Two recent studies analyzed small boat swarming attacks and found that no adequate training systems exist, particularly since live-firing at multiple targets is impractical. Augmented Reality (AR)—compositing real environments and simulated objects—can overcome this training gap as it allows replacing real ammunition and targets with virtual rounds and boats. Recent advancements in AR address the generation and display of shadows and lighting effects from the virtual objects onto the real scene. However, creating maritime AR environments bears additional difficulties due to the ocean's dynamics and its reflective surface.

This thesis presents methods for creating realistic reflections of computer-generated ships on live ocean video. After mirroring the ship, a custom graphics shader is applied to the reflection to distort the reflection and to smoothly blend it with the background ocean video. A user study was conducted in which the participants had to determine the authenticity of real and automatically augmented images, yielding over 30% of augmented images to be considered authentic.

improve an AR training system for shipboard personnel in small boat defense, other high-fidelity augmentations for marksmanship or convoy training improve immersion, hence, training effectiveness, and could ultimately save money, ships, and even lives.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	SMALL BOAT ATTACKS	1
B.	VIRTUAL AT SEA TRAINER AND AUGMENTED REALITY VIRTUAL AT SEA TRAINER	4
C.	AUGMENTED REALITY FOR OCEAN SCENES	6
D.	THESIS STRUCTURE	9
II.	BACKGROUND	11
A.	AUGMENTED REALITY	11
1.	Display Devices	12
2.	Registration Sensors	13
3.	Software	14
B.	ADVANCED BLENDING TECHNIQUES	14
1.	Rendering Virtual Shadows from Real Lighting on Virtual Objects	15
2.	Virtual Lighting Effects on Real Objects	17
C.	AUGMENTED REALITY IN THE MILITARY	18
1.	Augmented Reality for Operations	18
2.	Augmented Reality for Training	21
D.	MARITIME VISUAL CUES	21
E.	CURRENT TRAINING	24
III.	METHODOLOGY	27
A.	SCENE SETUP	27
B.	REFLECTION GENERATION	28
C.	TEXTURE COORDINATE GENERATION AND SHADERS	29
D.	SHADER IMPLEMENTATION	31
1.	Full Reflection Shader	31
a.	Wave Modification Function	32
b.	Wave Effect Pass-Through Function	33
c.	Reflection and Shadow Degradation Function	34
d.	Reflection Blurring Function	37
2.	Physics Shader	38
a.	Wave Distortion, Blurring and Shadow	39
b.	"Physics" Based Reflection	40
c.	Calculate RGB by Pixel	41
IV.	RESULTS	45
A.	USER STUDY	45
B.	DATA	51
V.	CONCLUSIONS	55
A.	CONCLUSIONS	55
1.	Background Differences	55

2.	Difference in Shader Output	55
B.	FUTURE WORK	57
1.	Computer Vision	57
2.	Refine Algorithms	58
C.	SUMMARY	58
	LIST OF REFERENCES	61
	INITIAL DISTRIBUTION LIST	63

LIST OF FIGURES

Figure 1.	Display of Arleigh Burke Destroyer and weapons coverage.....	2
Figure 2.	Aftermath of USS COLE attack.....	2
Figure 3.	VAST Concept Drawing.....	5
Figure 4.	AR-VAST Setup.....	6
Figure 5.	Virtual ocean using a bump map. Image from http://meshuggah.4fo.de/OceanScene.htm	7
Figure 6.	Use of complex geometry generated with a displacement map in a purely virtual environment. Image from http://www.turbosquid.com/3d-models/ocean-waves-max/423154	8
Figure 7.	Reality–Virtuality Continuum.....	12
Figure 8.	Shadows of virtual objects in AR scene. Left: original scene. Middle: virtual objects added. Right: virtual shadows drawn. (Images taken from Madsen, 2003).....	15
Figure 9.	Virtual fire illuminating a real world object (Image from Hughes et al., 2004).....	18
Figure 10.	Bow Wave.....	23
Figure 11.	Ocean scene with white caps.....	23
Figure 12.	Killer Tomato.....	24
Figure 13.	RHIB, Picture courtesy of www.navy.mil	26
Figure 14.	Augmented World Setup.....	28
Figure 15.	Left: object-linear texture coordinates, Right: eye-linear texture coordinates. Generated using 3DLabs GLSL ShaderGen.....	30
Figure 16.	Scene with undistorted reflection.....	31
Figure 17.	35	
Figure 18.	Reflections with large k.....	35
Figure 19.	36	
Figure 20.	Reflection with small k.....	36
Figure 21.	Blurred reflection.....	38
Figure 22.	Concept of physics-based reflection shader.....	39
Figure 23.	Output of physics shader.....	44
Figure 24.	Example of an output image of Physics Shader....	46
Figure 25.	Example of an output image of Full Reflection Shader.....	46
Figure 26.	Two Tested Static Backgrounds. Top: Agua Background. Bottom: Blue Background.....	48
Figure 27.	Ship Textures Used to Generate Reflections. In Order from Top to Bottom: Frigate, WhiteLiner, BlackLiner, LCC.....	49

Figure 28.	Virtual Ships Used to Generate Reflections. Top: CruiseLiner. Bottom: DDG.....	50
Figure 29.	BlackLiner.jpg, Blue Background, Shadow Off, Full Reflection Shader Test Image.....	51
Figure 30.	Cruise Liner, Aqua Background, Shadow On, Physics Shader Test Image.....	51
Figure 31.	Percentage of "real" classifications by type....	52
Figure 32.	Percentage of "real" classifications by background.....	53
Figure 33.	Percentage of "real" classifications by shader..	53
Figure 34.	Percentage of "real" classifications by shadow value.....	54
Figure 35.	Smaller reflection area with physics shader.....	56
Figure 36.	Larger reflection area with physics shader.....	57

LIST OF TABLES

Table 1.	Kernel for blurring reflection.....	44
Table 2.	Factors and Conditions for Test Images.....	47

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to take this opportunity to thank many of the people who have helped get through this endeavor. First of all, I would like to thank God for all of the blessings in my life. My wife, Sarra, and our two boys, Collin and Aiden, for sacrificing so much time over the last two years and giving me the encouragement and motivation to see this through to the end. I would like to thank Dr. Mathias Kölsch and Senior Lecturer John Falby for all of your guidance; thesis and classroom related. You two have been irreplaceable sources of knowledge. A lot of the computer graphics techniques were very helpfully explained to me by some of the Delta3D development team. Many thanks go to Michael Guerrero and Danny McCue. Finally, I'd like to thank my fellow section 081 classmates. Thanks for all of the laughs, support, and instruction. Thank you to everyone. I can't believe it's been two years already.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The United States Navy has some of the most powerful and capable ships that the world has ever seen. However, events such as the attack on the USS Cole in Yemen and the harassment of three U.S. Navy ships in the Straits of Hormuz in 2008 underscore the vulnerability and devastation associated asymmetric threats such as small boat attacks. To overcome this vulnerability an augmented reality training system prototype was developed to train US Navy personnel in small boat defense. One of the problems encountered that is associated with augmented reality use in maritime environments is the difficulty in compositing the virtual objects and the real ocean.

A. SMALL BOAT ATTACKS

Small boat attacks employ small, fast, highly maneuverable vessels to threaten, impede or disable larger ships. The small boats could be armed with rocket propelled grenades (RPG) and attempt to get close enough to launch their weapon or, if loaded with explosives, the intent is to collide with the ship. Typically these attacks involve multiple small boats from various threat axes. The general idea is to overcome the ship's self defense forces with overwhelming numbers of attacking craft.

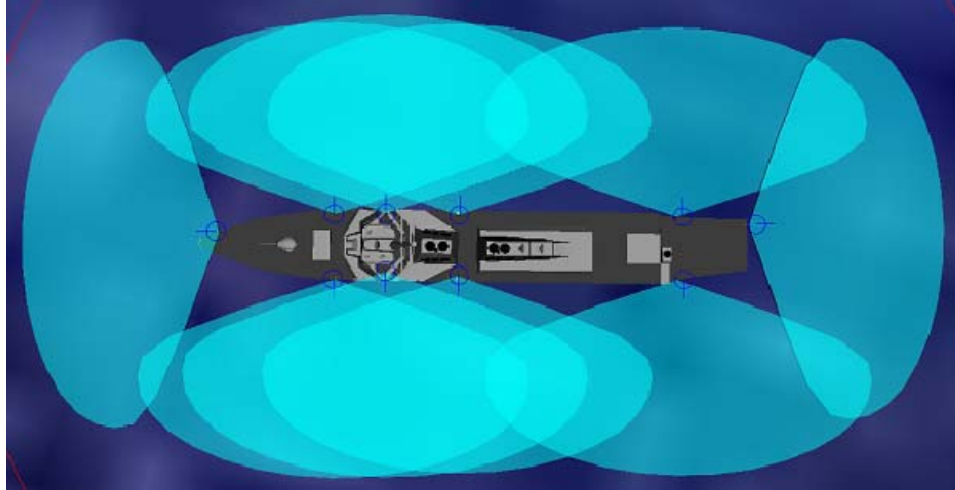


Figure 1. Display of Arleigh Burke Destroyer and weapons coverage

While not an attack of overwhelming numbers, the USS COLE attack of October 2000 displays the effectiveness of just one explosive-laden boat against a significantly larger and more powerful warship.



Figure 2. Aftermath of USS COLE attack

In January of 2008, five Iranian boats aggressively approached three U.S. Navy ships in the Straits of Hormuz. These five boats quickly closed distance on the US ships, possibly coming as close as 200 yards. While the U.S. ships had sent sailors to man their self-defense weapons, at 200 yards and assuming a 15 knot closer speed, the ship would have had 24 seconds ($0.1 \text{ nautical mile} \div 15 \text{ knots}$) to stop the boat from possibly colliding with the US ship. Never mind the fact that at 200 yards the Iranian vessel was well within RPG or other weapons range. This event clearly shows the importance and potential for use of small boat tactics to overpower larger naval ships.

