TRANSNAV
the International Journal
on Marine Navigation
and Safety of Sea Transportation
http://www.transnav.eu
Volume 17
Number 4
December 2023
DOI: 10.12716/1001.17.04.21

Low-Fidelity Radar Implementation for Real-Time Ship Manoeuvring Simulator with Unity3D

B.G. Leite, M.A.U. Pereira Jr., E. Szilagyi & E.A. Tannuri
University of São Paulo, São Paulo, Brasil

ABSTRACT: Because of the importance of maintaining safety at sea, great training efforts are required to ensure that operators act safely in any ship. In such context, ship manoeuvring simulators are used to ease operators' learning experience. On the one hand, it may assist in the education of new operators by simulating equipment interfaces in a controlled and predictable scenario; on the other hand, it may simulate non-conventional scenarios to train advanced operators under stresses. As modelling spurious phenomena that yields marine equipment malfunctions is significantly complex, low-fidelity solutions have been proposed to the task. Likewise, the current work is concerned with the development of a low-fidelity radar module to train experienced operators under non-typical conditions. Particularly, this paper describes the radar implementation from the TPN-USP Manoeuvring Simulation Center and presents how simple additional effects may be modelled with considerable simplifications to ensure real-time performance. The implementation may be replicated in any ship manoeuvring simulator based on the game engine Unity3D.

1 INTRODUCTION

The use of simulators for maritime training and education (MET) promotes several benefits compared with conventional approaches. Regarding safety, it provides a safe environment to practice high-risk tasks (such as accident scenarios) without actual risk. It promotes more flexibility due to the possibility of simulation of different environmental conditions allowing trainees to advance at their own pace. Furthermore, the data recorded during training sessions facilitates feedbacks, discussions and the improvement of training practices [1].

It should be noted that the maritime environment is notably harsh and complex to model completely [2], which significantly complicates the development of such training simulators. Typically, 'the ability of the simulator to closely replicate the real environment' is commonly referred to as fidelity. To be cost-effective, the fidelity of a simulator should match the requirements of the situated work tasks and learning objectives. Increasing the fidelity of a simulator increases its costs but does not necessarily improve trainee learning, which is the primary goal of any training session [3].

In this context, the current work is concerned with the development of a radar module for manoeuvring training under non-conventional scenarios within the TPN-USP Manoeuvring Simulation Center. The implementation is realized with data structures from the Unity3D game engine [4]. The next chapter overviews the utilization of simulators within the MET industry and overviews the TPN-USP Manoeuvring Simulation Center. Chapter 3 describes the radar implementation using Unity3D. Chapter 4 presents the proposed additional effects that are

implemented to simulate non-typical scenarios. Chapter 5 discusses potential training scenarios, and the last chapter presents the conclusions from the current work.

2 TRAINING SIMULATORS

Simulator training typically involves three sequential phases: briefing, scenario and debriefing. Initially, in the briefing phase, an instructor introduces the context of the training assignment to the trainees focusing on the scenario and learning goals. Next, the simulator plays the scenario, and the trainees may interact with the system towards their learning goals [5]. In the end, a debriefing phase is carried out to analyse the overall experience of participants [6].

The use of simulators within the MET industry is regulated by the International Convention on Standards of Training, Certification and Watchkeeping for seafarers (STCW) [7]. Specifically, in section A-I/12 of the code, it states that any simulator used for mandatory simulator-based training should [7]:
"be suitable for the selected objectives and training tasks; be capable of simulating the operating capabilities of shipboard equipment concerned, to a level of physical realism appropriate to training objectives, and include the capabilities, limitations and possible errors of such equipment; have sufficient behavioural realism to allow a trainee to acquire the skills appropriate to the training objectives; provide a controlled operating environment, capable of producing a variety of conditions, which may include emergency, hazardous or unusual situations relevant to the training objectives; permit an instructor to control, monitor and record exercises for the effective debriefing of trainees"

The TPN-USP Manoeuvring Simulation Center is used to evaluate new ports and operations, risk analysis, pilots and captains training. The center has 6 cabins, in which of them, 3 are classified as full-mission. The simulation can be executed in single or multiplayer mode. Its visualization system aims to immerse the pilot into a realistic virtual environment, providing a high-definition visual information to make an operation decision [8]. The implementation is realized across different modules running on a network. The diagram shown in Fig. 1 overviews the several modules from the TPN-USP Manoeuvring Simulator. The main full-mission simulator of the TPN-USP Ship Manoeuvring Center is shown in the Fig. 2.

