Original Article

Institution of MECHANICAL ENGINEERS

Journal of ENGINEERING FOR THE MARITIME ENVIRONMENT

Proc IMechE Part M:
J Engineering for the Maritime Environment
1–13

DOI: 10.1177/1475090216642470
pim.sagepub.com
SAGE

Geometry and visual realism of ship models for digital ship bridge simulators

Jose Miguel Varela and C. Guedes Soares

Abstract

This article addresses the main requirements and the process of creating the geometry of ship models that fulfil the highly demanding request for realism and performance of the virtual environments currently used in modern ship bridge simulators. It starts with a classification of the ships based on their role in the simulation and on the type of simulator used, and defines the main characteristics of the models. It also discusses the importance of a well-defined workflow and its impact on the modelling time and on the quality of the final product. The article provides contributions in the following areas: identification of the main requirements of polygonal models of ships for ship simulators; effective workflow for ship three-dimensional modelling and identification of most suitable modelling techniques for efficient creation of ship models. The study is supported by real examples of three-dimensional modelling of ships with different sizes and characteristics currently used by the ship manoeuvring simulator in the Centre for Marine Technology and Ocean Engineering of the University of Lisbon.

Keywords

Three-dimensional modelling, low poly, levels of detail, ship simulation, texturing

Date received: 10 October 2015; accepted: 7 March 2016

Introduction

For some time now, ship bridge simulators for training deck officers and masters have been an indispensable tool to assure the safety of maritime navigation, as mentioned in Ali.[1] As referred by Bowman and McMahan,[2] the importance of the graphics quality of the virtual environment (VE) in such applications is a fundamental factor to raise the level of immersion and presence experienced by the users, increasing the effectiveness of the application as a training tool. On the basis of the graphical environment are the polygonal models of ships and/or other floating structures. Therefore, fast and efficient polygonal modelling of three-dimensional (3D) digital ships is an essential task to produce modern, commercially viable ship simulators. Being quite complex 3D models, containing many objects with different shapes, textures and materials, ships can take excessive time and effort to model if an adequate workflow and modelling techniques are not followed from the beginning.

This article addresses the creation of 3D polygonal models of ships for modern ship bridge simulators. We assume that the specific models' requirements, the most important visual details, the way the techniques are applied and the final usage of the model are factors that can make the difference in a successful 3D modelling workflow. Therefore, we consider these factors as our object of study and improvement.

Concerning the different types of existing ship simulators, we start by defining a classification for the different ship types according to their role and participation in the simulation scenario. From this classification, we evaluate the main characteristics of their polygonal meshes. We then present the modelling techniques applied at each stage of the workflow and highlight the most significant details according to the type of ship being modelled. Ship models have some particular characteristics that differentiate them from other vehicle models, such as cars or even airplanes, when 3D modelling and simulation are concerned. For instance, the

Centre for Marine Technology and Ocean Engineering, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

Corresponding author:
Carlos Guedes Soares, Centre for Marine Technology and Ocean Engineering, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal.
Email: c.guedes.soares@centec.tecnico.ulisboa.pt

Table 1. Ship classification according to its role in the simulation and proximity to the observer.

Level of participation	Role in the simulation	Polygonal model detail		
		Ship exterior	Ship deck	Ship bridge
Primary ship	Main participant in the operation Controlled by the user	Not modelled	Medium/high	High
Secondary ship	Active participant in the operation Controlled by the system or by an instructor	Medium/high	Medium/high	Not modelled
Extra ship	Does not participate in the operation Controlled by the system	Low/medium	Low/medium	Not modelled

geometry of the hull to compute ship motions must obey specific requirements depending on the calculation method, while for collision detection or visualization the hull may have completely different requirements.

We assume that simulation systems are compatible with the latest real-time rendering capabilities of graphics engines. Naturally, it is impossible to address all modelling aspects in detail, in such a small extent such as this article. Therefore, we highlight mostly the differences and specific aspects of ship polygonal models taking into consideration their role in the ship simulator. Our contribution is a specification of the requirements, workflow and modelling techniques of ship geometric models for professional ship simulators. The study uses the models of a liquefied natural gas (LNG) tanker and a tugboat as the main references, which are currently used in the CENTEC ship manoeuvring simulator described in Varela and Guedes Soares.[3] Although modelling software and hardware capabilities are continuously changing and expanding, the specification and techniques presented are to some extent independent of these factors and we believe that they will remain unchanged for the next few years.

Model requirements

Ship classification

In general, the requirements of polygonal ship models depend mainly on their role in the simulation. For the classification of ship models, we rely on existing online information about professional ship bridge simulators such as the SimFlex4 and the K-Sim® Navigation developed by the FORCE Technology[4] and by the Kongsberg,[5] respectively, on our experience with Nautis Simulator developed by VStep,[6] and on our knowledge in developing ship bridge simulators.[3] We define three different roles of the ship in the simulation, which influence the characteristics of the final model: the primary ship, the secondary ship and the extra ship.

Primary ships correspond to the main actors in the simulation. They are controlled by the user, normally located at the ship bridge, where she/he spends most of the time during the simulation. Maritime operations involving more than one ship, such as the typical case of towing, may have two or more primary ships if they are controlled by different users located at their ship bridges.

Secondary ships are also involved in the operation, but they are controlled by the system or by external instructors. Typical examples are the assisted ships, either in towing, rescue or fire-fighting operations. As secondary ships participate actively in the operation and are observed closely, their geometry must include all the necessary details and equipment to provide the required level of realism for the simulation.

Extra ships do not participate actively in the operation. However, their presence in the scenario normally goes beyond visualization purposes. They may be considered for collision, path obstruction or scenario occlusion, similarly to what happens in a real scenario. In a towing operation inside a port or channel, extra ships may be, for instance, other ships in transit limiting the operational ability of the tugboat and assisted ship.

Table 1 presents our classification regarding the detail of the polygonal model. Within the scope of this article, we divide the model of the ship into the exterior and interior parts. The exterior part is then subdivided into the hull surface, the main deck and the superstructures (excluding the internal arrangement). The ship bridge includes the interior structures and equipment, such as the interior walls, floors, windows and instruments. Special attention should be given to the instruments' readouts, which may be included in the virtual model itself or may be given in separate viewports. In both options, the readouts must be fully visible and perceptible.

Target ship simulators

Concerning only interactive simulators, Hensen[7] mentions the existence of the five types of ship manoeuvring simulators presented in Figure 1: radar simulators, micro-simulators, mini-simulators, full-mission bridge simulators and virtual reality (VR)-based simulators.

