POLISH MARITIME RESEARCH 4 (96) 2017 Vol. 24; pp. 42-46
10.1515/pomr-2017-0134

VISUALIZATION OF A LIFEBOAT MOTION DURING LOWERING ALONG SHIP'S SIDE

Aleksander Kniat
Gdańsk University of Technology, Faculty of Ocean Engineering and Ship Technology, Poland

ABSTRACT

This paper presents description of a computer program for motion visualization of a lifeboat lowered along ship's side. The program is a post-processor which reads results of numerical calculations of simulated objects' motions. The data is used to create scene composed of 3D surfaces to visualize mutual spatial positions of a lifeboat, ship's side and water waving surface. Since the numerical data contain description of a simulation as a function of time it is possible to screen a static scene showing the simulated objects in an arbitrary instance of time. The program can also reproduce a sequence of scenes in the form of animation and control its speed. The static mode allows to view an arbitrary cross-section of the scene, rotate and enlarge specific details and make the image more realistic by hiding invisible lines or shading. The application of the program is aimed at making it possible to assess and analyze numerical calculation results in advance of their experimental verification.

Keywords: three-dimensional visualization, kinematics, computer animation, lifeboat lowering , numerical simulations

INTRODUCTION

Spatial visualization and animation of images in the form of a film represents computer techniques which are commonly applied to present numerical calculation results. This was made possible as a result of increasing computing power of PCs and universal window environment dedicated to graphic applications ensured by operating systems. Fast development of the technology has been stimulated first of all by the market of entertainment and games. Examples of highly sophisticated techniques used in cartoon films for home cinema can be found in [12]. In this context it is a natural trend of professional users to see results of their work not in the form of numbers but in a graphic form of imagination appealing pictures.

Theoretical background for solutions in the field of mechanics is contained in [8]. An example of simulation of human body motions are given in [9]. The application of simulation of crane operation in building industry is presented in [1]. In[10] a similar technique is shown for simulating interaction of many devices present at construction site. The visualization found its application also in robotics and CNC (Computerized Numerical Control). There are many programs which allow to carry out simulations of robot or machine tool operation in advance of their implementation to industrial tasks. Such approach makes it possible to avoid errors which may cause expensive failures of machines or destruction of valuable materials. An example of open source program applicable to simulation of robot operation is described in [15]. A broader analysis of programs available in this domain is given in [3]. Commercial programs examples for instance K-Roset program for Kawasaki robots or Italian universal program Eureka by Roboris can be found in [11] and [6]. Visualizations of results of operation of an algorithm for automatic routing the pipelines in three-dimensional space are presented in [14]. Above mentioned cases exemplifies design applications.

Making use of computer visualization becomes especially important when complex mathematical multi-parameter models are analyzed because description of their state contains yet greater number of data. The finite element method (FEM) as well as boundary element method (BEM) can be distinguished as an excellent example of such problems. Today nobody can imagine to work with a FEM solver without using a pre- and post-processor. Functional assumptions for such visualizing programs are presented in [2]. Many commercial programs are in use today for this purpose. The application concerning visualization of ship hull strength calculations [13] represents work made by the author in the domain in question. Further development of computer techniques in the field of visualization is inevitable. It proceeds especially dynamically in the area of home entertainment where virtual reality constitutes basis for computer games and animated cartoons. There is no reason for which the same technique could not be applied in professional domain. Additionally computer simulations are less expensive than physical experiments.

KINEMATIC MODEL AND RESULTS OF CALCULATIONS

This work concerns a particular issue which is visualization of lowering a lifeboat along ship's side. The kinematic model of the process includes the following objects:

- lifeboat,
- fender bar installed on boat's side,
- lines on which the lifeboat is hung,
- mechanism for lowering the lifeboat,
- boat hook which allows to set the lifeboat free after its launching,
- boat davits from which the lifeboat is lowered,
- ship's side along which the lifeboat is lowered,
- water waving surface onto which the lifeboat falls.