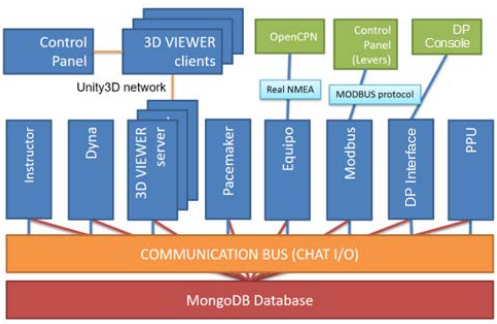

Figure 1. Modules from the TPN-USP simulator. Source: Authors

Figure 2. TPN-USP main full mission simulator. Source: Authors

The MongoDB database module stores 3D models of areas, vessels and typical nautical information of the scenarios. The Instructor module is the main interface for initializing, controlling and ending training sessions. The vessel position and orientation within the environment is updated in the Dyna component, which is based on a low-speed mathematical model developed by the University of São Paulo, with the support of Petrobras (Brazilian oil state company) and technical collaboration of Brazilian Pilots Association (CONAPRA) [9].

The navigational scenario is presented as a realistic virtual environment with a Camera component from the Unity3D game engine [4]. Each object is instantiated as a GameObject with a Transform component describing its position and orientation in the virtual environment. In this software framework, the visual appearance of objects are defined by meshes representing their shapes and materials describing the appearance of their surfaces. Meshes contain an array of vertices and an array of surfaces that indicates which vertices are interconnected. The visual appearance of surfaces is defined by a Material class. Materials contain references to Shader programs that compute the color of pixels on the screen. This process, typically referred as rendering, is one of the most resource-intensive operations within a simulation. The series of operations to achieve full rendering of the scenario is called render pipeline.

Generally, there are three main operations in a render pipeline. The first operation is called culling, which refers to the determination of the elements from the scene that are within the field of view of the camera. Next, the rendering operation occurs, which determines the visual appearance of objects from the scene accordingly to their geometries, materials and lighting conditions. The last step consists of the post-processing operation, a generic term to represent full-screen image processing effects that may be applied to the image after the rendering operation. The current work uses the default render pipeline from Unity3D (Built-in Render Pipeline) as it provides an optimized pipeline for extracting scene information within the Unity3D simulation.

3 IDEAL RADAR IMPLEMENTATION

The Radar is implemented with a Camera component fixed in the vessel within the environment. The Camera continuously spins, performing a horizontal scan. GameObjects from the scenario have their Shader programs changed in runtime to efficiently render a Normal&Depth Image of the scene. Detections are simulated from the depth information of all visible surfaces parallel to the image plane. These detections are rendered in a user interface that mimics a real equipment to enable trainees' interaction in real-time.

3.1 *Normal&Depth Image from Camera*

Typical representations for images in the Unity3D game engine use a 4-dimensional matrix to store color information. The first three channels store information regarding primary colors red, green and blue; the last channel store information about pixel transparency. Each pixel location in the image corresponds to a different direction with respect to the Camera coordinate system. This data structure is used to represent all surfaces within the field of view from the Camera component. A customized Shader program has been developed to perform this rendering. For each Mesh from each object of the scene, the normal direction of its surfaces is represented with colors from the first three channels of the image (red, green and blue) and the distance to each surface is represented with the last channel (transparency). Fig. 3 shows an example of a Normal&Depth Image (NDI) from the scene.