The type of simulator defines the characteristics of the primary ship models, mainly due to the different visual display approaches. Secondary and extra ship models are normally independent of the type of simulator, and their characteristics (detail and complexity) may depend on the available computational power and modelling schedules for the specific projects.

Figure 1. Main types of target ship bridge simulators.

Table 2. Main types of ship simulation systems for training.

Simulator type	Visual display	Primary ship polygonal model
Radar simulator	Radar screen	Symbolic representation of the ship model
Micro-simulator	Bird's eye view Synthetic radar view Panoramic view as seen from the user's eyes Ship instruments with the most important readouts	Exterior with medium detail for bird's eye view purposes Deck with medium detail for panoramic view Bridge interior with high detail including all instruments Readouts may be displayed in separate viewports
Mini-simulator	Panoramic view as seen from the windows of the wheelhouse	Exterior not modelled Deck with high detail for panoramic view Bridge is not modelled Readouts displayed by physical instruments
Full-mission bridge simulator	Full-scale display of the ship and surrounding area as seen through the windows of the wheelhouse Full 360° field of view for some types of ships	Exterior not modelled Deck with high detail for panoramic view Bridge is not modelled Readouts displayed by physical instruments
Virtual reality-based simulator	Panoramic view as seen from user's eyes Ship's instruments with all readouts Full 360° field of view for all scenarios	Exterior with medium to high detail Deck with high detail Bridge interior with high detail including all instruments and readouts

In Table 2, we define the main characteristics of the polygonal model of the primary ship, depending on the type of simulator, namely, the nature of the visual display, and concerning which features should be modelled. It can be seen that polygonal ship models like the ones addressed in this work are not applicable to radar simulators.

The micro-simulator and the VR simulator are the most demanding ones in terms of achieving a balance between graphics quality and system performance. These two simulators do not have the support of physical mock-ups of the ship's bridge, and therefore require additional elements to be modelled, such as the interior walls, windows and instruments. Moreover, the hardware normally used for these two types of simulators is not as powerful as the one used for mini- or full-mission simulators, due to portability or financial reasons. The micro-simulator normally allows a bird's eye view of the primary model (refer to the tugboat of the micro-simulator in Figure 1), and therefore the ship exterior must also be modelled. For the mini- and full-mission bridge simulators, often only the ship deck is modelled for visualization purposes, because the operator is always located at the ship bridge and most of the ship exterior is not visible. As a consequence, the ship deck should be modelled with high detail for the panoramic view.

VR simulators are the only ones that have the potential to achieve a level of immersion close to the one exhibited by full-mission bridge simulators. However, in this case, the VE must be completely generated by the computer, without physical mock-ups, and the level of immersion relies largely on the graphical realism of the models. In fact, physical instruments also exist in this type of simulators, as shown in Figure 1. However, they must be also modelled in the VE to maintain the consistency of the scenario for all possible orientations of the head mounted display (HMD). Therefore, this type of simulator should only be considered if highly detailed graphical models of the ship exterior and bridge can be built. Although they may have enormous training potential, VR simulators with HMDs are still not used for professional training of deck officers and masters as an alternative to full-mission simulators.

Hull geometry

The hull, in this case, is a particular object/geometry of floating structures, and special attention should be

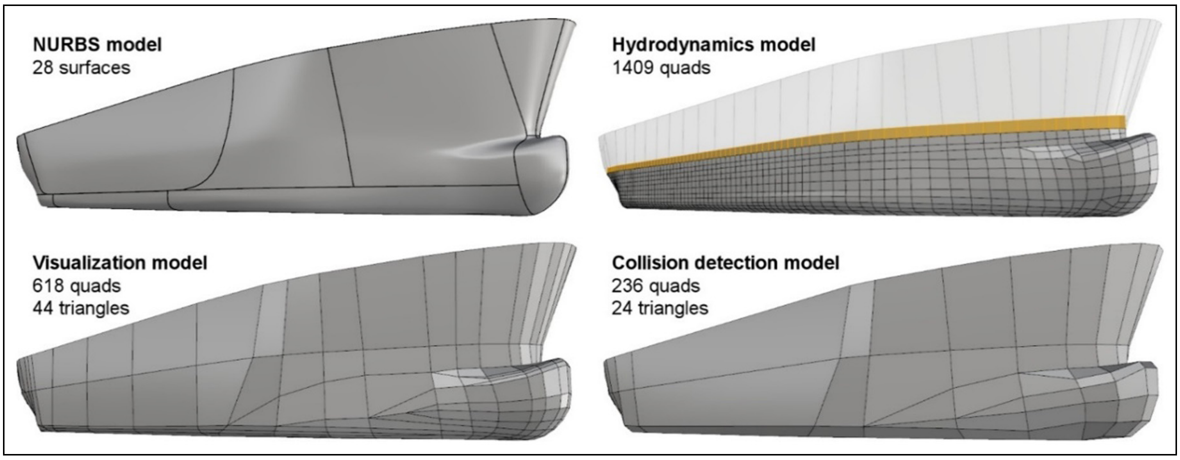

Figure 2. Geometric models of the ship hull.

taken when modelling this virtual object. Beyond being a significant visible part of the ship, the hull shape is also used for the computation of the ship motions, manoeuvrability and collision detection.

The physical models used to compute the ship motions dictate the approach taken for the 3D modelling of the hull. Professional ship bridge simulators normally use complex hydrodynamic equations that require the knowledge of ship parameters. In numerical simulations, ship motions are normally defined in frequency domain by computing the response amplitude operators (RAOs) of the ship for a set of different regular unidirectional waves and manoeuvring conditions. Sea surface is based on real wave spectra and the numerical simulation is obtained by the discretization and conversion of the spectrum signal from frequency to time domain through fast Fourier transform (FFT) algorithms.[8] The overall response of the ship is then computed in real time by summing all the contributions of the RAOs for a large set of regular waves with different frequencies, amplitudes and directions of propagation.[9] As in Datta et al.,[10] the boundary integral equation method (BIEM), also known as the panel method, is often used to pre-compute the RAOs for numerical simulations of ship motions. The conventional low-order method, typically, uses the hull geometry represented by small quadrilateral panels, while more powerful higher-order methods rely on more exact representation of the hull such as non-uniform rational basis spline (NURBS) geometry. Therefore, to allow higher-order methods on RAOs computation, the hull should preferably be modelled using NURBS surfaces (Figure 2, NURBS model). If these methods are not to be applied, then linear geometry of quadrilateral panels may be obtained from the NURBS model as described in Lee and Joun[11] or Datta and Guedes Soares,[12] and shown in Figure 2 – hydrodynamics model.

