Towards Real-Time Simulation of the Sidescan Sonar Imaging Process

JAMES RIORDAN, DANIEL TOAL & COLIN FLANAGAN
Department of Electronic and Computer Engineering
University of Limerick
Limerick
IRELAND

Abstract: - This paper describes the functional theory and design of a modular simulator developed to generate physically representative spatio-temporal sidescan sonar echo data from a fractal model of the seafloor topography. The main contribution of this paper is in significantly reducing the computational bottleneck inherent in existing simulation models due to the size and resolution of the complex seafloor models required for acoustic reverberation modelling. Discovery of the individual faces within the footprint of the acoustic beam at each ping is considerably accelerated by successfully adapting and integrating an optimised mesh refinement scheme intended for interactive rendering of large-scale complex surfaces described by polygonal meshes. Operational features of the simulator permit direct visualisation of the sonar image formed from successive echo lines and synthetic images generated during simulation are presented.

Key-Words: - sidescan sonar simulation, fractals, seafloor modelling, mesh optimisation, ray tracing

1 Introduction

Analysis of the geometric and intensity distortions inherent in sidescan sonar images and the development and validation of correctional techniques and algorithms ideally requires ground-truth seafloor data to compare with the reconstructed imagery [1]. Unfortunately due to the harsh nature of the underwater environment this data is often difficult to obtain. A simulator providing realistic echo data from an accurate representation of the seafloor topography and roughness is thus an invaluable tool in the development and testing of methodologies for processing real sidescan sonar datasets. Existing sidescan sonar simulation models, using alternative methods to solve the governing wave equation, have produced visually realistic results that have been verified using both statistical and spectral features, notably [2,3].

To model the seafloor at a level of realism sufficient for acoustic reverberation modelling, highly detailed large-scale meshes are required. The resulting complex surfaces present significant computational challenges due to the inherent bottleneck imposed in locating the individual faces within the footprint of the simulated acoustic beam. This drawback can often restrict the use of existing simulators to offline applications and prohibit their use in areas such as real-time AUV localisation, navigation and mapping simulation.

Recent advances in the field of 3D graphics rendering have enabled interactive visualisation of realistic and highly detailed virtual environments consisting of millions of polygons. It is our observation that the schemes used to optimise surface mesh representations, resulting in greatly accelerated rendering rates, are directly applicable to the field of underwater acoustic simulation. In this paper, we describe results from the ongoing development of a simulator to produce realistic sidescan sonar imagery from a three dimensional underwater scene. By successfully adapting and integrating an optimised mesh refinement scheme we have achieved a noticeable increase in simulation rates.

The remainder of the paper is organised as follows. Section 2 details the method used to generate a representative model of the seafloor. In section 3, we describe the optimised mesh refinement scheme implemented to represent the topography of the resulting seafloor terrain. Section 4 outlines the acoustic propagation model and ray-tracing algorithm used while the seafloor scattering model employed is described in section 5. We present synthetically generated sidescan images that focus on ocean floor mapping in Section 6. Brief discussion and conclusions are given in Section 7.

2 Fractal Seafloor Model

Naturally occurring objects are irregular and are difficult to model using the smooth shape primitives of Euclidean geometry. Fractals however have a non-integer dimension, which quantifies the roughness and the inherent irregularity of natural objects, and are often the result of recursive algorithms that modify shape through local actions; comparable to physical natural processes such as erosion or aggregation [4].

Fractal models provide a controllable method of generating visually and statistically realistic representations of seafloor topographies and textures within certain prescribed criteria. The geometry of the resulting surface can be represented as a polygonal mesh, permitting the scattering of the acoustic wave by the seafloor interface to be determined from the local geometry and texture of the mesh within the acoustic footprint. Our simulator employs the fractal model of fractional Brownian motion (fBm) to represent the seafloor. The power law relationship of fBm has been verified to provide good representations of rough seabed surfaces over a range of scales, from centimetre resolution to several kilometres [5]. In two dimensions, a fractional Brownian function b(x,y) has zero-mean Gaussian increments with variance:

$$\begin{cases} E[b(x+\delta, y+\varepsilon) - b(x,y)]^2 = \sigma^2 x\, d^{2H} \\ with\ d = \sqrt{\delta^2+\varepsilon^2} \end{cases} \quad (1)$$

where σ^2 is the variance of a zero-mean Gaussian distribution, H is the Hurst coefficient and is related to the fractal dimension D of b(x,y) by: D = 3 – H with 0 < H < 1.