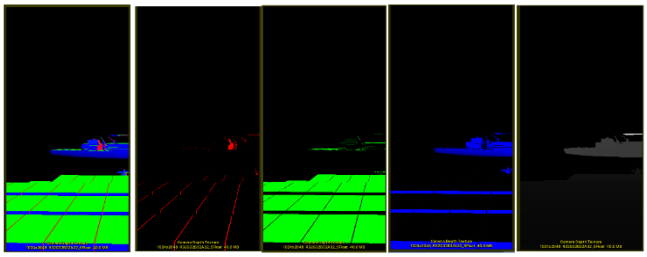

Figure 3. Normal&Depth Image (NDI) from Unity3D scene. Source: Authors

Note that the central part of the image corresponds to points that are in front of the Camera. Assume a radio signal propagating from the Camera component towards the Unity3D scenario in this direction. The signal will be reflected back to the Camera component for surfaces that are parallel to the image plane. Therefore, for all points located in the central part of the image, if their surfaces are sufficiently parallel to the Camera image plane (i.e., their blue channel value is higher than a threshold), then their depth information are sampled as radar detections in that direction. Fig. 4 exemplifies the process of simulating a set of directional radar detections from a Normal&Depth Image (NDI).

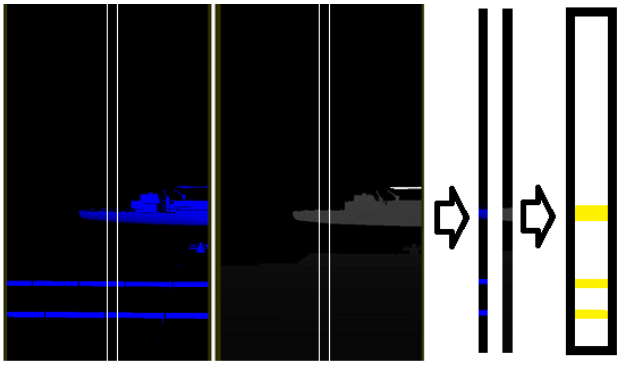

Figure 4. Directional radar detections from Normal&Depth Image (NDI). Source: Authors

The set of directional radar detections is represented in Fig. 4 at the right. In order to simulate detections in all directions, the *Camera* component continuously rotates.

3.2 *Horizontal Scan*

Consider an angular resolution θ_x. The horizontal scan of the Antenna is implemented with successive rotations of the same magnitude θ_x. Let N_{rot} be the number of successive rotations to complete a full turn of 360 degrees:

$$N_{rot} = \frac{360}{\theta_x}$$

Let ΔT_{scan} be the scan period and ΔT_{simul} be the time interval between two consecutive simulation iterations. Considering $\Delta T_{simul} \ll \Delta T_{scan}$, the number of successive rotations between two iterations of the simulator is smaller than N_{rot}. Let n_{rot} be the number of successive θ_x rotations between time interval ΔT_{simul}:

$$n_{rot} = \frac{\Delta T_{simul}}{\Delta T_{scan}} N_{rot}$$

Fig. 5 illustrates the Camera rotation during successive iterations of the simulation.

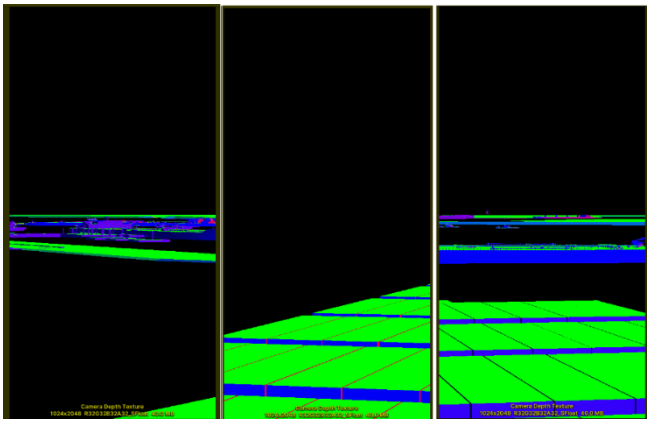

Figure 5. Normal&Depth Image from rotating Camera component. Source: Authors

Measurements from each direction are stored in a histogram image with width W_{hist} and height H_{hist}. The

image width W_{hist} corresponds to the number N_{rot} of successive θ_x rotations to complete a full turn. As illustrated in Fig. 6, each pixel in the horizontal direction (u) corresponds to a different angle θ and in the vertical direction (v) corresponds to a different distance R.

