A practical GPU-accelerated method for the simulation of naval objects on irregular waves

A. Bezgodov and A. Karsakov

ITMO University, Russia

Abstract

This paper introduce a new method for real-time simulation of naval objects (such as vessels, ships, buoys and lifejackets) with six degrees of freedom on irregular waves. Thus method is based on hydrodynamic and hydrostatic pressure integration using uniformly distributed random points that are built on each simulation step. Such approach allows us fast and stable pressure integration for arbitrary vessel hull and wave shape.

Categories and Subject Descriptors (according to ACM CCS): I.6.8 [Computer Graphics]: SIMULATION AND MODELING—Types of Simulation—Gaming,Animation

1. Introduction

There are several approaches to simulating floating bodies (floaters). Some of them, which are precise and acceptable for naval object design, utilize finite-element methods (FEM) [GXW04], smooth-particle hydrodynamics (SPH) [ULR13], but suffer from extremely high computational cost. Other approaches are faster but limited by less dimensions of freedom [BDB98], or aimed at solving particular problem [FGS]. However, there are a lot of applications such as simulators and video games where a precise solution is not critical. Some of these approaches is presented here [MM13], [CM11] and here [YHK07].

2. Proposed Method

A brief explanation of proposed method is presented in [BE14] and its application for the study of search and rescue operations. In this paper, we introduce a more detailed description of this method and its implementation on GPU, handling choppy waves and study of properties of proposed method. Consider a naval object to be a rigid body with six degrees of freedom and added mass. The total force $\mathbf{F}$ and torque $\mathbf{T}$ acting on a floating body could be expressed as follows.

$$\mathbf{F} = \iint_{\Phi} p\mathbf{n}d\sigma + \mathbf{D} \qquad \mathbf{T} = \iint_{\Phi} (p\mathbf{n}) \times (\mathbf{r} - \mathbf{p})\, d\sigma \tag{1}$$

where Φ — submerged surface of naval object, D — naval object weight, p — static and dynamic water pressure, $\mathbf{n}$ — surface normal, r — radius-vector of each point on submerged surface of naval object, $\mathbf{p}$ — naval object position.

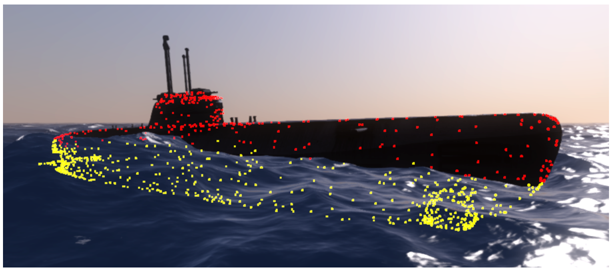

Figure 1: *Points are distributed on hull surface: yellow points are submerged. Notice more detailed parts (fins and sonar) have more dense points.*

Analytical integration of (1) is impossible for arbitrary hull shape and arbitrary sea surface. The solution is to divide the submerged hull surface into small surface elements [YHK07], but fixed regular or random discreet dividing will lead to non-compensated forces and constant drift especially on silent water. To avoid this effect, we uniformly (within triangles) place several hundreds random points on the naval object hull at each simulation step. Each point represents a surface element with a particular area and normal. Surface elements are considered to be so small that a change of pressure or force along these elements is negligible. Figure 1 shows an example of point distribution for a submarine-like naval object.

DOI: 10.2312/egp.20161044

To obtain water height-, offset- and velocity fields we use fast Fourier transform [T*01] with Pierson-Moskowitz spectra [PM64].

To compute force acting on each surface element we determine whether each element is submerged. If the element is submerged, we obtain wave height and water velocity at the centre of the element and the absolute velocity of this element. We compute the hydrostatic force from the wave height above a given point. The hydrodynamic force is computed as the sum of the drag ($\mathbf{F}_{drag}$) and lift ($\mathbf{F}_{lift}$) forces acting on given surface element. Drag and lift forces are computed as follows (2).

$$\mathbf{F}_{drag} = \frac{1}{2}\rho C_{drag} S \mathbf{u}^2 \quad \mathbf{F}_{lift} = \frac{1}{2}\rho C_{lift} S \mathbf{u}^2 \tag{2}$$

Where ρ — water density, $\mathbf{u}$ — incoming flow velocity, S — surface element area. Coefficients C_{drag} and C_{lift} depend on the angle α between surface element normal and negated velocity vector. We assume $C_{drag} = a\cos(\alpha) + b$ and $C_{lift} = c\sin(2\alpha)$. Coefficients a, b and c can be estimated experimentally. For our simulations we chose $a = c = 1$ and $b = 0.1$ which were found to provide believable motion damping.

