

Realistic Reflections for Marine Environments in Augmented Reality Training Systems

LT Jason Nelson
Naval Postgraduate School
Monterey, CA
janelson@nps.edu

Mathias Kölsch
Naval Postgraduate School
Monterey, CA

ABSTRACT

Training systems for many emerging threats need to be highly flexible and involve complex, realistic 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 because of the ocean’s dynamics, its reflective nature, and visibility constraints such as fog and haze.

This paper presents methods for creating realistic reflections of computer-generated ships on live ocean video. After mirroring the ship, custom GPU shaders are applied to the reflection to smoothly blend it with the background ocean video. To test the quality of the composite images, a preliminary user study was conducted in which the participants had to determine the authenticity of images that were either real or automatically augmented with varying levels of fidelity. The participants started accepting the composite images as believable under certain environmental conditions, showing that our methods produce believable composites.

We built our reflection model into the Augmented Reality Virtual At-Sea Trainer. AR-VAST trains deck officers in marksmanship and procedures in maritime environments that were previously prohibitive to training, such as harbors and sea states that do not permit “robo-skis” and “killer tomatoes.”

ABOUT THE AUTHORS

LT Jason Nelson is a Modeling, Virtual Environments, and Simulation (MOVES) Masters Student at the Naval Postgraduate School in Monterey, CA. He is a Surface Warfare Officer (SWO) whose previous duties include tours as the Electrical Officer and Combat Information Center Officer aboard USS Vandegrift (FFG-48) and the Operations Officer for the High Speed Vessel Swift (HSV-2). His awards include the Navy – Marine Corps Achievement Medal for his duties on USS Vandegrift and the Navy – Marine Corps Commendation Medal for his tour on Swift. His education includes a Bachelor of Arts in Geography from the University of Illinois.

Mathias Kölsch (Ph.D., University of California, Santa Barbara, 2004) is an Assistant Professor of Computer Science at the NPS in Monterey, CA. He is also affiliated with the MOVES Institute, Chair of the MOVES Academic Committee and the Academic Associate for the MOVES curriculum. His research interests include computer vision, hand gesture recognition, augmented and virtual environments, sensor networks and mobile/embedded computing.

Realistic Reflections for Marine Environments in Augmented Reality Training Systems

LT Jason Nelson
Naval Postgraduate School
Monterey, CA
janelson@nps.edu

Mathias Kölsch
Naval Postgraduate School
Monterey, CA

INTRODUCTION

Reflections on water are an important visual cue in maritime environments. One expects to see reflections under certain water surface conditions mainly related to wind, waves and weather. When this expectation is not met in virtual or augmented reality environments, the viewer's feeling of presence gets disturbed and immersion hampered. In training environments, such loss of immersion can drastically reduce training effectiveness.

In this paper, we introduce several methods for creating realistic reflections of computer-generated ships on live ocean video. We begin with a review of related work, then detail our approach and finally present the results of a small user study demonstrated the believability of the fake reflections.

BACKGROUND

The simulation and rendering of water surfaces has advanced far in the last few years, permitting real-time visual ocean and wave modeling. Reflections on the water surface contribute significantly to the realism and can easily be achieved with hardware shaders (for a good tutorial, see (Lombard 2004)) or even toolkits such as the Aquatica Engine. However, the techniques for generating reflections in entirely computer-generated scenes do not transfer to mixed reality, or Augmented Reality (AR) systems. Augmented Reality is the combination of real world environments and virtual world objects by placing the virtual objects in footage of the real world in real time. AR systems have capabilities that make it ideal for both operational and training uses. In a purely virtual scene, the exact geometry of the water surface is known, making calculation of reflections a rather straight-forward task. In AR scenes where the water is from a real video and the boats are virtual, the geometry of the water surface is unknown.



Figure 1. Augmented Reality Ocean Reflection

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 and addressed:

- shadows (Madsen et al 2003, Supan et al 2006),
- lighting virtual objects based on real illumination (Candussi et al 2005), and
- lighting real objects based on virtual illumination (Loscos et al 2000, Hughes et al 2004).

Our work addresses the need to create realistic contact between virtual objects on real water surfaces. This involves both a reflection and a shadowing component. Our focus is on creating believable, but not necessarily accurate reflections and shadows, at a high speed and with minimal user intervention. The main obstacle to accuracy is the lack of knowledge of the water geometry and its surface properties which both determine the appearance of the reflection. This is impossible to obtain for live scenes, in contrast to some of the discussed efforts that make use of significant manual interaction to re-create virtual geometric models of the real scenes (Loscos et al 2000). Hence, we focus on reflection manipulations that look natural.