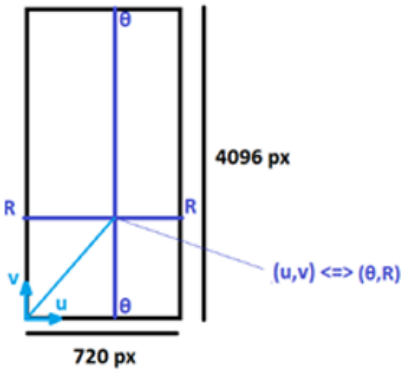

Figure 6. Schematic of histogram image to store detections from Normal&Depth Image. Source: Authors

Fig. 7 presents an example of a histogram image from a navigation scenario.

Figure 7. Example of a histogram image from a navigation scene. Source: Authors

The histogram image is continuously updated while the vessel navigates through the environment. The simulated measurements are visualized in an interface view with buttons to enable the trainee's real-time interaction.

3.3 *Graphical User Interface*

The histogram image (illustrated in Fig. 7) is converted from polar coordinates to a cartesian map and is displayed in a Raw Image component from the Unity3D engine. Fig. 8 exemplifies full radar measurements displayed with the Radar at the center and a predefined resolution.

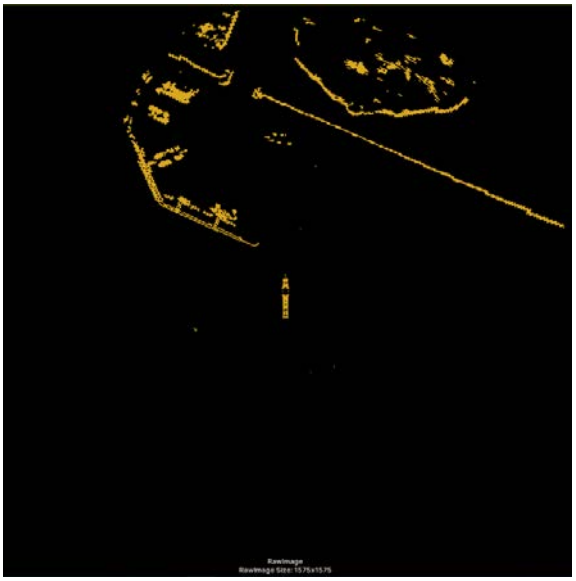

Figure 8. Raw Image with full measurements from a navigation scene. Source: Authors

The Raw Image containing the radar detections is overlaid with a graphical user interface (GUI) as illustrated in Fig. 9.

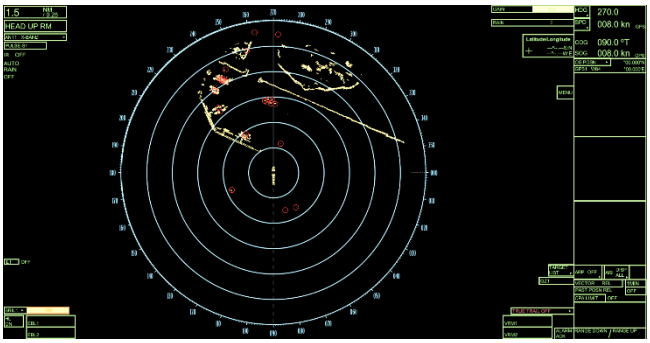

Figure 9. Radar interface with full measurements and control buttons. Source: Authors

An additional *Camera* component is used to present the interface to users. Note that if the interface is similar to a real equipment interface, knowledge acquired by the trainees during simulation is more easily transferable to real situations. Therefore, the graphical user interface (GUI) has been designed to mimic a real equipment interface [10]. A full description of features and buttons functionality is not within the scope of the present work. Generally, the important functions to be simulated in the Radar within a training section will depend on the set of skills being trained.

Note that it is possible to introduce additional artifacts into radar measurements through the histogram image. Such artefacts may be used to simulate non-conventional scenarios with low-fidelity.

4 ADDITIONAL ARTEFACTS

Consider a simulation with uniform rain in the navigation area. Rain artefacts may be introduced directly into the histogram image to avoid unnecessary computations in the render pipeline of

the radar scan. Let I_{rain} be a scalar denoting the intensity of the rain, where I_{rain}=0 corresponds to a scenario with full visibility without rain and I_{rain}=1 corresponds to a scenario with maximum rain-intensity. Simplified rain artefacts may be introduced by inserting detections in the histogram image with probability proportional to I_{rain}. Fig. 10 presents a schematic representation of the proposed procedure.