Each of the above specified objects was described by an appropriate set of parameters defining its features. A comprehensive description and results of operation of RESBO program which performs numerical simulation of behavior of the system composed of the so defined objects are given in [4]. A mathematical model of kinematics of the entire system can be found there too. The program produces numerical data which represent, in given time intervals, the state of the system characterized by forces and moments acting on the objects, as well as their velocities and locations. Description of the system's states as a function of time is recorded by the program in disk text files. Hence, to assess results of simulation produced by RESBO program it is necessary to analyze numerical data in the files. At first the results have been analyzed by presenting them on diagrams. Unfortunately, when it was necessary to analyze results of more difficult cases or their larger number, as e.g. in [5], difficulties in interpretation and correlation of relevant diagrams occurred. It was especially troublesome to check in which direction the lifeboat is rotated, having only numerical values of co-ordinates of its center of gravity and co-ordinates of the three versors of local reference frame.

Contemporarily there are available ready-to-use programs for kinematic simulation and spatial visualization, such as e.g. ADAMS discussed in[16]. However the author did not decide to apply any of them to the case in question. The reason was that it was impossible to simulate interaction between the lifeboat and water surface (slamming) and behavior of the lifeboat floating in waves.

ASSUMPTIONS FOR VISUALIZATION PROGRAM

To make analysis of calculation results obtained from RESBO program easier it was decided to prepare a new program called VRBS which makes it possible to visualize spacial location of objects taking part in simulation. The following functions were assumed for the program:

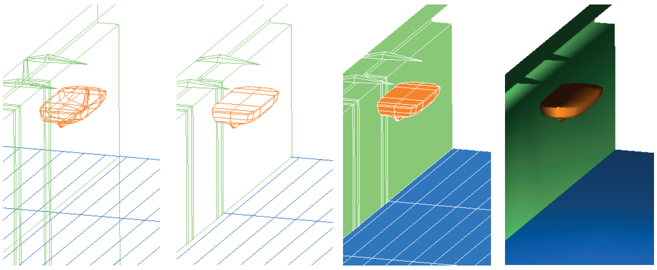
- visualization of an arbitrary scene composed of spacial objects represented by quadrangular and triangular mesh in the form of axonometric view;
- presentation of the scene in four different forms:
  - transparent (only edges of the mesh are visible),
  - hidden lines removed (edges of the mesh are obscured by elements located closer to observer),
  - opaque (interior of each mesh element is filled with color and only non-hidden elements or their fragments are visible)
  - shaded with lighting (interior of each mesh element is filled with color, tint of color depends on light incidence angle and only non-hidden elements are visible);
- shaping of water free surface onto which the lifeboat is lowered, in an arbitrary instant;
- positioning ship's side and ends of arms of boat davits from which the lifeboat is lowered, in an arbitrary instant;
- positioning and orientation of the lifeboat itself, in an arbitrary instant;
- simulation of motion of the objects in the form of a film by sequential projecting of following scenes based on calculation results obtained from RESBO program;
- stopping motion simulation, jumping to an arbitrary instant and continuing the motion simulation.

SPATIAL MODEL AND METHODS OF VISUALIZATION

The computational model of the scene consists of the objects specified in section „Kinematic model and results of calculations". Since not all of them are necessary for visualization, graphic representations were prepared only for:

- lifeboat,
- ship's side and davits by which the lifeboat is lowered,
- waving water surface onto which the lifeboat falls.

Each of the above mentioned objects is graphically represented as a 3D surface. Shape is assumed to be mapped by a quadrangular mesh. Since the spatial model will be imported from a CAD program and in order not to lose solution generality it was assumed that quadrangular mesh will be presented as NURBS surface of the first degree, described in [17]. In such case control points of the surface are the same as mesh vertices. This approach makes it possible to import from a CAD program not only mesh but also smooth surfaces. VRBS program uses the OpenGL library [18] to achieve professional effects of 3D object's visualization. Such approach let the author to focus on the details of the scene and motion but not on issues of geometrical transformations, projection and rendering. An exemplary image of a ready 3D scene consisting of objects subjected to motion simulation is presented in Fig. 1.