The main difference between the hydrodynamics and the visualization polygonal models is the subdivision of the underwater part of the hull. In fact, the wetted surface is the only part of interest for the hydrodynamics model. However, as referred by Datta et al.,[10] horizontal panels near the water-free surface (a situation that often occurs in the stern region) may generate instabilities in the calculation, and therefore, an artificial row of relatively small vertical panels (Figure 2, hydrodynamics model) may be added above the waterline. The hydrodynamics model must not have panels larger than a certain dimension independently of the curvature of the surface. For lower-order methods, the solutions for the velocity potential are approximated by piecewise constant values on each panel, and therefore, the size of the panel influences the accuracy of the results. Ko et al.[13] mention that a minimum of 8–10 panels for one wavelength of an incident wave need to be considered. This is the reason why, on the flat of side region of the hull, where the curvature of the surface is zero, the hydrodynamics model is still highly subdivided, contrarily to the homologous region in the visualization model. The resolution of the visualization model depends on the curvature of the surface and on the silhouette of the hull. Therefore, the density of the mesh increases at the bow and stern regions.

Collision models tend to be coarser to maintain the performance of the real-time collision detection. Collision detection algorithms use collision models to detect interferences between geometric entities, as well as the location of the collision point, the normal vectors to both surfaces and the velocities at the impact point. The accuracy of the calculation, which strongly depends on the resolution of the collision model, determines the accuracy of the resulting forces and the responses of the ships to the collision. Similar to the visualization model, the rule to determine the polygonal density of the collision model is the curvature of the surface. After defining the mesh for the collision model, popular model partitioning techniques such as the bounding volume hierarchy (BVH), described in Fang et al.,[14] may be applied to the hull model, generating more tree nodes in the zones with higher mesh density as described in Varela and Guedes Soares.[15]

Figure 3. Ship bridge console of a tugboat (left image) and a large merchant vessel (right image).

Within each model presented in Figure 2, the topology and arrangement of the meshes or surfaces may change depending on the type of vessel and shape of the hull. The important aspect is that the main features mentioned previously that characterize each model are maintained. In this context, they are used as a guideline to explain the main characteristics of each model.

Interior of the ship

Within the scope of ship manoeuvring simulation, the interior of the ship is normally restricted to the ship bridge. The interior of the ship bridge is a requirement for the primary ships in micro- and VR-based simulators. Special attention must be given to this part of the model, because it is the location where the user spends most of the time during the simulation. Moreover, the interior of the ship bridge also includes the instruments to control the ship and its systems, and the readouts that provide valuable information about the ship status.

Control console. The most important object in the ship bridge is definitely the control console. In modern ships, the console can be very complex with several buttons, lights, screen displays, small handles and clock meters as shown in Figure 3 for the case of a tugboat and a large commercial ship on the left and right images, respectively.

Micro- and VR-based simulators are more flexible to change the ship bridge configuration, because the scenario and interface is purely virtual (most of the times, this is not possible with more sophisticated systems, such as mini- and full-mission simulators).

In order to be recognized as professional training systems, modern ship bridge simulators are required to comply with the STCW 2010 (International Convention of Training, Certification and Watch Keeping for Seafarers),[16] International Safety of Life At Sea (SOLAS) Conventions,[17] and be certified by the Det Norske Veritas (DNV) Standard for Maritime Simulator Systems.[18] Control consoles of such systems are therefore equipped with the following multifunction displays and controls:

- Conning information display (Figure 4(a));
- Electronic chart display information system (ECDIS) display (Figure 4(b));
- Automatic radar plotting aid (ARPA) and automatic tracking aid (ATA) displays (Figure 4(c));
- Automatic identification system (AIS) display (Figure 4(d));
- Echo sounder (water depth indicator) display (Figure 4(e));
- Alert management system display (Figure 4(f));
- Engine/propulsion control;
- Steering control;
- Compass (gyro or magnetic);
- Navigational lights controls and indicators;
- Line handling possibilities and anchor handling controls and monitors;
- Radio-communication systems, including global maritime distress and safety system (GMDSS), phone, very high frequency (VHF) and ultra-high frequency (UHF).

The conning display includes the following information:

- Ship's heading, pitch and roll;
- Rudder angles;
- Rate of turn;
- Position fix;
- Speed (bow speed, longitudinal speed and stern speed);
- Propellers' speed and pitch;
- Wind speed and direction (both true and relative);
- Thrusters' pitch;
- Alert list;
- Bridge navigational watch alarm system (BNWAS) indication;
- Date and time.