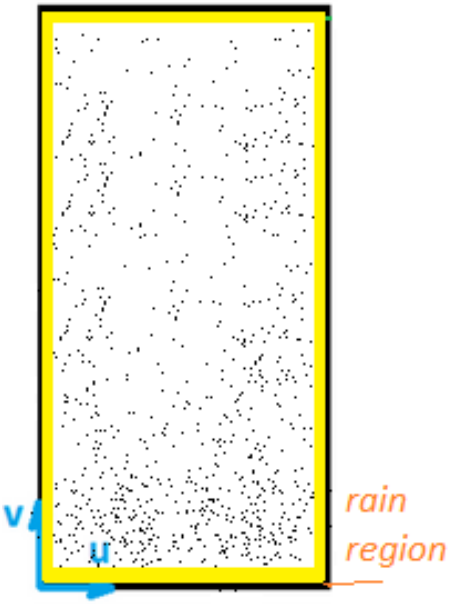

Figure 10. Insertion of rain artefacts in the schematic histogram image. Source: Authors

Occlusion of far surfaces is described by an additional parameter D_{occl} that specifies the distance of occlusion due to the rain. The parameter D_{occl} is inversely proportional to I_{rain} and is interpolated by two references D_{min} and D_{max}. All points with a distance superior to D_{occl} are removed from the histogram image as illustrated by Fig. 11.

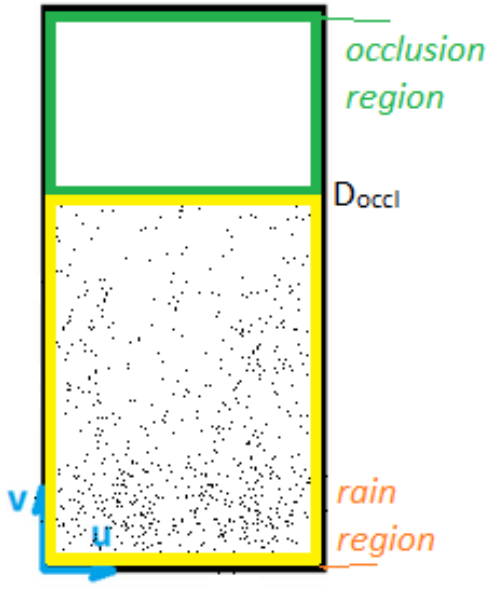

Figure 11. Occlusion region due to the rain in the schematic histogram image. Source: Authors

Fig. 12 presents the radar measurements affected by the proposed rain artefacts with I_{rain} =0.1.

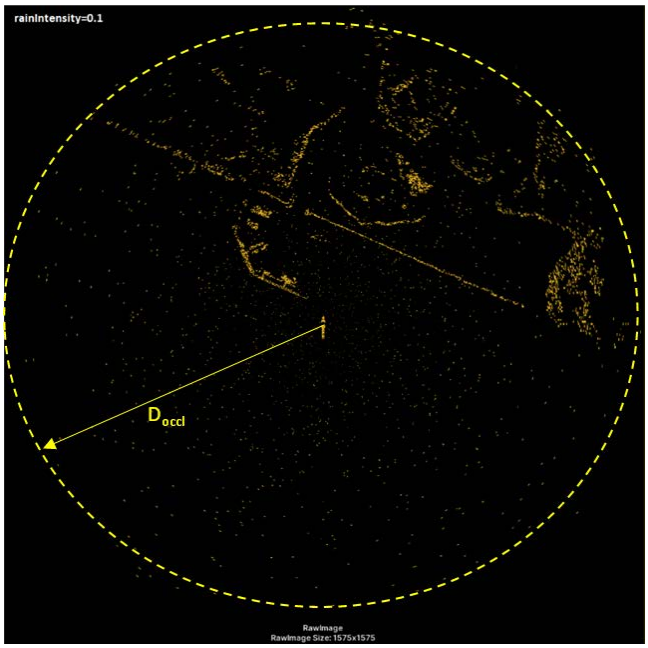

Figure 12. Measurements from navigation scene with I_{rain} =0.1. Source: Authors

Similarly, Fig. 13 presents the radar measurements affected by the proposed rain artefacts with I_{rain} =0.9.