A study conducted at the Naval Postgraduate School (Tiwari, 2008) concluded that two main gaps exist in small boat defense. The first gap is shortcomings in technology. The weapons aboard ship that are used for small boat defense were not originally designed for use on a ship. The original use with the army was against human and lightly armored targets. These weapons are intended to punch holes into their targets. This tactic is not effective against inherently buoyant vessels. In order to disable an incoming threat, the gunner must be able to destroy the engine or the driver, neither of which are easy tasks. The second gap Tiwari recognized is in training. Specifically, current training does not address a shipboard gunner's ability to hit and disable a moving target. Another aspect of the training gap is evident in current force protection requirements (Tiwari, 2008). These requirements do not address swarm training, but rather in-port and boarding-team training.

While the United States maintains a fleet of the most capable and technologically advanced ships in the world, it is not difficult to imagine being overrun by one or many small boats aimed at destroying the ship. Whether it is from restrictive rules of engagement, lack of training or the inability to stop an inbound threat even when it is hit, the ships of the United States Navy have a critical vulnerability to small boat attacks.

B. VIRTUAL AT SEA TRAINER AND AUGMENTED REALITY VIRTUAL AT SEA TRAINER

The Virtual At Sea Trainer (VAST) is designed to train ships and forward spotters in naval gunfire support. The spotters and ship's crew working in the combat information center see virtual representations of landmasses for targeting. The virtual landmasses are then fired at by the ship. Instead of terrain, the rounds land into an area surrounded by acoustic buoys. These buoys triangulate the point of impact and determine the accuracy of the shot (GlobalSecurity, <http://www.globalsecurity.org/military/systems/ship/systems/vast.htm>).

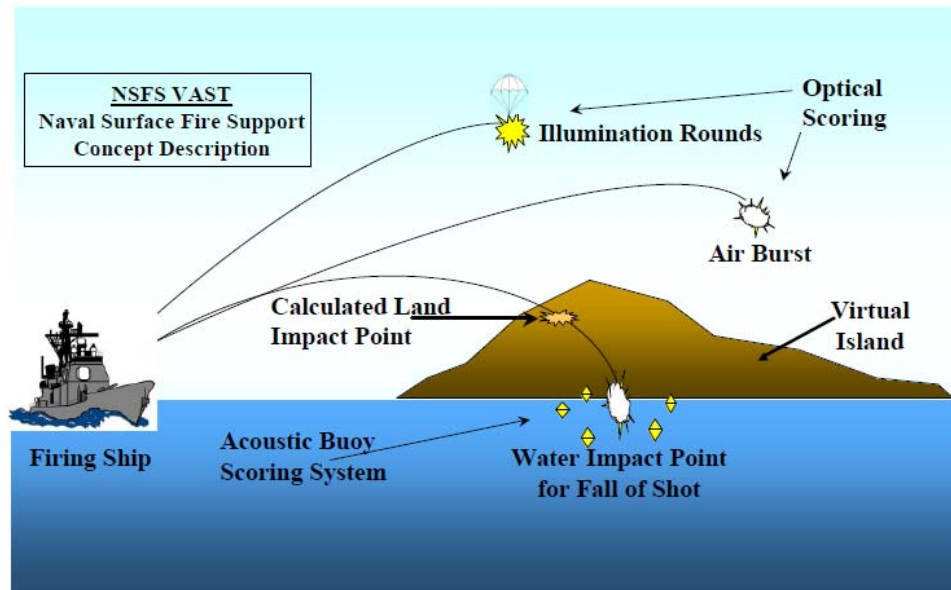


Figure 3. VAST Concept Drawing

The Augmented Reality Virtual At Sea Trainer (AR-VAST) is a prototype deployable crew served weapons trainer based upon the concepts associated with VAST. AR-VAST [Conger, 2008] was developed at the Naval Postgraduate School after Tiwari et al. (Tiwari, 2008) identified a gap in the ways that the United States Navy trained ships' crews against small boat attack.

AR-VAST is designed to take computer generated small boats and inject them into a display of the current operating environment. This is done through four main components: a video camera, an inertial measuring unit (IMU), a computer, and a video display mounted above the crew served weapon. The video camera captures the real world as seen by the trainee and passes that information into the computer which then generates small attacking craft and renders an augmented scene: video background of real world and virtual objects. AR-VAST then uses the IMU

to determine the rotation or movement of the camera so that the virtual world generated by the software rotates in conjunction with the change in the background video. The composited imagery is then displayed on the video display device, currently a liquid crystal display mounted on the gun. The setup is shown in Figure 4.

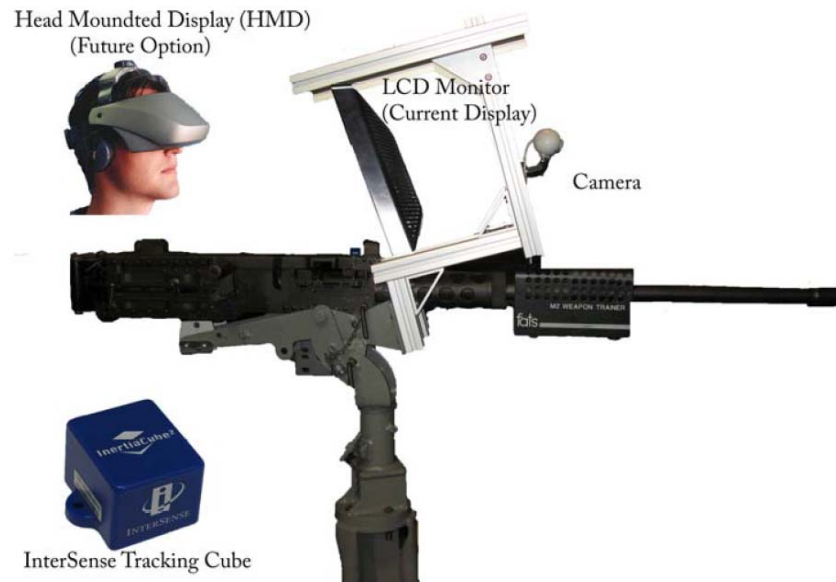


Figure 4. AR-VAST Setup

C. AUGMENTED REALITY FOR OCEAN SCENES

The simulation and rendering of water surfaces has advanced far in the last few years, permitting real-time visual, ocean and wave modeling. A significant contributor to realism includes reflections on the water surface which can be achieved with hardware shaders (for a good tutorial, see Lombard 2004) or even with toolkits such as the Aquatica Engine. In a virtual environment, the ocean surface can be simulated using various methods. One of these methods is using a bump map, as seen in Figure 5 (Meshuggah, <http://meshuggah.4fo.de/OceanScene.htm>).

Another option for creating realistic virtual water is by using a displacement map as seen in Figure 6 (TurboSquid, <http://www.turbosquid.com/3d-models/ocean-waves-max/423154>). A bump map is a data source that is mapped to a texture that defines how the normal vectors change throughout a triangle face while a displacement map actually changes the geometry using a texture (Shirley, 2005).



Figure 5. Virtual ocean using a bump map. Image from <http://meshuggah.4fo.de/OceanScene.htm>.



Figure 6. Use of complex geometry generated with a displacement map in a purely virtual environment.
Image from <http://www.turbosquid.com/3d-models/ocean-waves-max/423154>.

In a purely virtual scene, the exact geometry of the water surface is known, making calculation of reflections and refractions a rather straight-forward task. In an augmented reality ocean scene, this is not the case. Elements of the ocean, such as wave height, wave periodicity, and swell information, are currently not able to be obtained from a video feed. Another piece of information that is specified in virtual worlds, but not in augmented reality, is light source data. The ability to determine the real light source in an environment in augmented reality is a computationally expensive calculation and makes real time processing difficult, if not currently impossible. Clarity and the color of the

water are also immediately available in virtual maritime scenes, whereas this information must be extracted from the video of the environment.

D. THESIS STRUCTURE

After this brief introduction to the pertinent topics, the next chapter discusses relevant background information, such as further discussion on augmented reality and its uses. The methods used in this research, specifically the algorithms used to generate the reflections, are discussed in Chapter III. Chapter IV covers the user study conducted and associated results, which validate the research. Finally, Chapter V lays out the conclusions reached and offers follow-on work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

In order to allow training to be conducted in real operational environments, or at least in real environments similar to the operational environments, AR-VAST is designed using augmented reality. Augmented reality is a technology that focuses on adding computer graphics to real environments. One of the difficult problems associated with augmented reality is the interaction and contact between the computer-generated objects and the real world. The Department of Defense has started sponsoring research and programs designed to use augmented reality for both training and operational functions. This is especially important because of the current lack of effective training against small boat attacks.

A. AUGMENTED REALITY

Augmented reality "... melts the boundaries between computer generated 3D computer graphics and the physical reality of user's environment." (International Symposium on Mixed and Augmented Reality, ISMAR, 2009) In a more practical sense, this means that an operator can view and often interact with computer generated objects that are added to the real world via a display device.

Augmented reality is the representation of virtual objects in a display of the real environment. Milgram, Takemura, Utsumi, and Kishino defined the Reality - Virtuality (RV) Continuum to explain the spectrum of technologies between reality and virtual reality. In between reality and virtual reality is a broad category of mixed reality. Mixed Reality is defined as an environment

where "real world and virtual world objects are presented together within a single display" (Milgram et al., 1994). Augmented virtuality (AV) is the use of physical objects in a virtual world. This is in contrast to augmented reality where the virtual objects are used in the real world.