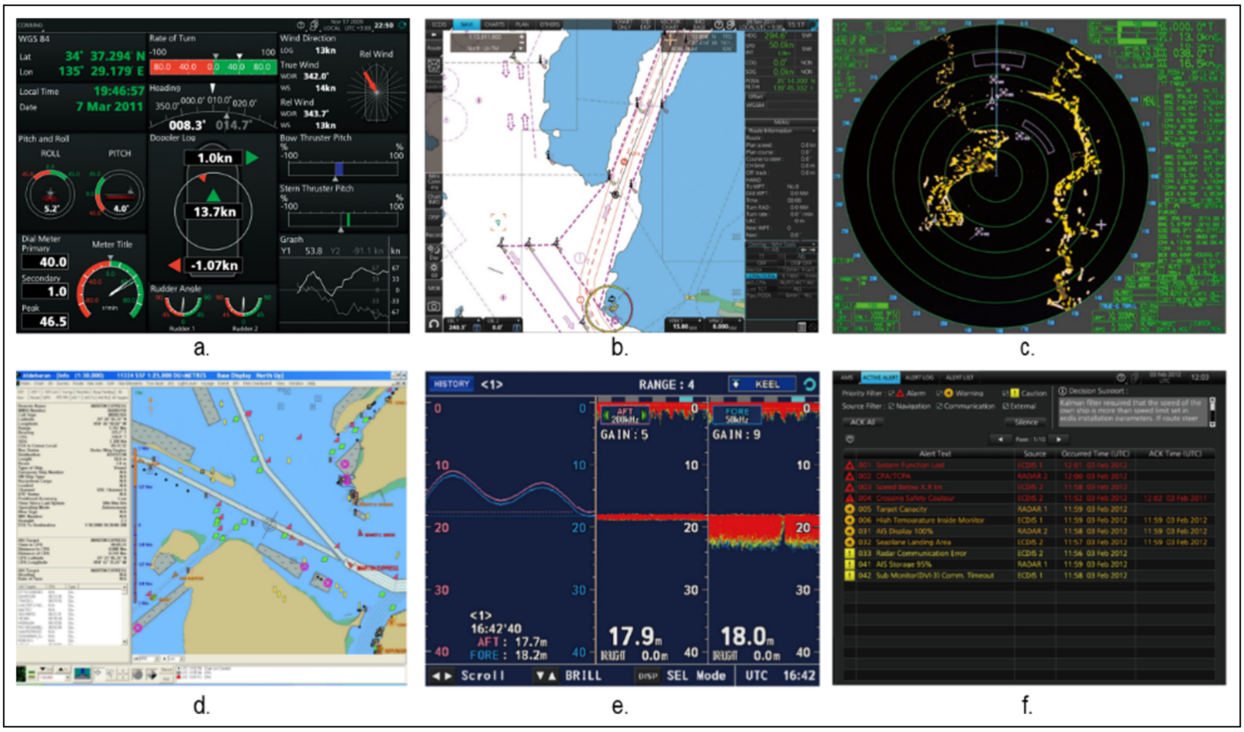

Figure 4. Multifunction displays of SBS control consoles: (a) conning information display; (b) electronic chart display information system (ECDIS); (c) automatic radar plotting aid (ARPA) and automatic tracking aid; (d) automatic identification system; (e) echo sounder; (f) alert management system.

Figure 5. Virtual ship bridge console with and without textures.

The use of textures to replace geometry and to provide more realism to the ship console is in this case advisable. Many buttons and panel lights are not geometrically modelled as can be seen in Figure 5, which presents the virtual ship bridge of the tugboat without textures on the left image and fully textured on the right image. Textures are adequate for objects that do not raise up too much from the console panels. Buttons, clock meters and screen displays are the ideal candidates. Handles, joysticks or buttons that raise considerably from the panels must be modelled with geometry.

Readouts. During the operation, the user must be able to perceive efficiently the information displayed in the instruments and controllers. Although in the real world, this is an obvious and easy task for an experienced operator; in the VE, this may be not so easy mainly due to limitations of the display devices. In order to overcome this limitation, we define three auxiliary readout types: the system indicators, the local indicators and local magnifiers. Figure 6 presents examples of each type of readout for the virtual console of the tugboat.

All the three types of readouts are impostors rendered on the top of the VE, hiding part of the scenario that supposedly should be always visible. Therefore, unless additional monitors or display devices are used, the user must be able to show/hide all the readouts individually.

The system indicator is a readout type that shows the status by system. All the relevant values for the system in question are shown by the indicator. In addition to the numerical values, graphical representations may also be added to this type of readout. Its size is

Figure 6. Additional readouts to provide clearer information of the console instruments.

normally small to avoid hiding large portions of the scenario. It is usual to use this type of readout as a controller of the system. Typically, it is used to show the status of propulsion, steering or navigation systems. System indicators are not zoomed displays of the instruments in the sense that their graphical representation may not coincide with the one in the console (which is the case in Figure 6).

Local indicators are small readouts indicating a single value of a control device or a button. They indicate simply if a button is on or off, the value of a small clock meter, or the position of a handle. Normally, they are only displayed while the cursor is crossing the instrument.

Local magnifiers are used to display more complex information such as radar screens, ECDIS maps or panels of buttons. Contrarily to system indicators, whose graphical representation may be symbolic, in this case the appearance of the readout should be as close as possible to the virtual object although zoomed and faced to the viewer (the reason why they are called magnifiers). For large local magnifiers, semi-transparency may be applied to avoid hiding significant parts of the scenario.

It is also important to highlight that multifunction displays and their corresponding local magnifiers should resemble as much as possible the real displays in the console. This means that colour and lighting changes specified by standards, such as the International Hydrographic Organization (IHO) S-52[19] must be respected. Images displayed on the multifunction screens are not part of the modelling itself and should be received by an external source as a texture, which is mapped into the screen or local magnifier areas.

High-poly and low-poly versions of the model

Similar to polygonal models used in other 3D interactive applications, there are two versions of the ship model: the low-poly and the high-poly models. The low-poly version of the ship is the one that is used in the simulation, and it is based on the high-poly version of the same model. However, both versions should be created on the top of an initial model usually called the control mesh.

High-poly version

The high-poly version is a model without any restrictions concerning the number of geometric primitives or objects. Consequently, it easily reaches the several hundred thousand or even millions of polygons as shown in Figure 7.

The detail of the high-poly version is independent of the number or complexity of the objects.

Figure 8 presents the high-poly model of hawser winch and anchor windlass of a tugboat. Bolts, nuts, buttons and small handles are modelled independently of their size. The large number of primitives masks the linear nature of the mesh, namely, at the edges of curved and rounded shapes, which become imperceptible.

Control mesh

Concerning the level of detail (LOD), the control mesh is an intermediate version of the model, which has the same number of objects as the high-poly version, but all modelled with relatively simple shapes as shown in Figure 9 for the case of the hawser winch engine of the tugboat (Figure 8).

The control mesh is the first polygonal model to be created, and its purpose is to define the main shape of the ship and all its objects. In the control mesh, all the objects of the ship are modelled with enough resolution to represent their overall shape. However, since they are created manually from scratch, their geometry must also be simple enough to apply methods normally used in hard surface modelling such as primitive modelling, box modelling or patch modelling.[20] From the control

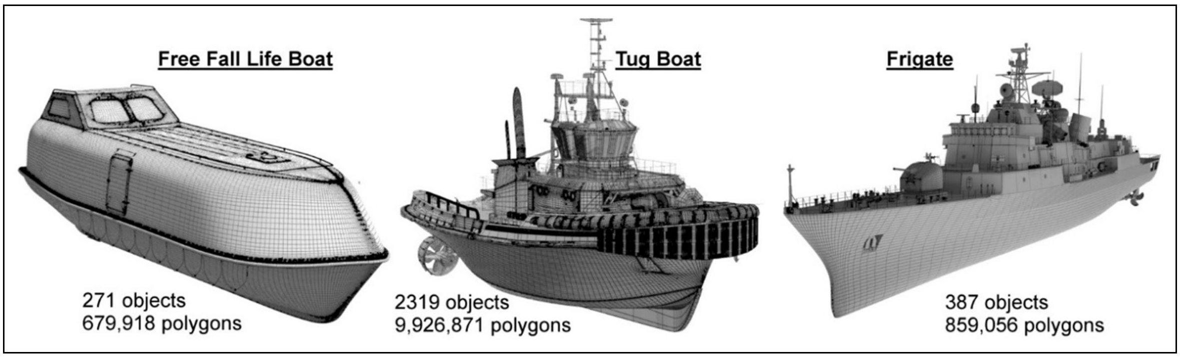

Figure 7. High-poly models of ships.