We implemented the rendering methods in the **Augmented-Reality Virtual At-Sea Trainer (AR-VAST)**. This is a deployable shipboard crew-served weapons trainer currently being developed at the Naval Postgraduate School. The system is used in conjunction with a shipboard .50 caliber machine gun. The weapon is outfit with a video camera, an inertia measurement unit (IMU) and a display device attached to the barrel. The video camera provides the attached computer with the real world imagery for the augmented world. The IMU detects the movement of the camera and supplies that information to the software. The software then uses the IMU data to move the virtual objects to keep them in the correct relative position to the real world. The display device is currently a liquid crystal display (LCD) although other configurations are possible, such as with a head-mounted display (HMD).

METHODOLOGY

Our goal is to create a reflection of virtual objects in the scene and to distort that reflection based upon information from the video background to make it appear more realistic. The scene setup is described below and schematically shown in Figure 2.

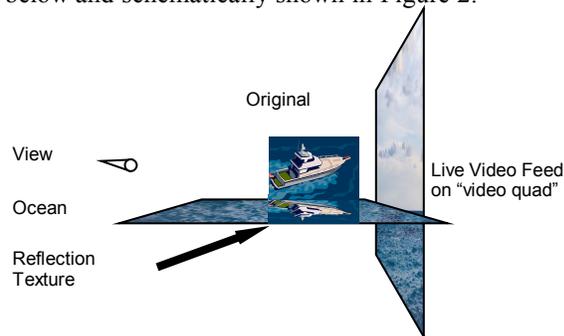


Figure 2. Augmented World Setup

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 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.

Three techniques 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.

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 $z=0$, 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, with a texture buffer object set as render target. The ocean quad is rendered with this reflection texture.

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. However, shaders do not have access to the framebuffer (which contains the video already). Therefore, the video texture is also attached to the ocean surface, providing the shader with the needed information. Since the video texture needs to be placed from a vertical surface to a horizontal surface, but still appear to the user as if it were vertical, texture coordinates are created automatically in eye-linear space. The difference to object-linear coordinates can be seen in Figure 3.



Figure 3. 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 4.

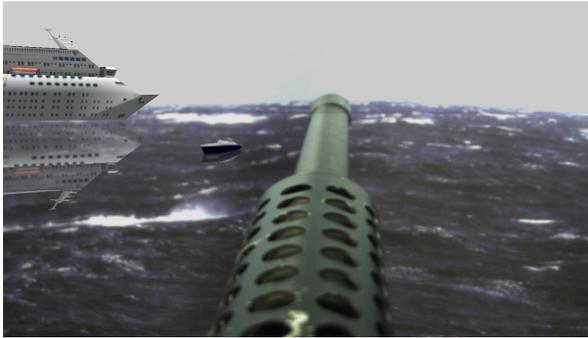


Figure 4. Scene with undistorted reflection

Reflection Distortion

The fragment shader attached to the ocean surface modifies the reflection texture coordinates to create a more realistic effect. These modification functions are based on the video texture and outlined in Figure 5. The following subsections will discuss the functions in more detail.

- Calculate local color gradients of video texture
- Add sine(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
- Calculate reflection alpha decreasing linearly from virtual object's intersection with reflection plane
- Calculate shadow effect decreasing linearly from virtual object's intersection with reflection plane
- Blur the reflection

Figure 5. Fragment Shader Algorithm

Wave Modification Function

The texture coordinates are modulated vertically and horizontally to approximate the look of reflections off waves. 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 texels immediately "below" using equation (1) from Figure 6. 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.

$$(1) \quad grad_n = \sqrt{(c_{0.r} - c_{n.r})^2 + (c_{0.g} - c_{n.g})^2 + (c_{0.b} - c_{n.b})^2}$$

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

Figure 6. Wave modification equations

The minimum *min* and maximum *max* of these nine gradients are linearly mapped to -1 and 1, respectively (see the normalization equation (2)). 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 output. As the color in the video texture changes, the texture coordinate for the reflection texture is moved left or right, 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.

Wave Effect Pass-Through Function

Another modification that is applied to the reflection texture is the pass-through of particular background wave effects. These wave effects are the extreme brightness associated with the sun's reflection on the water and the extreme darkness that occurs when the sun's light is blocked from the water. 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.

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 as 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.

Once sea state is determined, currently through user input prior to scenario run, the shader runs one of five parameterizations of equation (3) in Figure 7 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.

$$(3) \quad \alpha = \left[\begin{array}{c} -1 \\ \left(\frac{-1}{k} \right) * diff \\ k \end{array} \right] + 1.0$$

Figure 7. Calculation of reflection falloff.

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 8 and 9 show this effect.