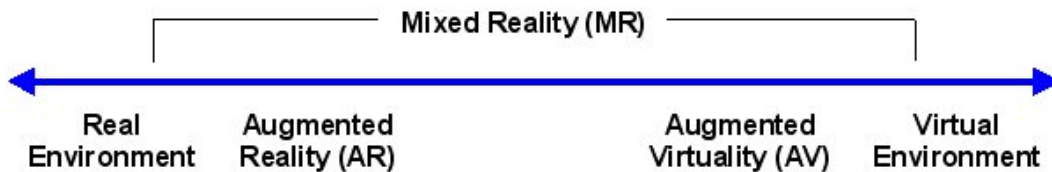


Figure 7. Reality-Virtuality Continuum

Augmented reality systems consist of a video see through, a display device, sensors to track and orient the real and virtual worlds, and a computer to run the software. The video camera captures the current environment while the computer generates the virtual objects. Sensors, such as global positioning systems (GPS), inertial measurement units (IMU), light detection and ranging (LIDAR), or computer vision based recognition software can be used to register the virtual objects into the real world. Once this information is processed the resulting image is displayed.

1. Display Devices

There are various methods of displaying the augmented world in AR systems. These methods include head mounted displays (HMD), retinal displays, hand held displays, projection systems and monitor based systems. The smaller of these systems, the head mounted, retinal, and hand held displays are light weight, portable, and are better suited for displaying information rather than complex computer

graphics. The larger displays, such as the projection and monitor based systems are able to display more realistic graphical displays but are larger and bulkier.

2. Registration Sensors

Different methods of tracking and registering the virtual world with the augmented world can be implemented based on need. These different needs may be based upon the level of movement of the user in the augmented reality world. Six degrees of freedom (6DOF) describes the ability to move in all three dimensions; up-down, left-right, forward-backward and having three axes of rotation; pitch, yaw, and roll. An example of an AR system that has 6DOF may be a device that displays information or objects while transiting through a city, then GPS may be the most effective method of registering the virtual objects into the real world. However, if the movement of the display device is stationary in the real world and only the orientation is changing, such as in a three degree of freedom (3DOF) system where movement is constrained to pitch, roll, and yaw, then an IMU may be the best choice. LIDAR is an optical remote sensing tool that determines distance to an object by measuring time delay between signal sent and received. LIDAR can potentially be used in conjunction with a spherical camera in an attempt to give the 2D image captured from the camera some depth values that are obtained from the LIDAR. This can be used to construct a model of the operating environment, and thus, allow for registration of the virtual objects. Finally, computer vision techniques could be used to recognize

specific buildings or areas which could be used to place the virtual objects in the real world.

3. Software

ARToolKit (ARToolKit, <http://www.hitl.washington.edu/artoolkit/>) is a software library that removes one of the hard problems in augmented reality: tracking the user's viewpoint. ARToolKit does this by calculating the camera position and orientation relative to physical markers called "fiducials" located in the real world.

Depending on the type of AR system, specific software like ARToolKit may not be needed. In particular, augmented reality systems that rely on marker-less tracking techniques may remove the need for specific AR software. AR systems do not need markers if the position of the virtual object is constrained. For example, in the AR-VAST systems, the virtual objects are constrained in their positions on an "ocean" surface. So, with the use of the IMU to determine viewport orientation, there is no need for AR specific software.

B. ADVANCED BLENDING TECHNIQUES

It is widely accepted that good "contact" of virtual objects with one another or with real objects is important, where contact primarily involves shadows, but also reflections and global illumination (see, for example, (Slater et al., 1995), (Madsen et al., 2003)). Past efforts at improving this contact were focused on virtual environments, but select efforts focused on AR scenarios. These efforts are in the areas of shadows and lighting.

Shadows can add depth and elevation to virtual objects being rendered in augmented reality systems. As seen in Figure 8, shadows added to the figure provide the virtual objects a height in relation to the real table that they are rendered onto.

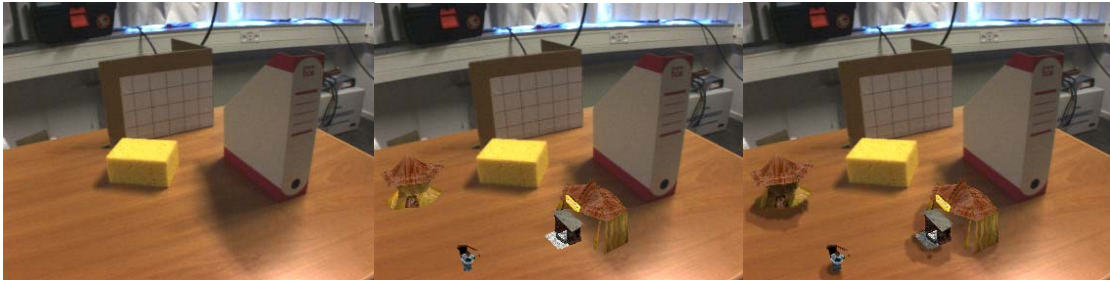


Figure 8. Shadows of virtual objects in AR scene.
Left: original scene. Middle: virtual objects added.
Right: virtual shadows drawn. (Images taken from
Madsen, 2003)

Two other important aspects of augmented world generation are the lighting of virtual objects based on real world lighting conditions and the lighting of real objects with virtual light sources. Rendering shadows on virtual objects based on real lighting conditions requires knowledge of the position of the light source(s).

1. Rendering Virtual Shadows from Real Lighting on Virtual Objects

One method involves finding the real shadows in the scene and then using that information to construct virtual shadows for virtual objects (Madsen, 2003). The real shadows in the scene are found by first segmenting the image by color of the object the shadows are cast upon based on hue and saturation values. In Figure 8, this object would be the table. After the pixels with the table

color are found, a histogram is created of those table pixels' intensity values. The intensity values are indicative of the amount of lighting being applied to the pixel. The method that Madsen describes assumes three levels of lighting, umbra (ambient lighting only), penumbra (mix), and direct lighting. A method of selecting a single threshold on a bi-modal histogram is then applied to the intensity values. This first iteration separates the pixels into those that are directly lit and those that are not. Then with the directly lit pixels removed from the histogram, the remaining intensity values have the same threshold selection process applied. This second threshold separates the umbra and penumbra pixels. If the surface is not dominated by directly lit pixels, then Madsen discusses a modification to the threshold selection function that finds two thresholds for a tri-modal histogram. The knowledge of what pixels fall into each category of lighting allows for the manipulation of a directly lit area into what it would look like if it were actually a penumbra or umbra area. The creation of the virtual shadow is accomplished by taking a sample of the pixels in the real shadow region and finding the average red, green, blue (RGB) values. The average RGB values of the area that is going to have the virtual shadow applied must also be calculated. For each RGB component, subtract the shadow component by the original component and divide by the original as shown in equation (1) for the red component. The alpha value should be greater than the largest of all three RGB components.

$$\frac{R_o - R_m}{R_o} \quad (1)$$

Using equation (2) for each component, the three components of the overlay that will be applied to the directly lit area can be computed.

$$R_a = \frac{1}{\alpha}(R_m - R_o) + R_o \quad (2)$$

Blending of the original surface and the shadow overlay can now be conducted using the RGB components and the alpha values.

2. Virtual Lighting Effects on Real Objects

To conduct lighting and shadowing of real objects from virtual illumination, virtual models of the real world objects must be available in order to apply lighting calculations in the virtual world. The creation of these models based on real world objects must be done prior to rendering the AR scene and thus requires knowledge of the real world environment. For example, in Figure 9 the Canon box that is being illuminated by the light must have a virtual model of it in the software that is creating the light source. The virtual light source is then applied to the virtual model of the real objects in the scene to determine the amount of lighting that should be applied to the real object on a pixel by pixel basis. Then in a fragment shader, the resulting color is determined by using

equation (3) where D is the original RGB values, M is the material reflectance properties, and S is the virtual lighting contribution.

$$C = D + M * S \quad (3)$$

Since the reflectance properties of the material are most likely unknown, the value for D will be substituted in its place (Hughes et al., 2004).

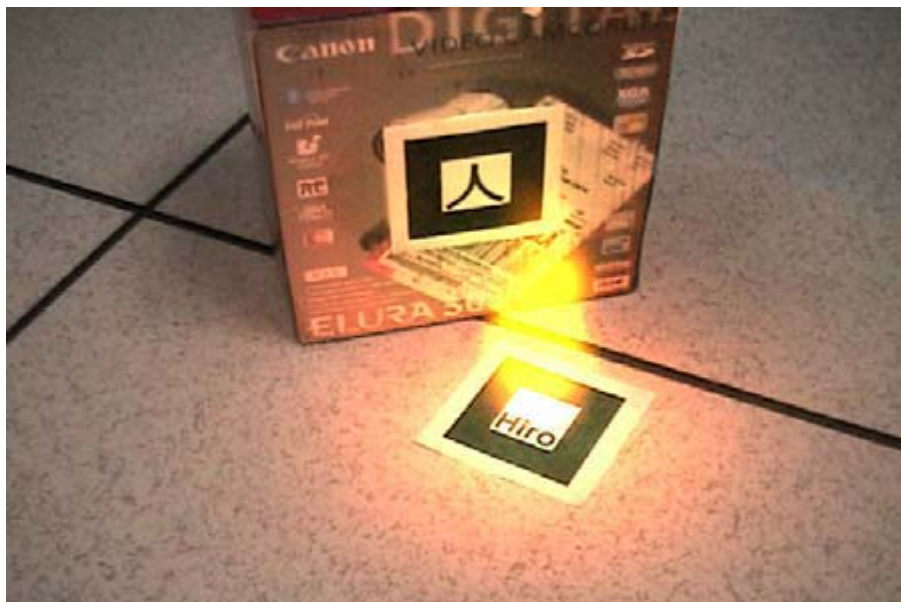


Figure 9. Virtual fire illuminating a real world object (Image from Hughes et al., 2004)

C. AUGMENTED REALITY IN THE MILITARY

Augmented reality can be used in the military for both operational and training purposes.