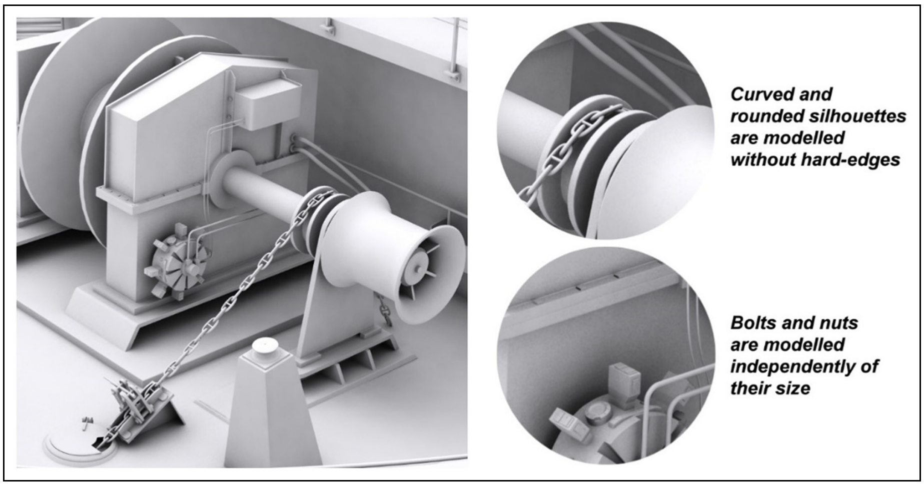

Figure 8. Characteristics of the high-poly version of the model.

mesh, the high-poly version of the model is automatically obtained through subdivision algorithms. Also, the low-poly version of the model may be obtained from the control mesh using the high-poly version as reference.

Low-poly version

The low-poly version of the model must be simplified as much as possible regarding the number of objects and the number of primitives that compose each object. Many geometric details and even entire objects are replaced by textures. Two main approaches may be taken to create the low-poly version of the model: create the low-poly model from the control mesh, or create the low-poly from scratch using the high-poly model as reference. For some objects, both approaches may be used for different parts of the geometry.

Most of the times, creating the low-poly model from the control mesh is the fastest option. In fact, most of the time is spent on the creation of the control mesh. Getting a low-poly version out if it is more a matter of deleting some objects and readjusting some geometry and topology.

The model of Figure 9 is a typical case where both approaches were taken to obtain the low-poly version of the model. The main body of the low-poly version was created as a new cylinder with 12 sides aligned with the main body of the control mesh. This was faster than removing all the small holes and buckles of the main body in the control mesh and then readjusting its geometry and topology. On the other hand, for the remaining objects, the readjustment of the control mesh geometry is faster. Moreover, radial objects around the main body were defined as instances of the same geometry and therefore, the simplification of one object was propagated to the geometry of all the instances. Some smaller objects are simply deleted (as can be seen by the difference in the number of objects and elements) and their geometry is then replaced by textures in the low-poly version.

The UV map

The creation of the UV map starts by unwrapping the model. Basically, unwrapping the model consists in unrolling the 3D mesh into a two-dimensional (2D) shape that can be fitted into a texture map. For the

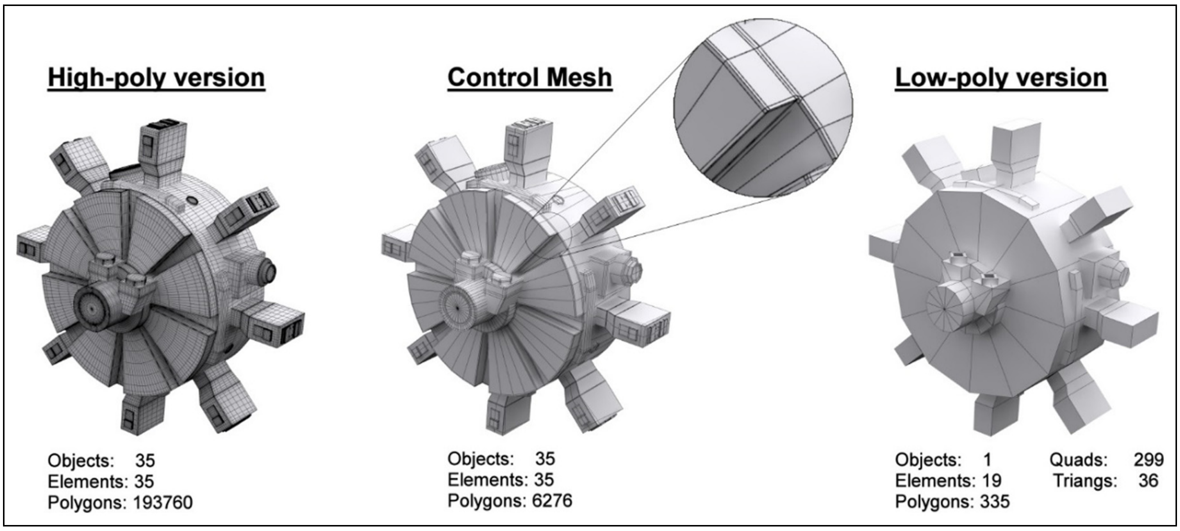

Figure 9. Comparison between the three versions of the same model.