a) b) c) d)

Fig. 1. Image of scene shown in axonometric parallel projection: a) edge transparent, b) edge one with hidden invisible lines, c) colur filled, d) shaded

MOTION AND TIME RELATIONS

Because the surfaces are assumed to be the NURBS surfaces the visualization program is to operate in the space of homogenous co-ordinates. If 3D space is taken into account in which the system under simulation is considered, then every point of the space is represented by four co-ordinates. The fourth co-ordinate is interpreted to be a weight. In kinematic system, objects have six degrees of freedom. These are three translations and three rotations with respect to each axis of co-ordinate frame. Superposition of appropriate translations and rotations makes it possible to obtain an arbitrary location of the simulated objects. The translation matrix used in geometrical transformations looks always similarly. It is represented by the expression (1).

$$\mathbf{T}(x,y,z)=\begin{bmatrix}1&0&0&x\\0&1&0&y\\0&0&1&z\\0&0&0&1\end{bmatrix} \qquad \textbf{(1)}$$

where: x, y, z - co-ordinates of translation vector.

The rotation is generally composed of three rotations. It is represented by the expression (2).

$$\mathbf{R}(\chi,\beta,\alpha)=\mathbf{R}_\mathbf{X}(\chi)\cdot\mathbf{R}_\mathbf{Y}(\beta)\cdot\mathbf{R}_\mathbf{Z}(\alpha)=\begin{bmatrix}1&0&0&0\\0&\cos(\chi)&-\sin(\chi)&0\\0&\sin(\chi)&\cos(\chi)&0\\0&0&0&1\end{bmatrix}\cdot \qquad \textbf{(2)}$$

$$\begin{bmatrix}\cos(\beta)&0&\sin(\beta)&0\\0&1&0&0\\-\sin(\beta)&0&\cos(\beta)&0\\0&0&0&1\end{bmatrix}\cdot\begin{bmatrix}\cos(\alpha)&-\sin(\alpha)&0&0\\\sin(\alpha)&\cos(\alpha)&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

As numerical results obtained from RESBO program do not contain the rotation angles χ,β,α but only directions of versors of objects' local co-ordinate frame the rotation matrix is described by the expression (3) acc. [7].

$$\mathbf{R}(\mathbf{V}_x,\mathbf{V}_y,\mathbf{V}_z)=\begin{bmatrix}x_{vx}&x_{vy}&x_{vz}&0\\y_{vx}&y_{vy}&y_{vz}&0\\z_{vx}&z_{vy}&z_{vz}&0\\0&0&0&1\end{bmatrix} \qquad \textbf{(3)}$$

where:

$$\mathbf{V}_x=\begin{bmatrix}x_{vx}\\y_{vx}\\z_{vx}\end{bmatrix},\ \mathbf{V}_y=\begin{bmatrix}x_{vy}\\y_{vy}\\z_{vy}\end{bmatrix},\ \mathbf{V}_z=\begin{bmatrix}x_{vz}\\y_{vz}\\z_{vz}\end{bmatrix}$$

are versors of object's local coordinate frame.

Therefore the translation of the object from its original location in the point O and orientation consistent with current reference frame to a new point Q and new orientation **Vx**, **Vy**, **Vz**, is described by the following matrix **M**, acc. to the formula (4).

$$\mathbf{M}=\mathbf{T}(x_q,y_q,z_q)\cdot\mathbf{R}(\mathbf{V}_x,\mathbf{V}_y,\mathbf{V}_z)\cdot\mathbf{T}(-x_o,-y_o,-z_o) \qquad \textbf{(4)}$$

where:

$$\mathbf{O}=\begin{bmatrix}x_o\\y_o\\z_o\end{bmatrix}\ \text{i}\ \ \mathbf{Q}=\begin{bmatrix}x_q\\y_q\\z_q\end{bmatrix}$$