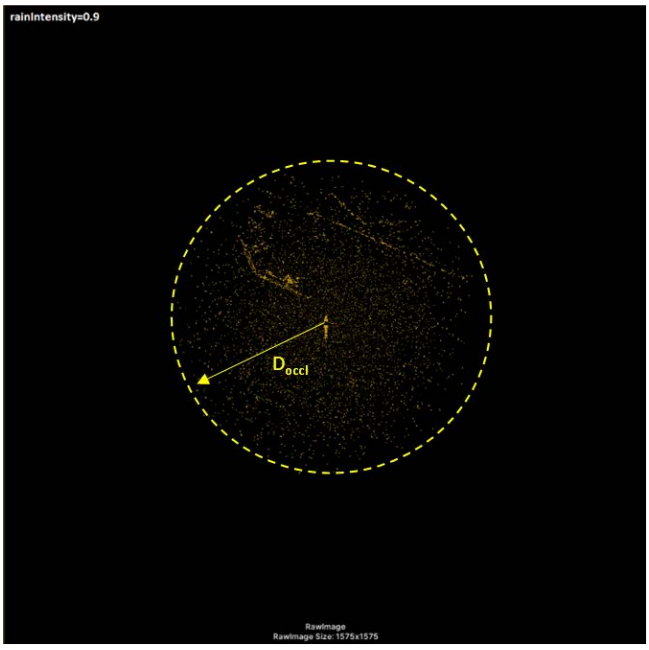

Figure 13. Measurements from navigation scene with I_{rain} =0.9. Source: Authors

Occasionally, in commercial marine radars, it is possible that false echoes appear on the screen at positions where there is no target. An example of such false echoes occurs when the Radar is operating in the vicinity of big metallic surfaces. Besides the true echo reflected from the surface, a second or more echoes may be observed at multiples of the true distance in the radar display.

Multiple echoes artefacts may be introduced with low-fidelity by duplicating original detections of the histogram image after the horizontal scan of the

simulated Radar. As illustrated by Fig. 14, for a surface detected by the Radar mapped in the histogram image at pixel position (u,v), it is possible to duplicate its detection by copying the pixel value to the position $(u,2v)$.

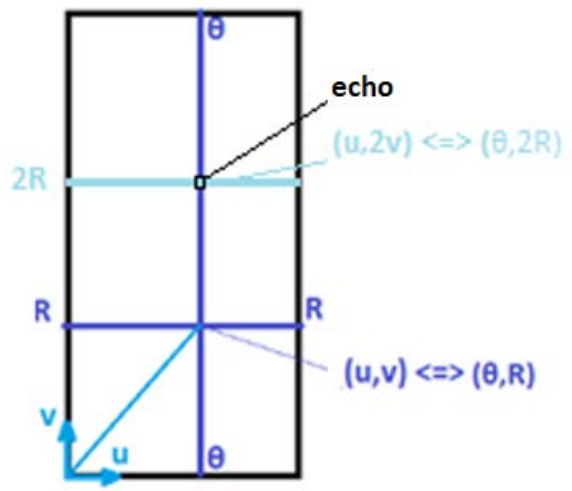

Figure 14. Multiple echo detection insertion in histogram image. Source: Authors

The Shader programs that render nautical buoys and other ships are modified to trigger multiple echo detections. A minimum distance D_{echo} is defined to determine the range in which such echoes should be inserted. Fig. 15 shows a second echo from a buoy at the right of the Radar. Note that the other objects that could trigger multiple echo detections are not within the range defined by D_{echo}.