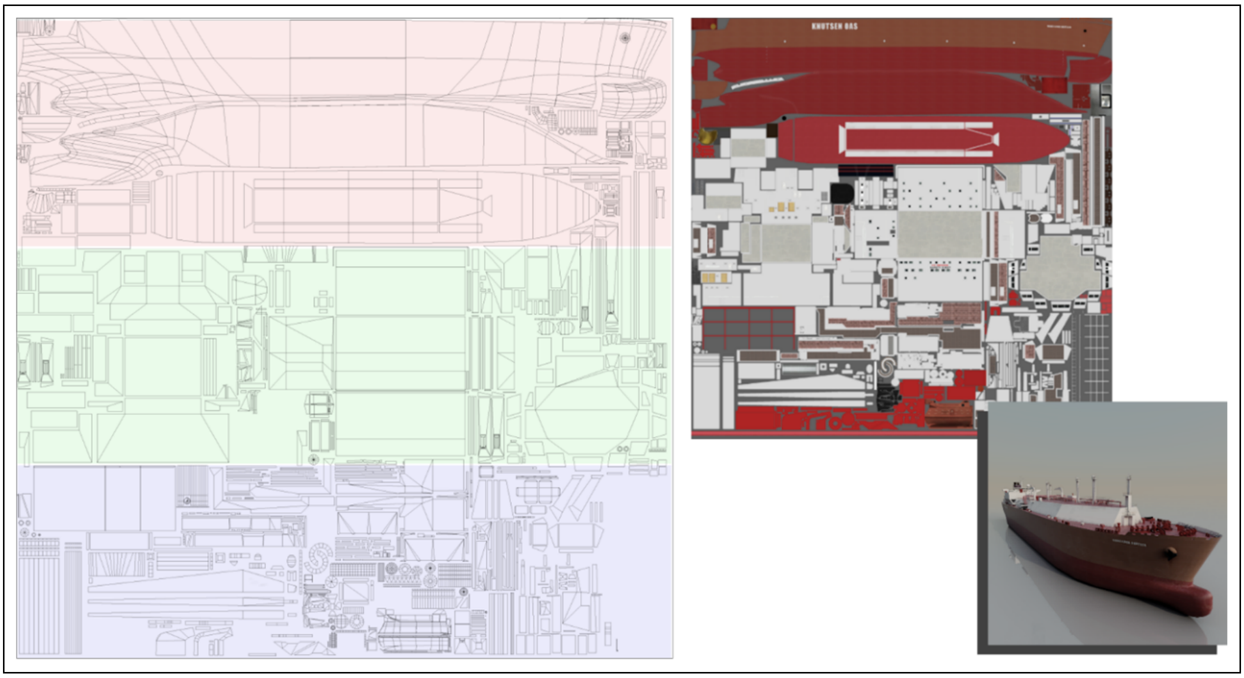

Figure 10. The unwrapped UV map of the LNG vessel.

unwrapping process, the number of texture maps, their size in pixels and the contents of each map should be considered. Figure 10 presents the unwrapped model of an LNG tanker for the case when a single map is used.

In the presented scheme, the map is divided into three main zones: the hull and deck (top zone), the superstructure (middle zone) and the equipment (bottom zone). Within each area, the distribution of the objects should also follow some criteria. This will help finding specific objects in the map for eventual editing of the texture. One possible and logical criterion is to distribute objects by systems. All the mooring, lifesaving, lighting or piping equipment should be gathered in specific areas.

Another characteristic of the unwrapped maps of ship models that are normally not adopted in smaller vehicles is the multi-scale applied to the different objects. This comes as an obvious choice in the sense that a large object such as a hull with more than 200 m cannot be unwrapped with the same scale as a mooring bollard with less than 1 m. For this purpose, it is also necessary to define the intended scale for each object or set of objects. We propose the following measure of the LOD based on the area represented by each pixel:

$$\mathrm{LOD} = \frac{N_P}{A_R} \tag{1}$$

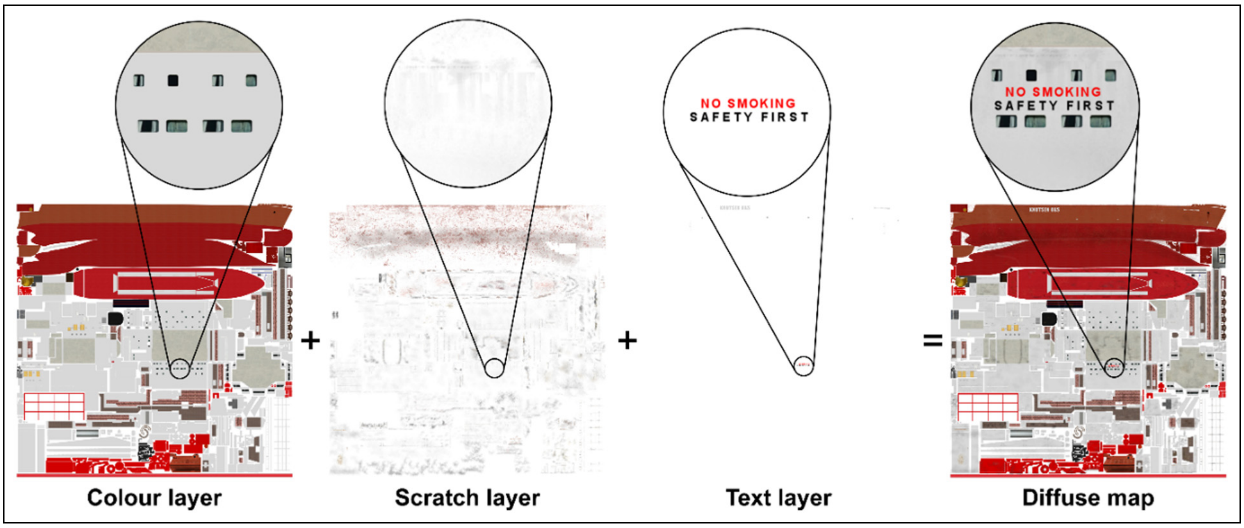

Figure 11. The diffuse map of the LNG tanker.

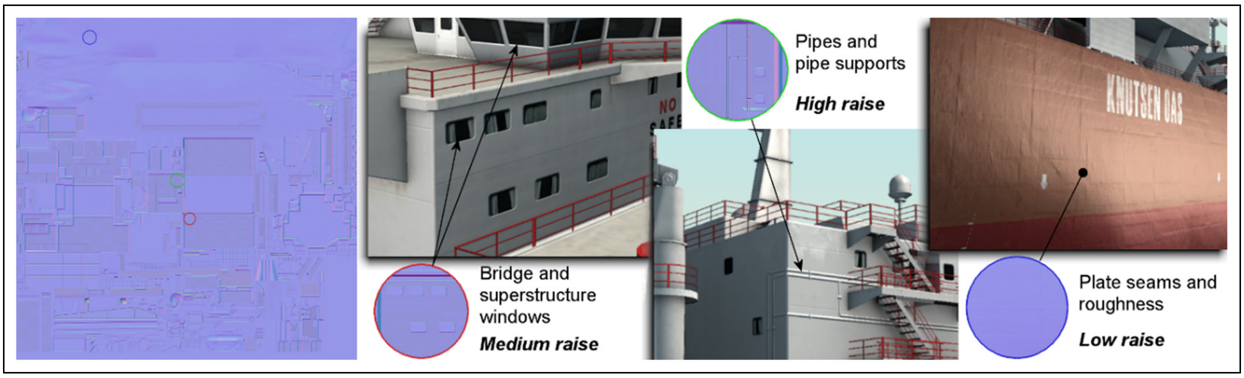

Figure 12. The normal map of the LNG tanker.

where A_R is the area of the surface of the object in the real world and N_P is the number of pixels that A_R covers in the texture map.

In order to maintain LOD of the map high enough to obtain realistic appearance, namely, for larger ships, the use of overlapping UVs is highly advisable. This means that the same pixel in the texture feeds two or more different areas in the mesh. The case of the hull is the most obvious one: due to its symmetry along the longitudinal axis, only one of the sides is represented in the texture. The same happens for the free fall life boat. Also the ship has various instances of the same object, such as hatches, anchors, windlasses, lights among others, which use the same texture area for their UV maps.