1. Augmented Reality for Operations

The NPS Vision Lab at the Naval Postgraduate School is developing a panoramic augmented vision system to improve situational understanding during urban operations

(Lindberg, 2009). This investigation is sponsored by the Joint Improvised Explosive Device Defeat Organization (JIEDDO) with the goal to reduce the threat of improvised explosive devices (IED) by allowing soldiers to see historical and locale information about their operating environment. A soldier traveling through a hostile area in a Humvee can view historical data about his operating area, such as a history of IED events near him or the history of military operations in an area. Important locale information can also be viewed by the soldier. Information such as homes and offices of local prominent people (friendly and non-friendly alike), sign translations, and street names could readily be displayed for the soldier's knowledge. All of this information could help increase a soldier's situational awareness of an area without requiring the soldier to look through countless amounts of after action reports, potentially forgetting or losing important information, and can be done in a real-time manner.

This is similar to the Battlefield Augmented Reality System (BARS) developed at the Naval Research Laboratory Virtual Reality Lab. BARS was designed to provide users with a "heads up" display of information and graphics of the real world operating environment (NRL, <http://www.nrl.navy.mil/pao/pressRelease.php?Y=2001&R=01-01r>). Although the systems are similar, differences between BARS and the work conducted at NPS are important to recognize. The BARS system is intended for use by a dismounted infantry man, as opposed to the NPS work that is designed for use on a vehicle. This leads to another difference between the systems. The BARS system utilizes a

HMD to display the world and augmentations to the user because of the mobility of the individual, while the project at NPS is using a large monitor due to the user being isolated inside of the vehicle.

Much like proposed firefighter uses, augmented reality could be utilized for damage control efforts aboard ship. Imagine a large conflagration aboard ship. To ensure the survival of the ship and minimize damage, the damage control teams must quickly isolate water systems, electricity and fuel systems. An AR system would help to minimize the likelihood of missing one of these important valves or switches in a high stress environment.

Another use of AR for the Navy is the development of a system that will help amphibious landing craft to navigate minefields safely by marking the mines on the display (Krane, <http://articles.latimes.com/2002/may/13/business/fi-techextra13>).

As mentioned above, augmented reality is being explored as a means to train small boat defense on ships with AR-VAST. AR-VAST also had potential uses in operational environments in addition to the previously discussed training uses. The system could potentially be connected with the ship's tactical data systems allowing the Combat Information Center (CIC) to interact with the weapons operator. This interaction would allow the weapons operator to provide information: speed, heading, and location of small boats to CIC that cannot be seen by radar. Additionally, CIC can also cue the weapons

operators toward potential threats before the threats are in visual range; providing early warning for the weapons operators.

2. Augmented Reality for Training

Augmented Reality can also be used for training purposes. Training using augmented reality allows the trainee to practice skills in environments that are similar to, or in the actual operating environment itself, exhibiting similar benefits of a live simulation. In addition to the ability to train in real environments, augmented reality training systems can allow training on the actual equipment that the trainee would use in a real world operation. Augmented reality for training also gives the trainer repeatability. Like virtual simulations, augmented reality trainers can be configured to meet specific training needs and can be restarted and reused to train numerous trainees ensuring that the trainees receive the same training. Augmented reality can allow the military to use the repeatability benefits of virtual simulations while also getting the hands-on and environmental benefits of a live simulation. One specific example of using augmented reality for training is the aforementioned AR-VAST.

D. MARITIME VISUAL CUES

Ships and boats operating in the water create numerous visual cues that provide information about the environment. A bow wave is the water that surrounds the front of a vessel as the bow (the front of the ship) cuts through the water, example shown in Figure 10. The wake is the trail of

disturbed water that is left from the ship having moved through the water. A bow wave and wake tell the speed and direction that the vessel is moving. Whitecaps, the sea spray that is produced by ocean waves, will give indication of general wind speed and direction. White caps also aid in the determination of the sea state, allowing for an estimation of the wave heights. Whitecaps are shown in Figure 11. Reflections of objects in the water are also evident in maritime environments and can tell about the current water conditions. Reflections can aid in noticing wave direction and speed by offering a contrast to the water. Additionally, the amount of distortion in the reflection can assist in determination of sea state.



Figure 10. Bow Wave



Figure 11. Ocean scene with white caps

E. CURRENT TRAINING

Currently two methods are used to train against the above mentioned small boat threat. The first of those methods is the use of a "killer tomato". A killer tomato, seen in Figure 12, is a large inflatable cube that is placed in the water by the ship's company and used solely as a target for the gun crews to aim at. Because of the stationary nature, color and shape of the killer tomato, no training of detection, identification, hostile intent determination, or watchstander communication can effectively be completed. The only benefit offered by the killer tomato is the ability to shoot live rounds at the target.



Figure 12. Killer Tomato

The second method used to train the ship's self defense force is to place the ship's own small boat or Rigid Hard Inflatable Boat (RHIB) in the water with part of the ship's crew as seen in Figure 13. Use of the RHIB does allow the ship's company to practice tracking of a small

boat maneuvering through the water, however many other aspects of small boat defense are lacking. Since the ship has to place the RHIB in the water and because RHIBs rarely operate past visual range of the ship for safety reasons, the trainees already have a general idea of where the "threat" is located. Additionally, due to the use of the ship's RHIB, the trainees know that when they can identify the RHIB they have effectively spotted a hostile contact. Unlike in the real world where ship's company must determine hostile intent and relay information about a vessel that may be easily confused with civilian vessels, a RHIB is easily identifiable and offers a familiarity that allows for easy communication. Finally, the obvious risk to personnel safety allows for no actual firing of ship's weapons when using the RHIB for training. No single system allows for complete training of all aspects of small boat defense on United States Navy ships. That is the goal of AR-VAST.



Figure 13. RHIB, Picture courtesy of www.navy.mil

This chapter has covered the motivation and work related to the generation of realistic reflections in maritime environments in augmented reality training systems. The related work in water scene generation in completely virtual environments and the current state of research in other augmented reality visualizations underscore the current lack of methods for reflection generation in AR systems.

III. METHODOLOGY

The goal of this thesis is to improve the blending of virtual with real objects. Two separate effects are created to this end: reflection of virtual objects and shadows cast onto the real water surface. Reflection and shadows are further modified based upon information from the video background to make it appear more realistic. The following sections describe the scene setup, then the three techniques that create the virtual reflections: 1) rendering the reflection to a texture as frame buffer object, 2) modifying the reflection texture with a fragment shader, and 3) blending the reflection and video textures onto the ocean surface, again in a shader.

A. SCENE SETUP

The scene setup is described and schematically shown in Figure 14.

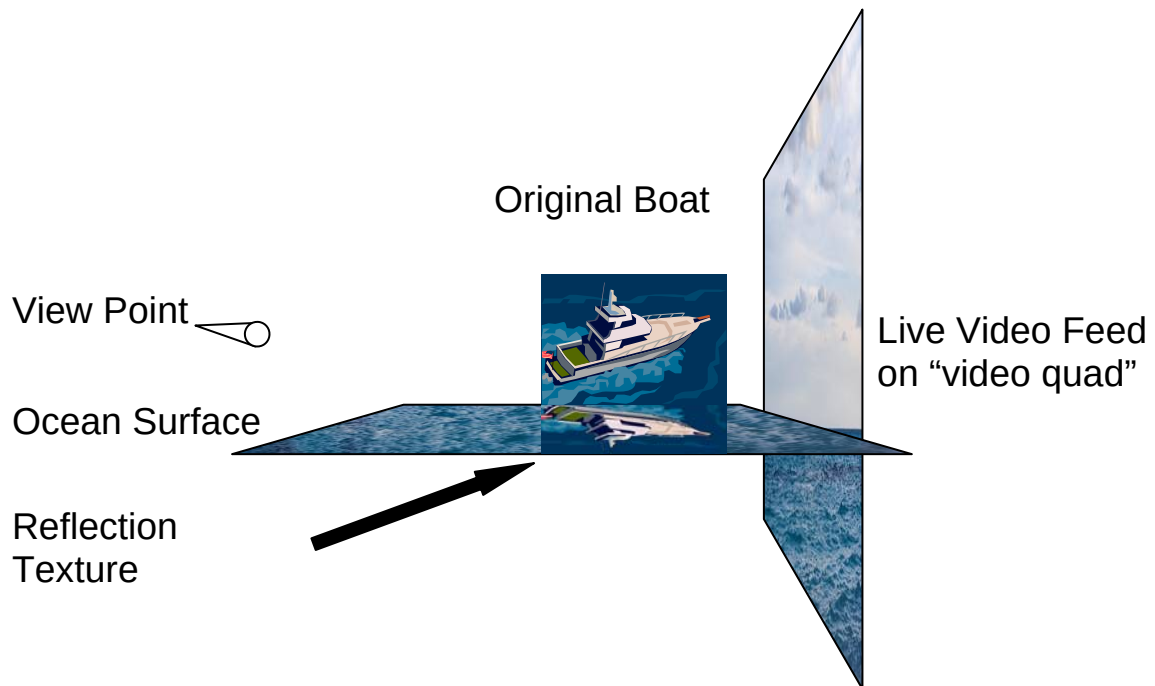


Figure 14. Augmented World Setup

To create video see-through AR, the (live or recorded) video feed is displayed on an OpenGL quad, rendered with an orthographic projection to fill the screen, without writing into the depth buffer. The remaining scene is projected perspectively. A horizontal "ocean surface" is rendered perpendicular to the video quad and placed so that it projects atop and covers the ocean in the video. This ocean quad is mostly transparent except in places where it will receive the mirrored object textures. Between the video quad and the camera are the virtual objects: a large ocean liner and a small speed boat that is controllable from keyboard inputs.