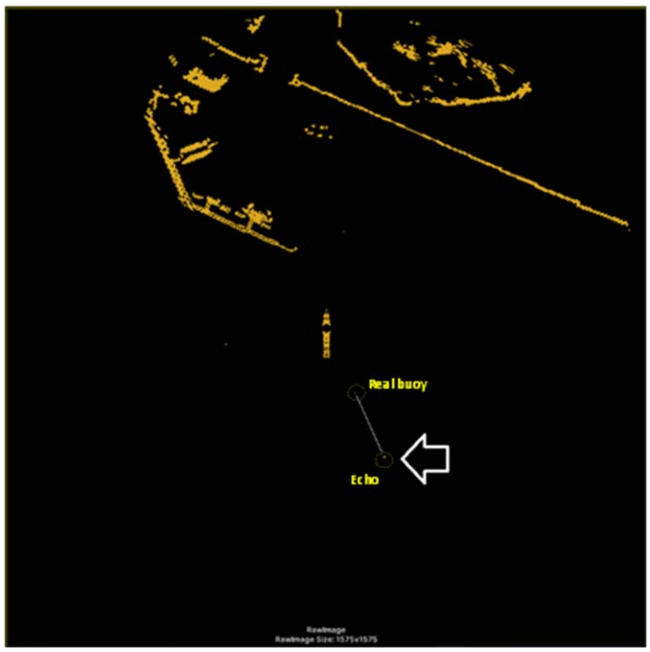

Figure 15. Measurements from navigation scene with multiple echo detection at right. Source: Authors

Another type of false echoes that may be easily inserted into the radar measurements are curved lines. Consider a line in the histogram image that begins at distance R_{min} from the center of the radar output ending at a distance R_{max}. Further, let θ_0 be the initial angle of the curved line and $\Delta\theta$ be its angular length. As illustrated by Fig. 16, this line in the histogram image is mapped into a curved line in the radar output.

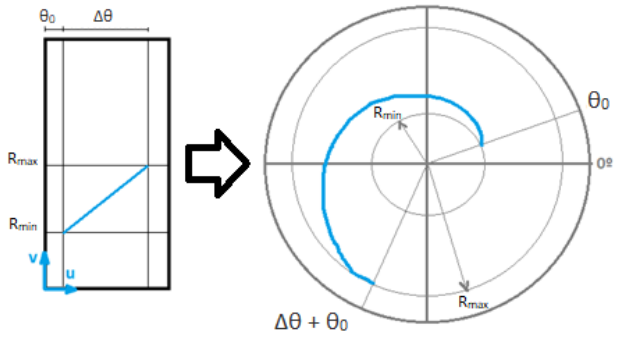

Figure 16. Mapping of lines from histogram image to curved lines. Source: Authors

It is possible to create interesting patterns into the radar measurements by the repetition of several curved lines. Fig. 17 presents different set of curved lines generated with different $\Delta\theta$. The proposed artefacts may be used to simulate interference scenarios with low-fidelity.

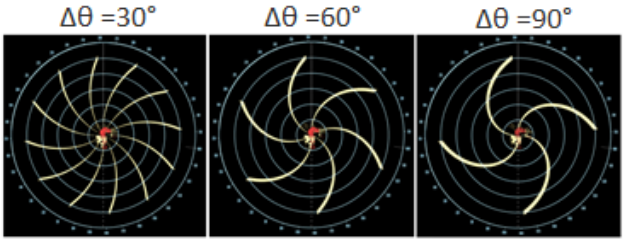

Figure 17. Curved lines for different $\Delta\theta$. Source: Authors

5 DISCUSSION

The proposed radar implementation has been developed to assist in manoeuvring training at the TPN-USP Manoeuvring Simulation Center. For this use case, the most important requirement is the accurate hydrodynamic modelling of the vessel and its interaction with the scenario. Regarding the radar simulation within this case, detections should be adequately positioned in the radar display accordingly to its visual representation in the environment. As the radar implementation is realized with Unity3D classes from the same execution context as the rendering of the visual environment, the correct association between the virtual environment and the radar measurements is ensured.

Another important requirement refers to the immersion that trainees feel during simulation. In this context, the simulator mimics the dependencies of the bridge from a ship. Fig. 18 presents the frontal view of a simulator from the TPN-USP Manoeuvring Simulation Center.

Figure 18. Frontal view of the main full mission simulator from the TPN-USP Manoeuvring Simulation Center. Source: Authors

The setup illustrated in Fig. 18 may be used to assist the training of operators in their spatial-localization skills. The identification of correspondences between visual obstacles at the environment and radar measurements at the display is an important task that may be easily reproduced with the proposed radar implementation.