Both isotropic and directional fractal surfaces can be generated using the method of spectral synthesis of fBm. The 2D fBm defined by equation 2 has a power spectral density:

$$\Gamma(u,v) = \frac{1}{(u^2+v^2)^{\beta}} \ with\ \beta = 2H+1\ (0 < H < 1) \quad (2)$$

By applying a filter varying in:

$$\frac{1}{(u^2+v^2)^{\beta/2}} \quad (3)$$

to the Fourier series of a 2D Gaussian white noise function the resulting $N \times N$ fractal heightfield can be obtained via the inverse fast Fourier transform. Isotropic fractal surfaces are generated if the power spectrum follows an inverse power law throughout its frequency range (fig. 1). Directional fractal surfaces are synthesised by shifting the maximum power from zero frequency (fig. 2).

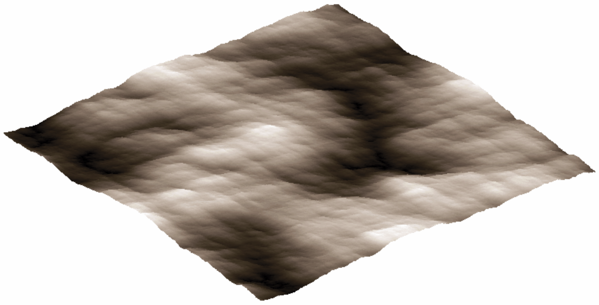

Fig. 1, Isotropic fractal heightfield displaying a scene representative of a rocky seafloor.

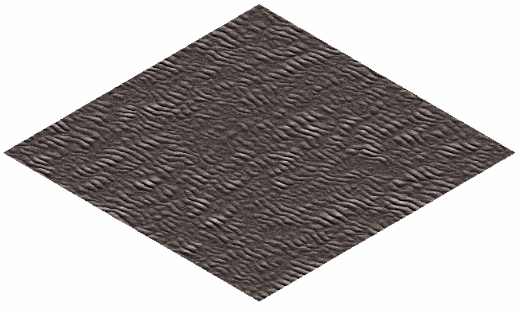

Fig. 2, Sand rippled seafloor mesh formed by a directional fractal field.

3 Selective Refinement of Seafloor Mesh Geometry

To model the seafloor at a level of realism sufficient for acoustic reverberation modeling, highly detailed large-scale meshes are required. Triangle meshes are the most widely used due to their mathematical simplicity and flexibility. The resulting complex surfaces, often including millions of triangles, present significant computational challenges when real time processing of the underlying faces is desired. Our acoustic simulation models uses ray theory and adopts ray-tracing techniques from 3D graphics rendering whereby the acoustic beam pattern is described by the geometry of a ray bundle. Discovery of the faces intersected by the ray bundle, to investigate their contribution to the level of energy received at the transducer face and to resolve the time-of-flight of the acoustic ping, proves to be the main computational bottleneck. Similar to the demands of interactive rendering of realistic virtual environments, the influences on the classic speed vs. memory trade off between highly realistic surface models and the computational demands imposed by these models ultimately determine the simulation rates achievable.