B. REFLECTION GENERATION

All objects that are to be reflected are rendered to a texture from a new viewpoint. Since the scene is set up such that the ocean quad is at a height (z) equal to zero,

the reflected viewpoint is obtained by scaling the original camera location in the negative Z direction. This scaling places the reflection camera directly below the original viewpoint, with both optical axes intersecting the ocean quad at the same point. The small boat and the ocean liner are then rendered with this reflection camera in a first render pass over the scene, with a texture buffer object set as render target. Later, in the final render pass, the ocean quad is rendered with this reflection texture.

C. TEXTURE COORDINATE GENERATION AND SHADERS

The ocean quad is also rendered with a fragment shader instead of using the fixed-function OpenGL pipeline. This shader analyzes the ocean video to calculate the reflection distortion. Because shaders do not have access to the framebuffer (which contains the video already), the video texture is also attached to the ocean surface, providing the shader with access to the ocean video. In particular, the shader now has access to color values of the video provided real world ocean. Since the video texture is captured vertically and the ocean surface is horizontal, the video textured must be projected vertically onto the ocean surface. To accomplish this, texture coordinates are created automatically in eye-linear space, providing to the user the appearance that the texture is still vertical. The difference to object-linear coordinates can be seen in Figure 15.



**Figure 15. Left: object-linear texture coordinates,
Right: eye-linear texture coordinates. Generated using
3DLabs GLSL ShaderGen**

The scene with the video quad, the ocean quad, reflections of the ocean liner and the boat, but without reflection distortion can be seen in Figure 16.



Figure 16. Scene with undistorted reflection

D. SHADER IMPLEMENTATION

In order to distort the reflection texture and the blend with the background video, two fragment shaders were experimented with using two different methods. One shader, referred to as “full reflection” shader, modifies the reflection as a whole, regardless of wave height and periodicity conditions. The other shader, “physics” shader, determines if a part of the reflection is visible based on an assumed wave height and periodicity.

1. Full Reflection Shader

The full reflection shader modifies the reflection texture coordinates to create a more realistic effect than a simple mirror effect. These modification functions are based on the video texture and described below.

- Calculate local color gradients of video texture
- Add $\sin(\text{gradient})$ to reflection texture coordinates to create wave effect
- If new texture coordinate is shifted a predetermined amount, then set reflection alpha to 0.0
- If background is very bright or very dark, set reflection alpha to 0.0
- Decrease reflection alpha linearly from virtual object's intersection with reflection plane
- Decrease shadow effect linearly from virtual object's intersection with reflection plane
- Blur the reflection

The following subsections will discuss the functions in more detail.

a. Wave Modification Function

Before any modifications to the reflection are accomplished, the reflection appears as if it is an exact mirror reflection. This mirror-like reflection has too much clarity and linear features and is unrealistic in ocean scenes. In order to remove some of this clarity and these linear features, the texture coordinates are modulated vertically and horizontally to approximate the look of reflections off waves. In a single video frame, this avoids the appearance of a reflection off a flat mirror-like surface. In successive video frames, this gives the reflection the appearance that it is moving with the water. Parameters to these modulators are derived from the color gradients in the video texture, calculated between the current texture coordinate and each of the nine texture coordinates immediately "below" using equation (4). c_0 is

the current color with red, green and blue components ($c_0.r$, $c_0.g$, $c_0.b$), and c_n is the color n texels below.

$$grad_n = \sqrt{(c_0.r - c_n.r)^2 + (c_0.g - c_n.g)^2 + (c_0.b - c_n.b)^2} \quad (4)$$

$$normalizedAvg = \left[\frac{2}{(max - min)} * (avg - max) \right] + 1 \quad (5)$$

The minimum *min* and maximum *max* of these nine gradients are linearly mapped to -1 and 1, respectively (see the normalization equation (5)). Additionally, the nine gradients are averaged (*avg*) so that there is one value per texture coordinate. The sine of this value produces both positive and negative changes to the texture coordinates, simulating a smooth and continuous wave-like distortion of the reflection. As the color in the video texture changes, the texture coordinate for the reflection texture is moved left or right by adding the value to the texture coordinate *s* and *t* values, emulating a wave effect. If the texture coordinate shift exceeds a predetermined limit, the reflection color (the fragment) is discarded and the video background used instead.

b. Wave Effect Pass-Through Function

In order to remove the appearance that the reflection is simply pasted on the ocean surface and to create a sort of interaction between the ocean and reflection, another modification that is applied to the reflection texture is the pass-through of particular background wave effects. These wave effects are the brightness associated with the sun's reflection on the

water and the darkness that occurs when the sun's light is blocked from the whitecaps. These occur in the real world when small waves move through the reflection.

This effect is accomplished by determining the background video's color value for a point and if that value is in one of the extremes, then the alpha value for the reflection texture is set to 0.0, discarding the reflection and showing the video instead.

c. Reflection and Shadow Degradation Function

The reflection texture is also modulated based on sea state conditions of the current real world operating environment. The Beaufort Wind Scale, developed in 1805 by Sir Francis Beaufort, delineates 13 levels of classification based on wind speed and sea conditions. This application considers sea states zero through four. Beyond sea state four, reflections are not noticeable. Additionally, the training device that this software is being developed for, ARVAST, is not suited for higher sea states. Small boats in the real world do not have this sea state limitation and can still provide threats to ships in sea states higher than four. This is a shortfall that is not unique to AR-VAST, but to many training systems. Weather influences and deters training routinely.

Once sea state is determined, currently through user input prior to scenario run, the shader uses one of five parameterizations of equation (6) to determine the alpha level (opaqueness) per pixel. The equation is based on the difference *diff* in vertical pixel position from the

current pixel to where the virtual object intersects with the ocean plane. The alpha value is then computed linearly based on this difference.

$$\alpha = \left[\frac{-1}{k} * diff \right] + 1.0 \quad (6)$$

Figure 17.

The variable k is specific to the sea state level and determines how quickly the alpha value falls off with increasing distance from the ship-ocean intersection. A larger k means there is less degradation and the reflection is more “mirror-like” whereas a smaller k causes less reflection to be seen. Figures 17 and 18 show this effect.



Figure 18. Reflections with large k .



Figure 20. Reflection with small k .

A similar calculation determines the amount of shadow cast to the ocean surface. The shadow level is predetermined by user input into one of three classes, zero through two. Shadow level zero causes the shader to produce no shadow. Shadow level one applies a gradual decrease in shadow intensity, similar to the reflection alpha value: closer to the ship-ocean intersection the shadow is stronger and for pixels further from the origin the shadow weakens. Shadow level two provides a very strong shadow immediately below the origin of the shadow, appearing near the waterline of the virtual ship and fades away very quickly. The calculation uses the equations (7) and (8). The alpha calculation above is able to determine the alpha

value of the pixel and set that built in variable for the pixel directly by using the .a vector component. Unlike that alpha calculation, the output of this formula does not have an associated built-in variable in the shader. The output here is used to determine the proportion of original texture color and black. The variable k determines again how quickly the shadow fades, while the variable i is the initial darkness of the shadow.

$$proportion = \left[\frac{-1}{k} * diff \right] + i \quad (7)$$

$$color = \frac{[(1 - proportion) * color]}{2} \quad (8)$$

d. Reflection Blurring Function

The final aspect of distortion that is applied to reflection texture is blurring. Even with the aforementioned image modifications applied, the augmented reflections still appear too sharp to be realistic. To counter this, the final fragment color is the box-filtered (average) value of the eight surrounding texels. Figure 21 shows the blurring effect on a reflection at sea state one without shadow effects.