Finally, the reuse of hidden zones in the map should also be considered. In this case, hidden zones in the map correspond to the regions of an object that are not visible in the model but fill some considerable space in the map. This space can then be used to overlap the UV map of other visible objects. In Figure 10, one of these zones is the deck geometry below the continuous tank cover, which is reused to map the cover itself with a slightly lower LOD.

Textures

Specification for textures refers to the contents of the UV map. Adding detail to the texture is a highly time-consuming task, which may have a significant impact on the schedules and on the final cost of the model. Therefore, the specification for texturing will help to make an estimation of the time and cost associated to this task. We consider four types of textures as the main ones that should always be used: the diffuse, the normal, the lighting and the opacity maps.

Diffuse map

Figure 11 presents the diffuse map layers used on the LNG tanker model.

The diffuse map is the one that most influences the appearance of surface. It defines the material colour, scratches or dirt, and may include signs or text. Although the final diffuse map used in the simulator is a single texture, it is almost mandatory to maintain a multi-layered version of this map for post-editing. This will allow changing the appearance of the material, adding or removing text, signs or scratches much more

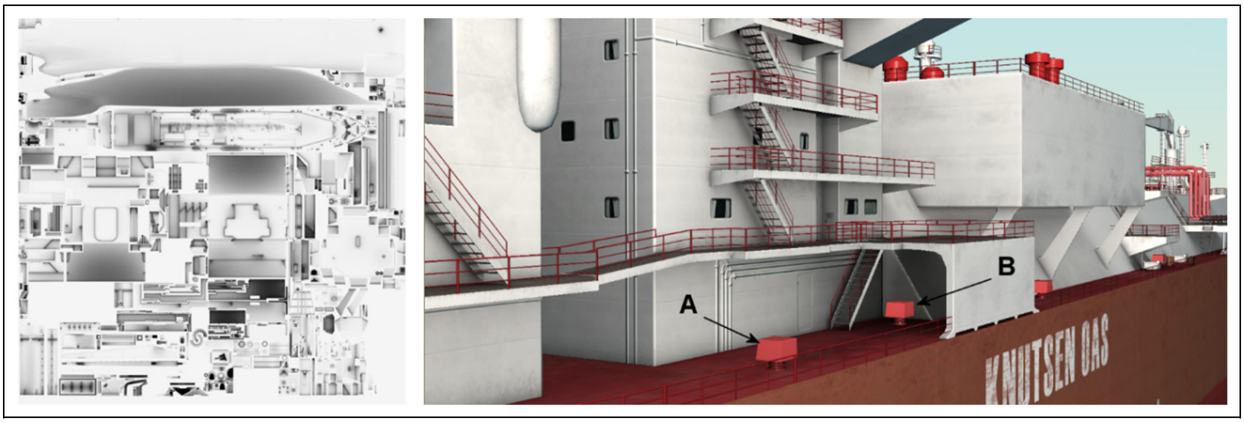

Figure 13. The lighting map of the LNG tanker.

efficiently. The final diffuse map should be delivered at least with the following three layers: the colour, the scratch and the text.

The colour layer specifies the base colour or appearance of the material. The realism is normally achieved through material photos or procedural textures. Then, the scratch layer includes the rust, the scratches and the dirt on the surfaces. These may reflect the age of the ship, operation or maintenance conditions. The text layer includes items such as the ship name and symbols painted in the hull or superstructure.

Normal map

The normal map modifies the normal vector of the surface according to the colour of the pixel in the texture (mapped into the 3D surface). This will affect the view angle in the shading equation and therefore change the appearance of the surface. The use of the normal map is related with the required LOD for the appearance of the object. Normal maps are applied to surfaces to replace details or objects that otherwise would have to be modelled with geometry. The range of application of normal maps is defined by the amount that these objects raise from the surface. In ship models, the normal map is adequate to represent objects and details from a few millimetres, like the plate seams and roughness, to several centimetres, like pipes running along the ship superstructure.

Figure 12 presents three types of details of the LNG model obtained with the support of normal maps, where different raising levels are observed.

Details with medium and high raises are usually supported by the lighting map to increase the 3D effect. Small soft shadows are pre-computed and added to lighting map in the zones where the detail raises from the surface. For the low-raise cases, these shadows are imperceptible and therefore there is no need to compute them. The use of normal maps is also related with the size of the objects. If the object is small, the 3D effect generated by the normal map may not be perceptible and therefore there is no use of modelling it.

Lighting map

The lighting map controls the exposure of each point in the surface to the ambient light. With modern rendering algorithms and hardware, it is already feasible to compute ambient exposure in real time. However, these techniques consume extra computational resources and are normally used when dynamic computation of soft shadows due to ambient light is required. For the case of ship models, static lighting maps are pre-computed for a determined lighting condition. Static lighting maps may raise some visualization issues for objects that have multiple instances in the ship model. This happens when the same lighting map is used for two or more instances of the same object located in zones with different light exposure. Figure 13 presents the lighting map we used for the LNG model (left image) and the final effect on the ambient light (right image).

The same ambient light (taken from the map) is applied to the ventilation pipes A and B on the right image. However, the lighting map was computed for the exposure conditions of the ventilator A and therefore the ventilator B, located at a lower light exposure zone, is over lighted. In order to overcome this issue, each instance of the object (ventilator) should indicate the exposure level, as a mean to define the blend level applied to the lighting map.

Opacity map

Figure 14 presents the opacity map with some of its main applications to the LNG ship model.

The opacity map in ship models is mainly used on balustrades, chains, vertical stairs and pipes. It can be easily seen in Figure 13 that modelling all the rail stanchions and lines of the balustrades with geometry would increase drastically the number of primitives of the model. The map of Figure 14 highlights the much higher scale of objects such as balustrades, vertical stairs and chain cables. Objects that take advantage of opacity maps normally have higher scales in the texture to avoid excessive aliasing between the opaque and