Figure 8. Reflection with large *k*



Figure 9. Reflection with small *k*

A similar calculation is used to determine the amount of shadow applied to the reflection. 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. The implementation for shadow classification level two provides a very strong shadow immediately below the origin of the shadow and fades away very quickly. The calculation uses the equations from Figure 10. Unlike the alpha calculation for reflections, 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.

$$(4) \quad \text{proportion} = \left[\begin{array}{c} -1 \\ \left(\frac{-1}{k} \right) * diff \\ k \end{array} \right] + i$$

$$(5) \quad \text{color} = \frac{[(1 - \text{proportion}) * \text{color}]}{2}$$

Figure 10. Equations for shadow effects

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 11 shows the blurring effect on a reflection at sea state one without shadow effects.



Figure 11: Blurred reflection

EXPERIMENTS

To determine the quality of the composite images, a preliminary user study was conducted in which participants had to distinguish between photographic water reflections and our augmented reality composite reflections. Participants were shown various images of reflections and asked to assess if the reflections were from photography (“real”) or computer-generated. The reflections shown came from one of three categories:

- 1) actual photography,
- 2) computer-generated with virtual ships models, and
- 3) computer-generated with billboarded photographs of real ships.

The latter category attempts to avoid confounding effects of the rendering quality of virtual objects since we wanted to determine the reflection quality only. To that same effect, the participants were not allowed to see the entire image, just the reflection itself. Four independent variables were investigated:

- 1) textured images vs. virtual objects,
- 2) background image (two different ocean conditions, seen in Figure 16),
- 3) with and without blurring, and
- 4) with and without shadow modification.

Since this application is designed for military training and the end-user will be under some amount of stress, the participants were placed under a time constraint intended to induce a mild stress condition. The participants were given three seconds to decide if the reflection is real or not real (dependent variable). Some examples of these test images are shown in Figure 12.

The 10 study participants were 16-30 year old male and female students, some of which had prior computer (gaming) experience.

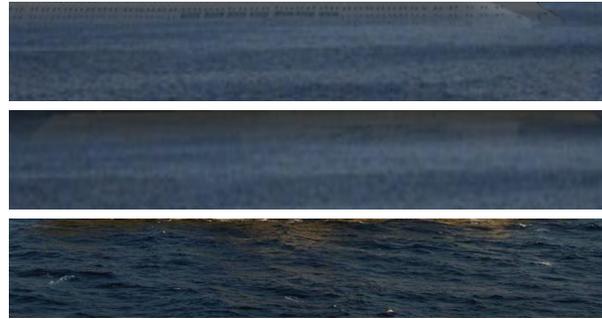


Figure 12. Sample Test Images. Reflected Virtual Object (top), Reflected Texture Object (middle), Photograph Reflection (bottom). The differences are apparent only in color.

RESULTS

Real reflection photos were correctly identified by subjects as ‘real’ 83% of the time, with a standard error of 3.9%. As expected, the results for the two classifications of composite images were lower as for the “real” category. However, more than two-third of the images of the two computer-generated image categories were identified as ‘real’ as well: 38% with standard error 4.6% for entirely virtual geometry and 41% with 3.2% standard error for textured billboards. A plot of these results can be seen in Figure 13.

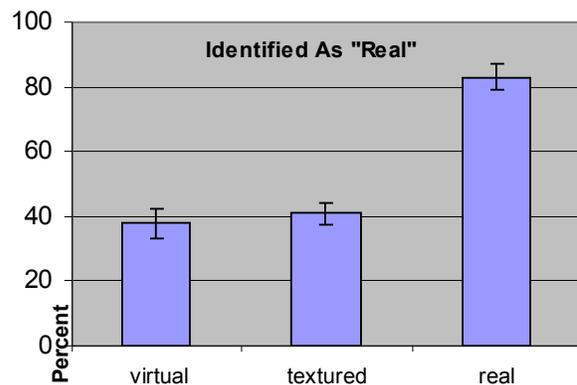


Figure 13. Percentage of images classified as “real” even though they were virtual, a textured billboard, or an actual (real) photograph of a reflection.

Three conditions within the computer-generated reflections were compared to identify their influence on the believability of the composite images. Of the three conditions background image, blurring, and shadow, only background had a noticeable effect. Background 2 was accepted as “real” for only 29% of the images (standard error, 2.2 %) while 55% of the images with background 3 were labeled “real”

(standard error 3.5 %). These results can be seen in Figure 14.