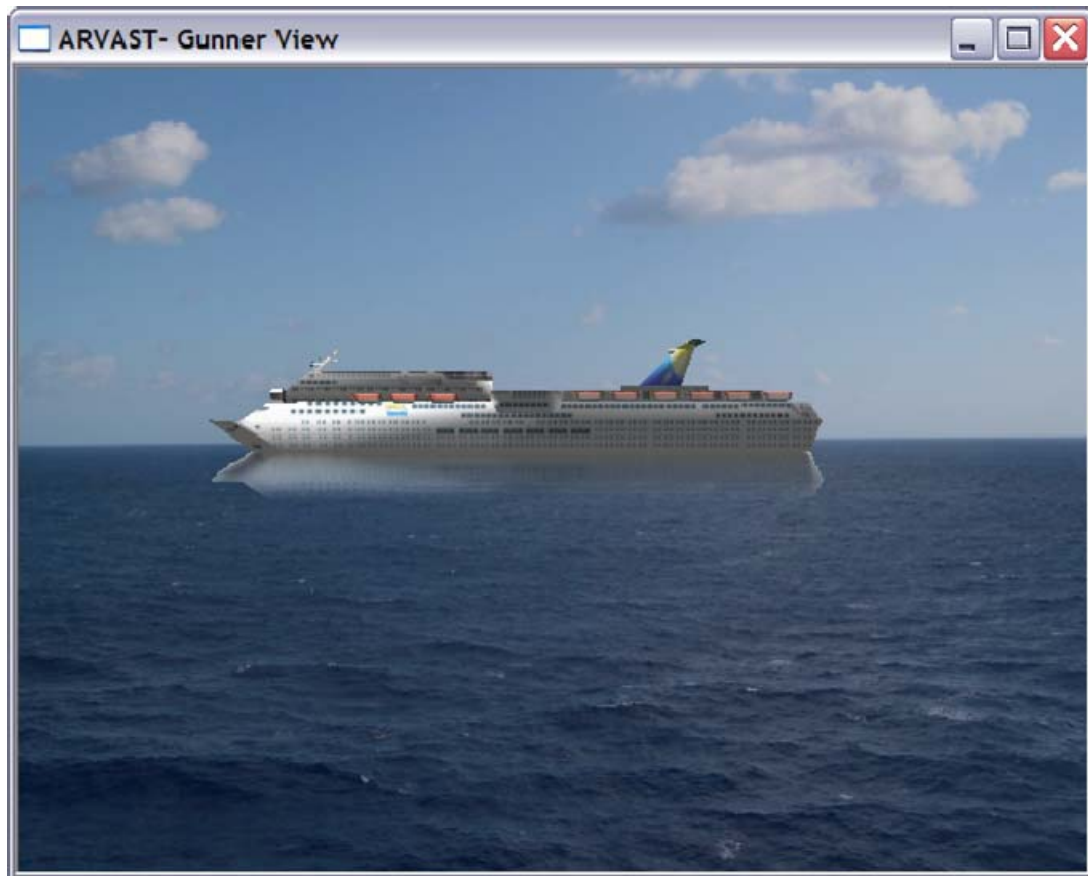


Figure 21. Blurred reflection

2. Physics Shader

With physics shader, the attempt is to generate more distortions and wave interaction with the reflection in the generated reflection by only displaying the reflections in areas where the reflection image would be seen in the real world. In Figure 22, the red vertical bar on the right side of the image represents the "ship." The viewpoint is at coordinate (0, 10) on the left side of the scene. The red spots on the surface represent where the reflection is drawn to the scene. This figure will be covered in more detail later.

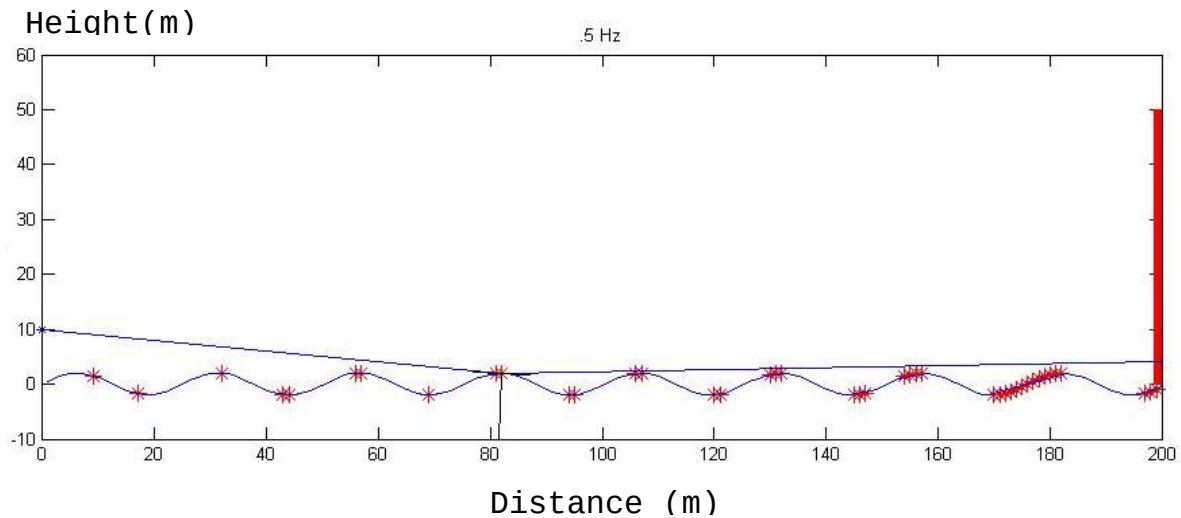


Figure 22. Concept of physics-based reflection shader.

Below the methods used to implement physics shader are outlined.

- Calculate local color gradients of video texture
- Add $\sin(\text{gradient})$ to reflection texture coordinates to create wave effect
- Calculate if reflection is seen for a given pixel value
- Calculate shadow effect decreasing linearly from distance away from virtual object
- Blur the reflection

a. Wave Distortion, Blurring and Shadow

Many of the same methods outlined in Full Reflection shader were reused for the Physics shader. The same texture coordinate manipulations were used to add sine wave distortions in the .s and .t coordinates based on color gradients of the video image. Additionally, the blurring algorithm utilized in the Full Reflection shader

was implemented in the Physics shader to help remove unrealistic visuals from the reflection texture. The shadow function used in physics shader is based on the shadow function used in the Full Reflection shader with some minor modifications. The Physics shader has the world coordinates associated with each pixel, so instead of calculating pixel coordinate differences, the shader is able to calculate world coordinate differences. This allows the shader to base the amount of shadow applied to the reflection texture on the world coordinate distance from the virtual object.

b. "Physics" Based Reflection

The key difference between the full reflection shader and the physics shader is in the application of a more physically realistic model to determine the locations of reflection and no-reflection. The world coordinate of the current pixel is calculated in the vertex shader and passed into the fragment shader through a varying variable. The position of the world coordinate is accomplished by multiplying the model view matrix by the vertex. The OpenGL Shading Language (GLSL) derivative functions `dFdx(var)` and `dFdy(var)` were utilized to obtain the world coordinates for the surrounding pixels. These functions determine the change in a variable, `var`, in either the `x` or `y` direction at the current pixel. This essentially returns a "delta" that can be added and subtracted from the current pixel world coordinate to obtain the world coordinates for the surrounding pixels. The red, green, blue (`rgb`) value for the current pixel and the eight surrounding pixels are then obtained using the below method.

c. Calculate RGB by Pixel

The eye vector, a vector that points from the camera to the current world coordinate, must be determined. The world coordinate is subtracted from the camera position and the result is the eye vector. The world coordinate is on a two dimensional plane represented by the ocean surface. A virtual three dimensional sinusoidal surface is created to provide a wave height for the world coordinate. This sinusoidal surface is generated using equation (9), where HEIGHT, FREQ, SWELL_HEIGHT, and SWELL_FREQ are predetermined constants. The sine wave is a combination of two sine waves, one to emulate the basic ocean wave, with a second (SWELL) to represent a larger swell in the ocean. HEIGHT is the height of the basic ocean wave, while FREQ is the frequency of the ocean wave. SWELL_HEIGHT and SWELL_FREQ perform the same roles for the swell wave. Depth is the world coordinate position of the intersection point along the same axis that runs from the eye point to the ship. The variable leftToRight is the world coordinate position that runs from left to right on the screen.

$$\begin{aligned} waveHeight = & HEIGHT * \sin(FREQ * depth) \\ & + HEIGHT * \sin(FREQ * leftToRight) \\ & + SWELL_HEIGHT * \sin(SWELL_FREQ * depth) \\ & + SWELL_HEIGHT * \sin(SWELL_FREQ * leftToRight) \end{aligned} \quad (9)$$

Additionally, a similar calculation calculates the slope of the tangent line at the same intersection point (depth, leftToRight). The variables and constants are the same as those to determine the wave height, the

difference being that the slope of the tangent function is the derivative of the wave height function.

$$\begin{aligned}
 slopeOfTang &= HEIGHT * FREQ * \cos(FREQ * depth) \\
 &+ HEIGHT * FREQ * \cos(leftToRight * FREQ) \\
 &+ SWELL_HEIGHT * SWELL_FREQ * \cos(SWELL_FREQ * depth) \quad (10) \\
 &+ SWELL_HEIGHT * SWELL_FREQ * \cos(SWELL_FREQ * leftToRight)
 \end{aligned}$$

As stated above, the eye vector is a vector drawn from the current camera position to the world coordinates. Subtracting the intersection point (waveHeight, depth) from the current camera position produces the eye vector.

The intersection point (waveHeight, depth) and the slope of the tangent calculated above are enough to find the tangent vector. The two intercepts (yIntercept, 0) and (0, xIntercept) of the tangent line are calculated, as shown in equations (11)-(14). Subtracting one point from the other will yield the vector for the tangent line.

$$xInterX = -1 * (waveHeight) / slopeOfTang + depth \quad (11)$$

$$xInterY = 0 \quad (12)$$

$$yInterX = 0 \quad (13)$$

$$yInterY = slopeOfTang * (-depth) + waveHeight \quad (14)$$

The normal or perpendicular line is now obtained using the values of the tangent vector. The method for determining the normal vector is simplified because the normal vector and the tangent vector are perpendicular to each other. Because of this, the tangent vector is found by taking the y component of the tangent vector and multiplying by a negative one and placing it as the normal

vector's x component. Then take the x component of the tangent vector and place as the y component of the normal vector.

The normal vector and the eye vector are normalized in order to ensure accurate results when reflecting the eye vector over the normal vector. Dividing both x and y components of the vector by the length of the vector accomplishes the normalization. This allows the reflection vector to be calculated by using the two normalized vectors.

The function used to determine the reflection vector is equation (15). The slope of the reflection vector is used to establish whether or not the reflection vector intersects with the virtual object. The slope of the reflection vector is the division of the y component of the reflection vector by the x component of the reflection vector. The variable, `slopeOfReflection`, is the slope of the reflection vector and `shipDist` is the distance of the ship from the eye point and calculated using equation (16). The result, `testHeight`, is the height of the reflection vector at the distance of the ship. If `testHeight` is less than the ship height, then the reflection vector hits the virtual ship and the pixel receives the rgb value of the reflection texture. Otherwise, if the `testHeight` is greater than the ship height or is less than zero (under water); then the pixel receives the rgb value of the background texture. All nine pixels, the current and eight surrounding, all now have accurate rgb values.

$$reflectionVec = vec2 (2 * normalNorm * (dot(normalNorm, normalEye))) - normalEye \quad (15)$$

$$testHeight = (slopeOf Reflection * (shipDist - depth) + waveHeight) \quad (16)$$

Blurring these nine pixels together removes sharp edges. This is accomplished using a kernel to add some of the surrounding color to the current pixel. The kernel is shown in Table 1.