In the case of Gerstner's waves, an extra effort to prevent visual detachment of floating body and wave is required [Ger52], especially for that naval objects are small relative to wave-length (like buoys). This problem is solved using iterative search along the wave gradient.

The proposed method is implemented using DirectCompute. Rigid body simulation is performed using the BEPUphysics engine. Ocean simulation is performed on GPU. Surface elements and theirs instantaneous velocities are computed on each step on CPU and then copied to GPU memory. When the simulation step is complete forces are copied back to system memory and applied to a rigid body through the interface of the physics engine.

Due to the random nature of surface elements the forces acting on a floating body differ slightly between simulation steps. This difference produce numerical drift and floating body slowly moves even on silent water. Experiments show that drift velocity does not exceed 1–2 m/min and depends on the number of the surface elements (500–1000 is sufficient) and slightly on the size of the floating object.

3. Conclusion

This method has linear scalability and does not depend on number of floating bodies. It depends only on total amount of surface elements used during simulation. This method efficiently handles large objects like ships and vessels, as well as small ones, like buoys or naval mines. See figure 2.

Future extensions of our approach include reducing readbacks from GPU, particle generation for splash simulation and attempting to reproduce secondary waves and turbulent flows around of floating body.

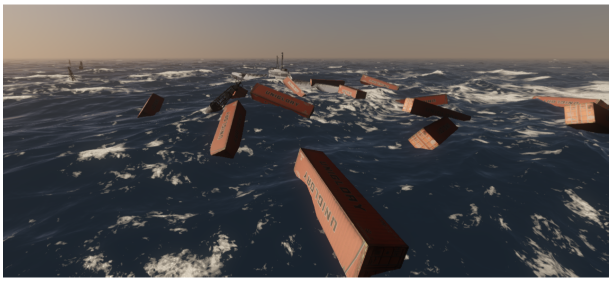

Figure 2: *More than 20 simulated naval objects.*

References

[BDB98] BELENKY V. L., DEGTYAREV A. B., BOUKHANOVSKY A. V.: Probabilistic qualities of nonlinear stochastic rolling. *Ocean Engineering 25*, 1 (1998), 1 – 25. 1

[BE14] BEZGODOV A., ESIN D.: Complex network modeling for maritime search and rescue operations. *Procedia Computer Science 29* (2014), 2325–2335. 1

[CM11] CHENTANEZ N., MÜLLER M.: Real-time eulerian water simulation using a restricted tall cell grid. In *ACM SIGGRAPH 2011 Papers* (New York, NY, USA, 2011), SIGGRAPH '11, ACM, pp. 82:1–82:10. 1

[FGS] FONSECA N., GUEDES SOARES C.: Comparison of numerical and experimental results of nonlinear wave-induced vertical ship motions and loads. *Journal of Marine Science and Technology 6*, 4, 193–204. 1

[Ger52] GERSTNER F. J.: *Theory of waves*. University of California, Institute of Engineering Research, 1952. 2

[GXW04] G. X. WU Z. Z. H.: Simulation of nonlinear interactions between waves and floating bodies through a finite-element-based numerical tank. *Proceedings: Mathematical, Physical and Engineering Sciences 460*, 2050 (2004), 2797–2817. 1

[MM13] MACKLIN M., MÜLLER M.: Position based fluids. *ACM Trans. Graph. 32*, 4 (July 2013), 104:1–104:12. 1

[PM64] PIERSON W. J., MOSKOWITZ L.: A proposed spectral form for fully developed wind seas based on the similarity theory of sa kitaigorodskii. *Journal of geophysical research 69*, 24 (1964), 5181–5190. 2

[T*01] TESSENDORF J., ET AL.: Simulating ocean water. *Simulating Nature: Realistic and Interactive Techniques. SIGGRAPH 1*, 2 (2001), 5. 2

[ULR13] ULRICH C., LEONARDI M., RUNG T.: Multi-physics sph simulation of complex marine-engineering hydrodynamic problems. *Ocean Engineering 64* (may 2013), 109–121. 1

[YHK07] YUKSEL C., HOUSE D. H., KEYSER J.: Wave particles. In *ACM SIGGRAPH 2007 Papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. 1