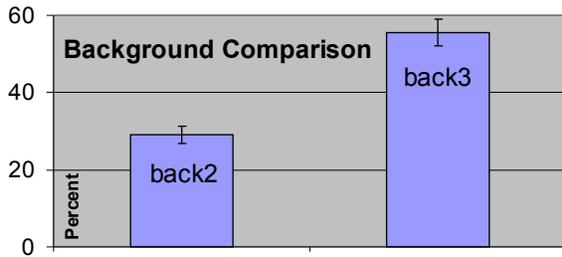


Figure 14. Percentage of images classified as "real" by background image

The other independent variables did not produce significant effects. With blurring, 41% (standard error, 3.9%) of the images were identified as "real," compared to 38% (standard error, 3.7%) without blurring. The use of the shadow effects had a similar outcome. 41% of the images with shadow (standard error, 4.5%) and 39% (standard error, 3.0 %) without shadow generation were identified as "real," respectively. These results are shown in Figure 15.

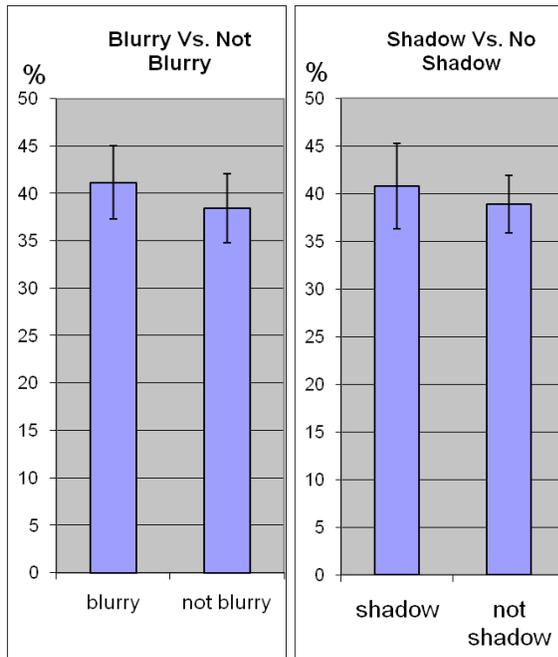


Figure 15. Effect of the blurred reflections and shadow calculations: percentage of responses indicating that the images were "real"

DISCUSSION

The results indicate that our methods are partially successful at the generation of believable reflections and that they can frequently "fool" onlookers. The results point into promising directions for further improvements. Particularly, the study's background images had a significant effect on the quality and realism of the reflections. We believe that this is due to the water being smoother in Background 3, so that an onlooker expects higher specular reflections (see Figure 16). This seems to be what our current method can produce best. Better methods are still needed for more diffuse reflections that are common at higher sea states. In addition, the photograph used for Background 3 has more highly defined wave crests than Background 2.

Blurring and shadowing did not show an effect. We believe that blurring is more important for time sequences rather than still images. Shadowing might require more sophisticated blending methods with the video texture.



Figure 16. Background Images Used in Reflection Generation

CONCLUSIONS

We presented a set of real-time methods that improve the realism and believability of augmented reality water scenes. Multiple render passes combined with a fragment shader for image analysis and custom blending create a reflection of virtual objects onto the video-see-through water surface. These methods do not require any knowledge about the ocean geometry and only require user interaction only to enter a sea state (1 through 4). A user study showed that even though participants are explicitly trying to spot such composed images, they are frequently made believe that the reflection is real.

Future work includes testing the methods on video sequences and estimating ocean state based on the video, which then influences the reflection distortion,

automatically creating realistic reflections in calm seas as well as in higher sea states. Our 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.

REFERENCES

- Candussi, A., Hollerer, T. & Candussi, N. Real-Time (2005). Rendering of Realistic Trees in Mixed Reality. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, pp. 204-205, Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'05).
- Conger, Nathan W. (2008). Prototype Development Of Low-Cost, Augmented Reality Trainer For Crew Service Weapons (Thesis, Naval Postgraduate School).
- Hughes, C.E, Konttinen, J. & Pattanaik, S.N. (2004). The Future of Mixed Reality: Issues in Illumination and Shadows. *Interservice/Industry Training, Simulation, and Education Conference(IITSEC) 2004*.
- Madsen, C.B., Sorensen, M.K.D., & Vittrup, M. (2003). The Importance of Shadows in Augmented Reality. *Workshop on Presence*.
- Slater, M., Usoh, M., & Chrysantou, Y. (1995). The Influence of Dynamic Shadows on Presence in Immersive Virtual Environments. *Proc: Virtual Environmenets*.
- Supan, P., Stuppacher, I., & Haller, M. (2006). Image Based Shadowing in Real-Time Augmented Reality. *The International Journal of Virtual Reality*. 5(3):1-7.
- Tiwari, A. (2008). Small Boat and Swarm Defense: A Gap Study, Thesis, Naval Postgraduate School.
- Yann, L., Realistic Natural Effect Rendering: Water I. Retrieved 26 June, 2009. From <http://www.gamedev.net/reference/articles/article2138.asp>.