In 3D visualization many faces of the model are either orientated away from the viewer or lie outside the “cameras” view frustum and thus do not contribute to the final image displayed. While these faces are typically culled early in the rendering pipeline, this processing incurs a cost. Likewise, in modelling the sidescan sonar imaging process a simulated acoustic ping will generate a footprint on the seafloor mesh with a coverage encompassing only a small number of faces in proportion to the scale of the overall mesh. Usually the majority of faces on the mesh lie outside the acoustic beam footprint and do not contribute any backscatter energy to the recorded intensity. For both applications it is desirable to optimize scene representations and ray tracing algorithms to avoid unnecessary intersection tests whenever possible.

Several schemes have been developed in research to address the challenges involved in processing and rendering large-scale polygonal meshes at interactive rates. Mesh simplification techniques and multiresolution representations are primarily concerned with the visual appearance of the rendered models and optimise surface representations by replacing the original surface mesh with nearly indistinguishable approximations containing far fewer faces, subject to a maximum allowable projected screen space error [6]. Adaptive refinement schemes in contrast, permit on-the-fly view-dependant level-of-detail (LOD) control by locally adapting the surface geometric complexity to changing viewing parameters. The resulting mesh is a combination of mesh subsets covering the entire original mesh at different resolutions. Faces not contributing to the final rendered image are culled before reaching the rendering pipeline thus greatly accelerating rendering performance while preserving the integrity of the final image displayed to the viewer.

If the planes describing the view frustum of the "camera" implemented in 3D graphics rendering are reconfigured to describe the surfaces of the bounded volume of the sonar fan-beam, view-dependant level-of-detail control provides an effective mechanism for reducing the number of faces to be interrogated at each ping. With the area of the sonar footprint usually being many magnitudes smaller than the scale of the overall high resolution mesh and the high ping rates characteristic of high frequency imaging sonar's ensuring some footprint overlap at each ping, the computational overhead involved in continuously refining the mesh is more than compensated by the resulting improvement in efficiency in discovering the faces within the acoustic footprint.

3.1 Progressive Meshes

Adaptive refinement schemes usually build upon the framework of the *progressive mesh* (PM), which was first introduced by Hoppe [7]. A PM is a multiresolution representation of an irregular mesh whereby a vertex hierarchy of an arbitrary mesh $\hat{M}$ is constructed by simplifying, through a sequence of *n edge collapse (ecol)* transformations, to a base mesh M^0 of lowest possible resolution:

$$(\hat{M} = M^n) \xrightarrow{ecol_{n\text{-}1}} \ldots \xrightarrow{ecol_1} M^1 \xrightarrow{ecol_0} M^0 \qquad (4)$$

In each *ecol(v_{si}, v_{ti}, v_{ui}, v_{li}, v_{ri})* edge collapse transformation, the edge $e_{ti/ui}$ is collapsed to a new vertex v_{si} and the *parent-children relationship* is established between the resulting vertex v_{si} and original vertices v_{ti}/v_{ui}. After n successive edge collapse transformations on the given mesh $\hat{M}$, a base mesh M^0 and the vertex hierarchy H required to restore M^0 back to $\hat{M}$ is obtained. The root nodes of the hierarchy H are the vertices of the base mesh M^0 and the leaf nodes of H are the vertices of the original mesh $\hat{M}$ (fig. 3).

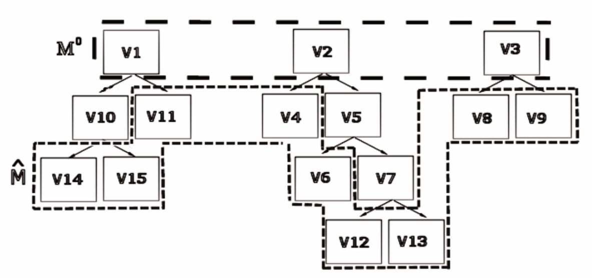

Fig. 3, The branching structure of the vertex hierarchy of a Progressive Mesh.