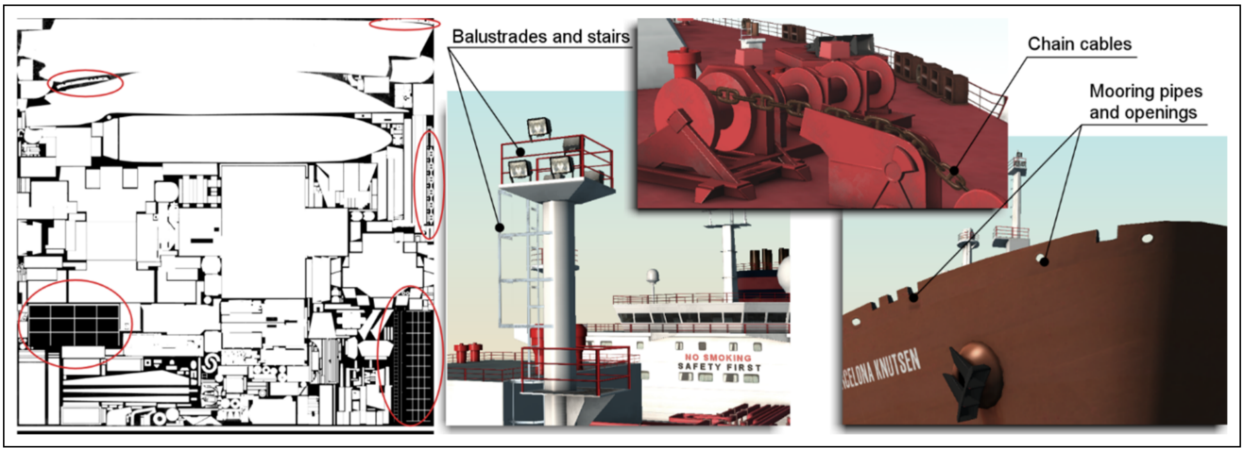

Figure 14. The opacity map of the LNG tanker.

transparent areas. Balustrades are normally the objects where the advantage of using overlap UVs is more noticeable. A relatively small UV area representing only five rail stanchions is enough to map all the balustrades of the ship.

Conclusion

In this article, we have presented a comprehensive analysis of the requirements and techniques to produce the geometry of ship models for modern ship bridge simulators. We have specified some of the most important aspects that influence the modelling time, efforts and quality of the ship models.

We concluded that the modelling procedure should be planned, based on the type of ship simulator and the level of participation of the ship in the simulation. These two factors indicate which objects should be modelled or which LOD should be assigned to the model. The hull geometry is to be addressed independently due to its role on the ship hydrodynamics and collision detection. If the ship bridge is modelled, special attention should be given to the control console and instruments. For an efficient control and monitoring of the ship, the auxiliary readouts should be created for the most important instruments.

The modelling procedure follows approximately the current standard modelling procedure for generic interactive 3D models, using the control mesh as the base model to derive the high-poly and low-poly versions of the model, through the unwrapping and baking techniques. However, the unwrapped UV map contains certain particularities, not commonly find in other models. This is due to the large number of objects in ships, many of them which are instances of the same geometry, with very different sizes and materials. As a consequence, objects are unwrapped with different scales in the same UV map, and their distribution through the 2D space should be well organized. The use of overlap UVs in this type of models is highly advisable.

Concerning the use of textures, we addressed what in our opinion are the four most important textures to obtain a realistic model. We highlighted the importance of maintaining the different layers of the diffuse map for post-editing. For large vessels, opacity maps are fundamental to reduce the polygon count of objects such as handrails, stairs and chains.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

This work was performed within the Strategic Research Plan of the Centre for Marine Technology and Ocean Engineering, which is financed by Portuguese Foundation for Science and Technology (Fundação para a Ciência e Tecnologia-FCT).

References

1. Ali A. Role and importance of simulator instructor. *Int J Mar Navig Saf Sea Transp* 2008; 2(4): 423–427.
2. Bowman D and McMahan R. Virtual reality: how much immersion is enough? *Computer* 2007; 40(7): 36–43.
3. Varela JM and Guedes Soares C. Interactive 3D desktop ship simulator for testing and training offloading manoeuvres. *Appl Ocean Res* 2015; 51: 367–380.
4. FORCE Technology. SimFlex4 ship bridge simulator, http://forcetechnology.com/en (accessed 9 February 2016).
5. Kongsberg. K-Sim® navigation ship bridge simulator, http://www.km.kongsberg.com/ (accessed 9 February 2016).
6. VStep. NAUTIS ship bridge simulator, http://vstepsimulation.com/ (accessed 9 February 2016).
7. Hensen H. *Ship bridge simulators: a project handbook*. London: The Nautical Institute, 1999.
8. Varela JM and Guedes Soares C. Ring discretization of the wave spectrum for sea surface simulation. *IEEE Comput Graph Appl* 2014; 34(2): 58–71.
9. Varela JM and Guedes Soares C. Interactive simulation of ship motions in random seas based on real wave spectra. In: *GRAPP 2011 – proceedings of the international conference on computer graphics theory and applications*,

Vilamoura, 5–7 March 2011, pp.235–244. SciTePress Digital Library, http://www.scitepress.org/DigitalLibrary/PublicationsDetail.aspx?ID = e%2fPKDdDY260%3d&t = 1, (accessed on 12 April 2016).
10. Datta R, Rodrigues JM and Guedes Soares C. Study of the motions of fishing vessels by a time domain panel method. *Ocean Eng* 2011; 38: 782–792.
11. Lee M and Joun M. General approach to automatic generation of quadrilaterals on three-dimensional surfaces. *Commun Numer Meth En* 1998; 14: 609–620.
12. Datta R and Guedes Soares C. NURBS based scheme for automatic quadrilateral mesh generation for FE and BIEM analysis. *Marine Systems and Ocean Technology*. 2012; 7(1):29–35.
13. Ko K, Park T, Kim K, et al. Development of panel generation system for seakeeping analysis. *Comput Aided Design* 2011; 43: 848–862.
14. Fang J, Clark D and Simmons J. Collision detection methodologies for rigid body assembly in a virtual environment. *Virtual Real* 1995; 1: 41–48.
15. Varela JM and Guedes Soares C. Software architecture of an interface for three-dimensional collision handling in maritime virtual environments. *Simulation* 2015; 91(8): 735–749.
16. IMO. *STCW – conference of parties to the international convention on standards of training, certification and watchkeeping for seafarers, 1978 – agenda item 10. STCW/Conf.2/34*, 3 August 2010. Manila, Philippines, http://www.imo.org/en/OurWork/HumanElement/Training Certification/Documents/34.pdf (accessed on 12 April 2016)
17. SOLAS - Consolidated Edition 2014. IMO publishing. London, United Kingdom. www.imo.org.
18. Standard DNVGL-ST-0033:2014-08:2014. DNV GL AS and maritime simulator systems.
19. International Hydrographic Organization (IHO) S-52. Specifications for chart content and display aspects of ECDIS (Edition 6.0). International Hydrographic Bureau, Monaco, March 2010.
20. Vaughan W. *Digital modeling*. Berkeley, CA: New Riders, 2012.