1/16	1/16	1/16
1/16	1/2	1/16
1/16	1/16	1/16

Table 1. Kernel for blurring reflection

The output of the Physics shader is shown in Figure 23.



Figure 23. Output of physics shader

IV. RESULTS

The output of each shader was tested by a user study aimed at determining the believability of the reflections generated. In this chapter, the user study and associated results are discussed.

A. USER STUDY

A user study was conducted to test the realism of the reflections produced by the two generated fragment shaders shown in Figures 24 and 25. The study consisted of 23 participants, ages 20 to 40, with varying amounts of computer graphics experience. Participants were shown a series of 82 images using Microsoft PowerPoint. Of these 82 images, 18 were photographs of real ship reflections. The remaining 64 images were produced with the two shaders in various configurations. Four independent variables were tested with each image, yielding 2^4 conditions tested on 4 images each. These factors are listed, along with the two conditions, in Table 2.



Figure 24. Example of an output image of Physics Shader



**Figure 25. Example of an output image of Full
Reflection Shader**

Factor	Conditions
Shader	Full reflection shader / Physics shader
Shadow	On / Off
Background	Aqua / Blue
Type	Textured / Virtual

Table 2. Factors and Conditions for Test Images

The “shader” factor (Physics-based or Full) has already been discussed in Chapter III. The second factor was also discussed in Chapter III and was tested to determine whether or not having the shadow effect turned on or off affected the realism of the reflection. The background was varied between two different static ocean backdrop images. These backgrounds are shown in Figure 26. The ‘type’ factor describes if the reflection is produced for a geometric computer graphics model of a ship (virtual) or if it is a flat quad that is textured with an image of a real ship (texture). Figure 27 shows the four types of textured images used and Figure 28 show the two virtual ships used for the user study.



Figure 26. Two Tested Static Backgrounds. Top: Agua Background. Bottom: Blue Background



**Figure 27. Ship Textures Used to Generate Reflections.
In Order from Top to Bottom: Frigate, WhiteLiner,
BlackLiner, LCC.**



**Figure 28. Virtual Ships Used to Generate Reflections.
Top: CruiseLiner. Bottom: DDG**

During the user study, each participant was shown images of the reflections generated by each set of conditions. Only the reflections were shown to the participants; they did not see the original ship or any surrounding environmental artifacts such as the sky or horizon. The images were seen for three seconds and the participant was asked to identify each image as "real" or "not real." "Real" images are those that the participant believes were taken from an actual photograph of a reflection. "Not Real" images are defined as anything that the participant believes were created in the software. The participant provided a verbal response to each image which the study administrator recorded. Two sample images are shown below in Figures 29 and 30.



Figure 29. BlackLiner.jpg, Blue Background, Shadow Off, Full Reflection Shader Test Image



Figure 30. Cruise Liner, Aqua Background, Shadow On, Physics Shader Test Image

B. DATA

Results indicate successful reflection generation and modification under specific conditions and promising directions for the remaining conditions. As can be seen in Figure 31, the real reflection images were identified as "real" in 87% of the cases with a standard error 0.52. Approximately 30% of the textured images were identified as "real" with a standard error of 0.98, while virtual images were identified as real by about 17% of the participants with a standard error of 0.71.

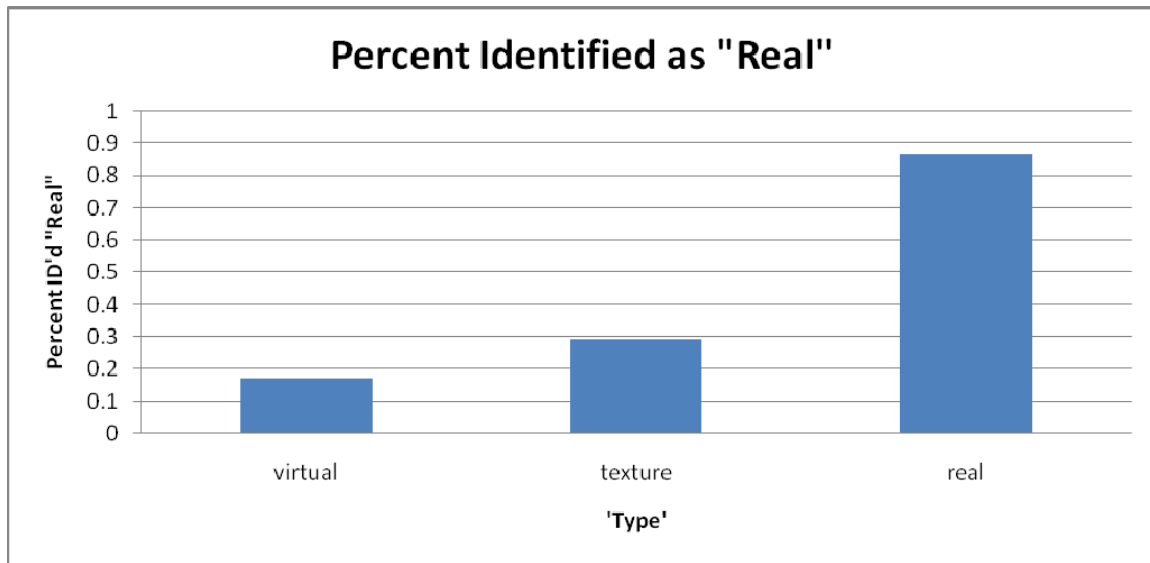


Figure 31. Percentage of "real" classifications by type

By analyzing the specific factors applied to each image, certain trends can be noticed from the data. Both 'background' and 'shader' conditions produced an appreciable difference in the average number of "real" classifications by the participants. When using Aqua Background, on average, 28% with a standard error of 0.83, of the participants incorrectly identified the software-generated reflection as "real." When compared to only 18%, standard error 0.90 of the participants classifying those images with Blue Background. This is shown in Figure 32. Similarly, the factor 'shader' had an effect on the number of false classifications of "real". As seen in Figure 33, the Full Reflection shader led to false classification in 30% of the generated images shown, with standard error 1.0 where as the Physics shader was only 17%, with standard error 0.7. Whether shadow was on or off did not have any noticeable effect on whether the participant identified the

image as "real" or "not real". With shadow on, about 22%, with standard error 0.9 of the users misidentified the software created reflection, while 23%, with standard error 0.8 did the same while shadow was off.