Initial location of all points Pi belonging to scene objects is known. Hence, if at any instance of time t a new location of the origin of object's co-ordinate frame, i.e. the point **Q(t)**, and its versors **Vx(t), Vy(t), Vz(t)** are known, then by using the matrix **M** acc. (4) one is able to find a new location of any point of the scene at this instance of time.

$$\mathbf{P}_i(t)=\mathbf{M}(t)\cdot\mathbf{P}_i=\mathbf{T}(x_a(t),y_a(t),z_a(t))\cdot$$
$$\mathbf{R}(\mathbf{V}_x(t),\mathbf{V}_y(t),\mathbf{V}_z(t))\cdot\mathbf{T}(-x_o,-y_o,-z_o)\cdot\mathbf{P}_i \qquad \textbf{(5)}$$

Numerical results from RESBO program contain subsequent states of the system in given time intervals. If a motion visualizing program has to make it possible to show the motion with its real velocity it must adjust time of displaying subsequent states to real time. The problem consists in that recalculation of new locations of scene points and displaying them on screen does not for sure last precisely as long as the time step used during numerical simulations. In the case when displaying a new scene lasts shorter than time step during numerical calculations, the program waits appropriately long before it starts to display the next scene. When scene displaying lasts longer than the time step there is no other possibility except to omit some of calculated system's states and to display the one which is consistent with capability of the visualization program. However it may cause the effect of lack of motion smoothness as a result of neglecting the intermediate states.

SIMULATION RESULTS

Fig. 2 shows selected subsequent states of simulation of motion of a lifeboat being lowered. Results of the numerical calculations covered the time period lasting 35 s. The calculation time step was equal to 1 ms but only every 10th state was recorded in text files of the results. In such a case the visualizing program should display one by one 3500 states in the real time of 35 s. Unfortunately it turned out that the displaying time of a single scene on an available computer was equal to 16 ms. For this reason it appeared necessary to neglect every second state/scene. In spite of that the final effect was better than satisfactory. Scenes were displayed smoothly, without stopping and jumping, and their animation made impression of real motion.

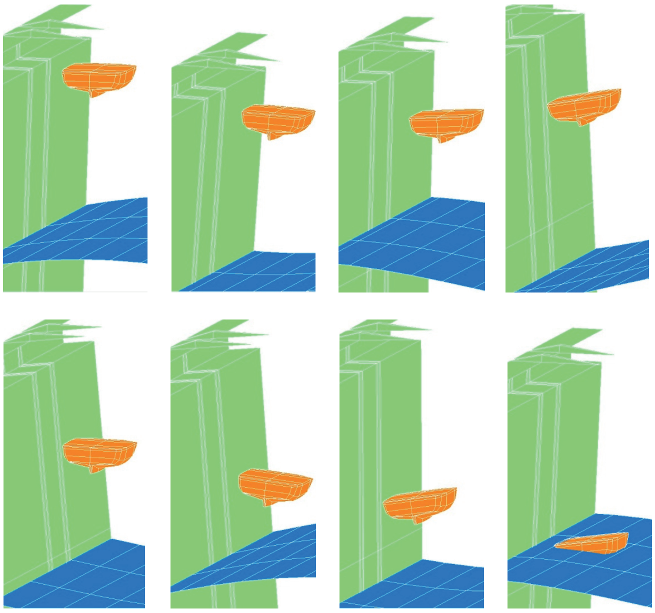


Fig. 2. Single scenes of animation showing motion of lifeboat under lowering

SUMMARY

The VRBS program has been created and it enables visualization of selected scenes and motion of lifeboat being lowered along ship's side. Tests on a few selected results of numerical calculations produced by RESBO program have been done. As the program allows to control motion of visualized objects in time it is possible to directly compare the motion with the film recorded during experiment in real conditions. Synchronizing the visualization of simulation and the recorded film, one is able to compare motion trajectories of objects and analyze differences in mutual location of the objects taking part in the entire process. Therefore it can be stated that the visualizing program VRBS makes possible:

- to analyze various scenarios of lowering lifeboat without incurring cost of real experiments,
- to easily identify dangerous moments during lowering process – impacts against ship's side, capsize of lifeboat,
- to check visually what has happened just before a dangerous situation and to verify motion parameters (acceleration, velocity, distance).