Moreover, as mentioned earlier, the use of simulators within the MET industry is formally regulated by the International Convention on Standards of Training, Certification and Watchkeeping for seafarers (STCW). Specifically, in the section A-I/12 of the code, it states that radar simulation equipment should incorporate facilities to [7]:

"operate in both sea and ground stabilized relative motion and true motion modes; model weather, tidal streams, current, shadow sectors, spurious echoes and other propagation effects, and generate coastlines, navigational buoys and search and rescue transponders; create a real-time operating environment incorporating at least two own ship stations with ability to change own ship's course and speed, and include parameters for at least 20 target ships and appropriate communication facilities"

A full description on the fulfilment of STCW requirements for radar simulation equipment is not within the scope of the present paper. Still, it is interesting to note that the described radar implementation provides a relative motion mode of operation in real-time. Shadow sectors are implicitly incorporated into the radar with the culling operation from the Unity3D default render pipeline. Regarding weather effects, the proposed rain artefacts enable the simulation of simple rain scenarios with low-fidelity. The simulation of a simple set of low-fidelity spurious echoes effects are also described.

6 CONCLUSION

We described an implementation for a low-fidelity Radar with software components from the Unity3D game engine. The use of this game engine facilitates the integration of radar measurements with the training simulation and facilitates the correct association between the virtual environment and the radar measurements. The implementation is built from normal and depth information of the navigational scene, which is rendered as a realistic virtual environment, and may be replicated in any ship manoeuvring simulator that is based on this game engine. Furthermore, additional effects described in the current work enables the simulation of non-typical scenarios in accordance with most of the STCW requirements.

REFERENCES

[1] T. Kim, A. Sharma, M. Bustgaard et al. "The continuum of simulator-based maritime training and education." WMU J Marit Affairs 20, 135–150 (2021). https://doi.org/10.1007/s13437-021-00242-2

[2] R. Iversen "The mental health of seafarers." Int Marit Health. 2012;63(2):78-89. PMID: 22972547.

[3] M. Hontvedt, "Professional vision in simulated environments — Examining professional maritime pilots' performance of work tasks in a full-mission ship simulator", Learning, Culture and Social Interaction, Volume 7, 2015, Pages 71-84, ISSN 2210-6561, https://doi.org/10.1016/j.lcsi.2015.07.003.

[4] "Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine." Unity® software. https://unity.com/ (accessed Feb. 28, 2023).

[5] C. Sellberg "From briefing, through scenario, to debriefing: the maritime instructor's work during simulator-based training." Cogn Tech Work 20, 49–62 (2018). https://doi.org/10.1007/s10111-017-0446-y

[6] M. Hontvedt, H.C. Arnseth, "On the bridge to learn: Analysing the social organization of nautical instruction in a ship simulator." Computer Supported Learning 8, 89–112 (2013). https://doi.org/10.1007/s11412-013-9166-3

[7] International Maritime Organisation (IMO), "International Convention on Standards of Training, Certification and Watchkeeping for Seafarers, (STCW) 1978, as amended in 1995/2010." (2011). London, UK

[8] H. Makiyama, E. Szilagyi, G. Pereira, L. Alves, B. Kodama, D. Taniguchi E. Tannuri (2021). "Computational Graphics and Immersive Technologies Applied to a Ship Maneuvering Simulator." In: Cheng, LY. (eds) ICGG 2020 - Proceedings of the 19th International Conference on Geometry and Graphics. ICGG 2021. Advances in Intelligent Systems and Computing, vol 1296. Springer, Cham. https://doi.org/10.1007/978-3-030-63403-2_56

[9] E. A. Tannuri, F. Rateiro, C. H. Fucatu, M. D. Ferreira, I. Q. Masetti, and K. Nishimoto, "Modular Mathematical Model for a Low-Speed Maneuvering Simulator." Proceedings of the ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering. Volume 1B: Offshore Technology. San Francisco, California, USA. June 8–13, 2014. V01BT01A036. ASME. https://doi.org/10.1115/OMAE2014-24414

[10] Furuno - Operator's Manual: MARINE RADAR/ARPA FAR-2xx7. Furuno Electric CO., LTD. Accessed: Feb. 28, 2023. [Online]. Available: https://www.furunousa.com/-/media/sites/furuno/document_library/documents/manuals/public_manuals/far2xx7_operators_manual.pdf