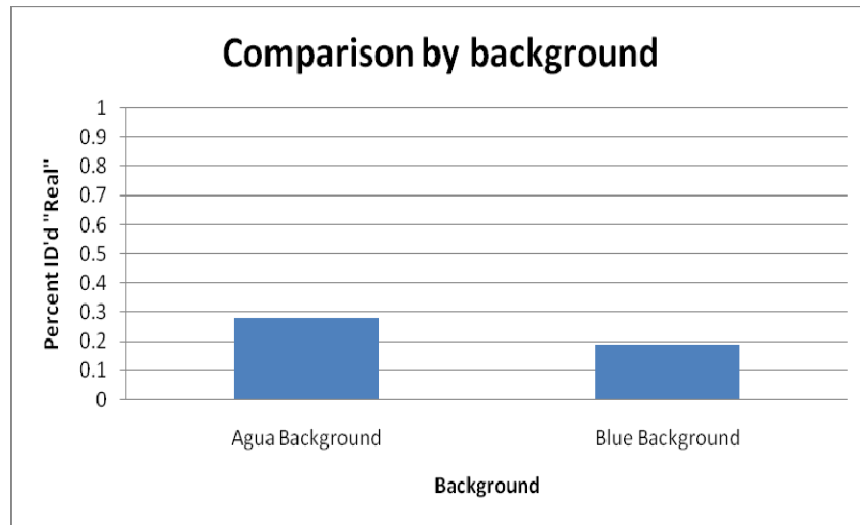


Figure 32. Percentage of "real" classifications by background

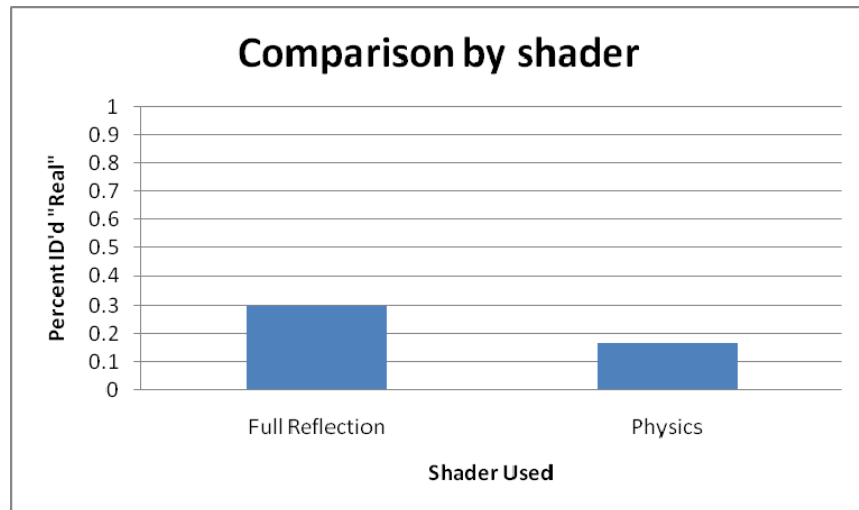


Figure 33. Percentage of "real" classifications by shader

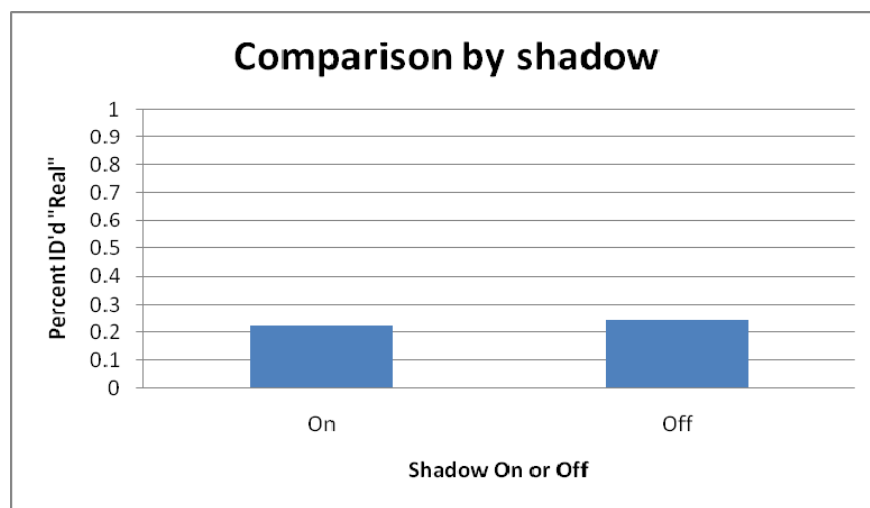


Figure 34. Percentage of "real" classifications by shadow value

V. CONCLUSIONS

A. CONCLUSIONS

1. Background Differences

The first conclusion that can be drawn from the results of this work is that the background, or more technically, the ocean characteristics plays a vital role in the realism of the reflections. With an exception of the gradient-based texture coordinate distortion function in each shader, the modifications applied to the reflections did not depend on the actual background. However, in reality the ocean surface plays a large role in determining the characteristics of the reflection. The two background images used in the user study had different sunlight and reflective properties which change how the generated reflection fits into the image. Higher sea states, different water colorings, wave direction, and lighting all influence the appearance of reflections in the real world. The Full Reflection shader takes into account sea state, while both shaders use the video water color to distort the reflections, however; neither shader takes into account any lighting conditions nor wave direction.

2. Difference in Shader Output

The marked difference in performance between the Full Reflection shader and the Physics shader can most likely be attributed to a known shader artifact: The 3D sinusoidal function was designed for smaller reflections and tends to produce regular patterns that are visually apparent in areas of larger reflections. These artifacts are generally

rings that relate to the sinusoids' test amplitude and test periodicity. These artifacts make identifying the images as software generated easier.

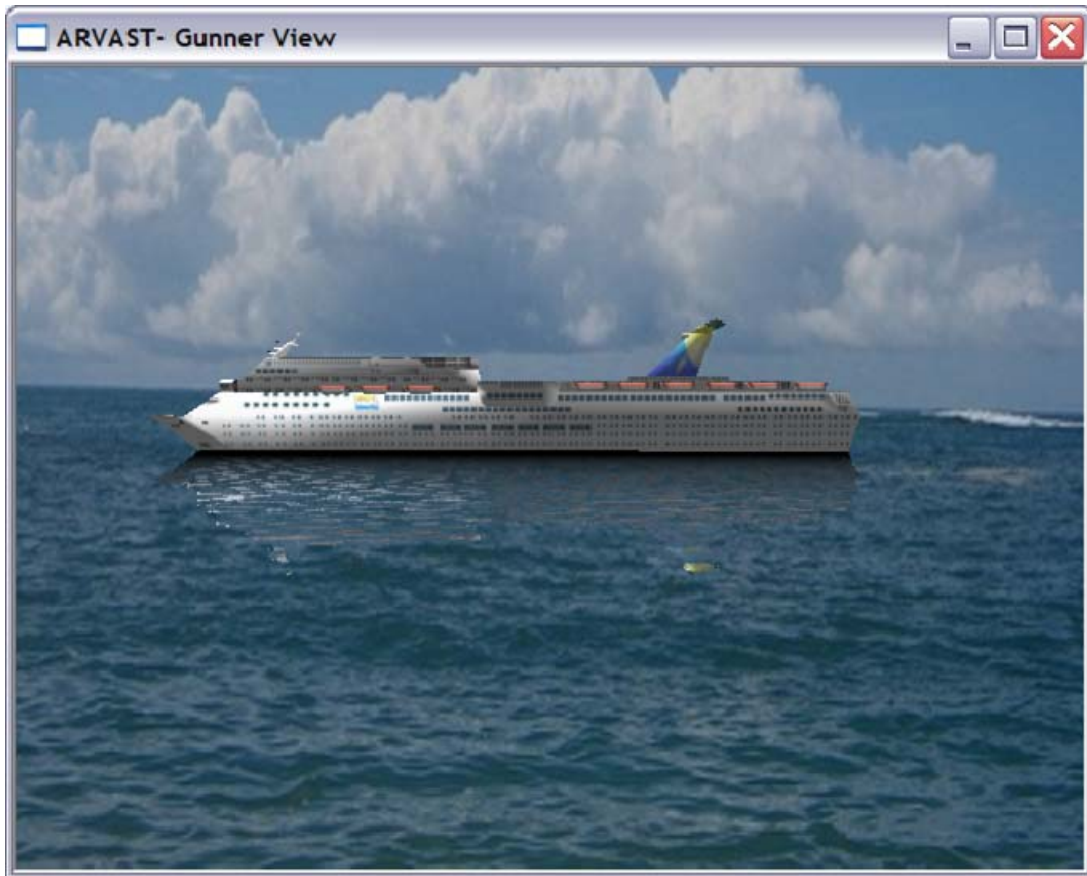


Figure 35. Smaller reflection area with physics shader



Figure 36. Larger reflection area with physics shader

B. FUTURE WORK

Our methods succeeded in producing believable images in 23% of the test cases. To further improve upon these results for the remaining user study's conditions, we suggest the following directions and approaches.

1. Computer Vision

The current methods for adaptation to the real world are mere kernel-based image processing for wave distortion. More involved computer vision techniques can improve this process through the added information gained from the real world. For example, such vision methods would allow the software to process the video feed, frame by frame, to find

horizontal lines that relate to waves. Then, Fourier transform analysis could determine the wave periodicity of the video from these horizontal lines. This wave periodicity can then parameterize the physics shader to get a more realistic wave model representation.

2. Refine Algorithms

The algorithms used throughout both shaders were developed with the end objective of visually apparent quality. The algorithms worked well, however realism may be able to be added to the reflections by fine tuning them.

The Full Reflection shader was built with ad-hoc methods that yielded a good visual result (as confirmed by the user study). We then injected more realistic physics-based models to produce the Physics shader with the hope of creating even more realistic images. The user study did not show that it met this goal. However, heuristics and ad-hoc methods will be limited in their asymptotical performance and surpassed by physics-based methods. Hence, research into improving the results from the Physics shader promises to be more promising than research into the FR shader.

C. SUMMARY

The focus of this thesis was to create realistic reflections in maritime environments in an augmented reality training system. These reflections provide a more believable and realistic contact between the virtual objects and the real world video capture.

A user study showed that even though participants are explicitly trying to spot such composed images, they are

frequently made to believe that the reflection is real. The overall goal is to improve the augmented imagery as it increases believability and immersion. This, in turn, improves the training effectiveness of systems such as AR-VAST.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- ARToolKit. Main Page. Retrieved 3 September 2009. From <http://www.hitl.washington.edu/artoolkit/>
- Conger, Nathan W. (2008). Prototype Development Of Low-Cost, Augmented Reality Trainer For Crew Service Weapons (Thesis, Naval Postgraduate School, 2008).
- GlobalSecurity. Retrieved 3 September 2009. From <http://www.globalsecurity.org/military/systems/ship/stems/vast.htm>.
- Hughes, C.E, Konttinen, J. & Pattanaik, S.N. (2004). The Future of Mixed Reality: Issues in Illumination and Shadows. Interserver/Industry Training, Simulation, and Education Conference(I/ITSEC) 2004.
- International Symposium on Mixed and Augmented Reality. Retrieved 10 August 2009. From <http://www.ismar09.org>
- Krane, Jim (2002). 'Augmented Reality' Adds New Layer to Real World. Los Angeles Times. 13 May 2002. Retrieved 3 September 2009. From <http://articles.latimes.com/2002/may/13/business/fitextra13>
- Lindberg, B.D, (2009). Panoramic Augmented Reality for Persisting Information in Counterinsurgency Environments. (Thesis, Naval Postgraduate School, 2009).
- Madsen, C.B. (2003). Using Real Shadows to Create Virtual Ones. SCIA 2003.
- Madsen, C.B., Sorensen, M.K.D., & Vittrup, M. (2003). The Importance of Shadows in Augmented Reality. Workshop on Presence.
- Meshuggah Demo and Effect browser (2009). Ocean scene. Retrieved 15 August 2009. From <http://meshuggah.4fo.de/OceanScene.htm>.

Naval Research Laboratories (2009). NRL VR Lab Finding New Ways to Enhance Flow of Information. Retrieved 3 September 2009. From <http://www.nrl.navy.mil/pao/pressRelease.php?Y=2001&R=01-01r>

Shirley, Peter (2005). Fundamentals of Computer Graphics. A.K. Peters.

Slater, M., Usoh, M., & Chrysantou, Y. (1995). The Influence of Dynamic Shadows on Presence in Immersive Virtual Environments. Proc: Virtual Environmenets.

Tiwari, A.N. (2008). Small Boat and Swarm Defense: A Gap Study, (Thesis, Naval Postgraduate School, 2008).

TurboSquid.com (2009). Ocean. Retrieved 15 August 2009. From <http://www.turbosquid.com/3d-models/ocean-waves-max/423154>

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Mathis Kölsch
Naval Postgraduate School
Monterey, California
4. John Falby
Naval Postgraduate School
Monterey, California
5. CDR Joseph Sullivan
Naval Postgraduate School
Monterey, California