The information obtained this way is a very important element of analysis of considered case. It allows to interpret the data more comprehensively and faster than in the case of using numerical quantities only. Having at one's disposal both the program for numerical calculations (RESBO) and that for visualization of their results (VRBS), one is able to perform simulations which may be helpful in finding optimum scenarios for lifeboat lowering process and avoiding situations which may be hazardous to life and health of evacuated persons.

BIBLIOGRAPHY:

1. Al-Hussein M., Niaz M. A., Yu H., Kim H., Integrating 3D visualization and simulation for tower crane operations on construction sites. Automation in Construction, 15(5), pp. 554-562, (2006).

2. Benzley S. E., Merkley K., Blacker T. D., Schoof L., Pre- and post-processing for the finite element method. Finite elements in analysis and design, 19(4), pp. 243-260, (1995).

3. Craighead J., Murphy R., Burke J., Goldiez B., A survey of commercial & open source unmanned vehicle simulators. 2007 IEEE International Conference on Robotics and Automation, pp. 852-857, (2007).

4. Dymarski P. Dymarski Cz., Computational model for simulation of lifeboat motions during its launching from ship in rough seas. Polish Maritime Research, Vol. 19 No. 3(75), pp. 45-52, DOI: 10.2478/v10012-012-0030-6, (2012).

5. Dymarski C., Dymarski P., Kniat A., Searching for critical conditions during lifeboat launching – simulations. Polish Maritime Research, Vol. 24 Special Issue 2017 S1 (93), pp. 53-58, DOI: 10.1515/pomr-2017-0021, (2017).

6. Roboris, https://www.roboris.it/en/product/eureka-robot.

7. Foley J. D., Van Dam A., Feiner S. K., Hughes J. F., Phillips R. L., Introduction to computer graphics. Addison-Wesley, (1994).

8. Haber R. B., Visualization techniques for engineering mechanics. Computing Systems in Engineering, 1(1), pp. 37-50, (1990).

9. Jan S. V. S., Clapworthy G. J., Rooze M., Visualization of combined motions in human joints. IEEE Computer Graphics and Applications, 18(6), pp. 10-14, (1998).

10. Kamat V. R., Martinez J. C., Dynamic 3D visualization of articulated construction equipment. Journal of computing in civil engineering, 19(4), pp. 356-368, (2005).

11. Kawasaki, https://robotics.kawasaki.com/en1/products/other/simulation-OLP/index.html.

12. Kerlow I. V., The art of 3D: computer animation and effects. John Wiley & Sons, (2004).

13. Bogdaniuk M., Kniat A., Puch W., Effective pre- and post-processors for FEM strength analysis using AutoCAD possibilities. MARINE TECHNOLOGY III, Book Series: MARINE AND MARITIME edited by: Graczyk T., Jastrzebski T., Brebbia CA., 3rd International Conference on Marine Technology (ODRA 99), 1, pp. 35-43, (2000).

14. Kniat A., Optimization Of Three-Dimensional Pipe Routing. Schiffstechnik (ShipTechnology Research), 47, pp. 111-114, (2000).

15. Koenig N., Howard A., Design and use paradigms for gazebo, an open-source multi-robot simulator. 2004 Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), vol. 3, pp. 2149-2154, (2004).

16. Li Z., Kota S., Virtual Prototyping and Motion Simulation with ADAMS. J. Comput. Inf. Sci. Eng., 1(3), pp. 276-279, (2001).

17. Piegl L., Tiller W., The NURBS Book. Second Edition, Springer, (1997).

18. Shreiner D., Sellers G., Kessenich J., Licea-Kane B., OpenGL Programming Guide: The Official Guide to Learning OpenGL, 8th Edition, Addison-Wesley, (2013).

CONTACT WITH THE AUTHOR

Aleksander Kniat

Gdańsk University of Technology
11/12 Narutowicza St.
80 - 233 Gdańsk
Poland