Because edge collapse transformations are invertible, an arbitrary triangle mesh $\hat{M}$ can be represented as a simplified mesh M^0 together with a sequence of n vertex split (*vsplit*) records:

$$M^0 \xrightarrow{vsplit_0} M^1 \xrightarrow{vsplit_1} \ldots \xrightarrow{vsplit_{n\text{-}1}} (M^n = \hat{M}) \qquad (5)$$

Constructing an adaptive mesh is time consuming if an accurate representation is demanded. Consequently the computationally intensive operations are best-performed offline during the *analysis phase* by a decimation or refinement process which reserves the mesh M^0 and the modifications performed. As the *detail information* for recovering M^{i+1} from M^i is reserved, the structure can then be queried efficiently online during the *synthesis phase* to obtain adaptive meshes on the fly by reconstructing the vertex dependency relations. Given a level of detail query, in our case a criterion based on the view frustum defining the bounded volume of the sonar fan beam, an adaptive refinement process will extract the mesh M^n of minimum size from the original mesh $\hat{M}$ satisfying the query.

3.2 Selective Refinement Scheme

Our model implements the selective refinement scheme of Kim and Lee [8]. The source code to implement this scheme is available in OpenMesh, a mesh representation and manipulation library [9]. Their approach to representing the detail of a mesh eliminates the need for possible additional mesh modifications that might otherwise be required to preserve the local mesh connectivity when a vertex is split or an edge is collapsed. This significant improvement in efficiency is due to the concept of the *dual piece*.

The dual of a polygonal mesh is another mesh with the same topography but different connectivity, constructed by assigning a dual vertex to each face (computed as the barycentre or centroid (centre of mass) of the vertices that support the corresponding face). A pair of dual vertices is then connected with a dual edge if the corresponding faces shared an edge in the original mesh. The *fundamental dual piece* of a vertex $\hat{v}$ in the original mesh $\hat{M}$ is thus defined as the closed region over $\hat{M}$ which is surrounded by the dual edges that connect the dual vertices assigned to the faces adjacent to $\hat{v}$. Furthermore, the *dual piece* $\mathbb{D}$(v) of an internal node vertex v in the vertex hierarchy H is defined as the union of the fundamental dual pieces of all leaf nodes in the subtree of H whose root is v.

For a selectively refined mesh M^i consisting of the set of vertices V from the vertex hierarchy H, the dual pieces of V cover the original mesh $\hat{M}$ without overlap or holes. An edge between a pair of vertices in M^i emerges as an adjacency between the corresponding dual pieces of the original mesh $\hat{M}$ and the dual pieces of all vertices in the vertex hierarchy H adhere to the following rules.

- $\mathbb{D}$ (v_{si}) = $\mathbb{D}$ $(v_{ti}) \cup$ $\mathbb{D}$ (v_{ui}) and $\mathbb{D}$ $(v_{ti}) \cap$ $\mathbb{D}$ (v_{ui}) = $\varnothing$ for all i.
- $\mathbb{D}$ (v_{ti}) and $\mathbb{D}$ (v_{ui}) are adjacent to each other, for all i.
- $\mathbb{D}$ $(v_q) \subset$ $\mathbb{D}$ (v_p) if v_p is an ancestor of v_q in H.
- $\mathbb{D}$ $(v_q) \cap$ $\mathbb{D}$ (v_p) = $\varnothing$ if v_p and v_q have no ancestor-descendent relationship in H.

Using the concept of the dual piece, Kim and Lee designed improved *vsplit* and *ecol* transformations, re-tiling a piece $\mathbb{D}$ (v_{si}) with two adjacent pieces $\mathbb{D}$ (v_{ti}) and $\mathbb{D}$ (v_{ui}) for *vsplit* transformations and merging two adjacent pieces $\mathbb{D}$ (v_{ti}) and $\mathbb{D}$ (v_{ui}) to form $\mathbb{D}$ (v_{si}) for *ecol* transformations. This locally confined refinement is achieved by discovery of the two vertices v_{li} and v_{ri}, termed the *cut vertices* of the vertex v_{si}, which determine the connectivity between the new vertices v_{ti}/v_{ui} and the 1-ring neighbourhood $N(v_{si})$ of the original split vertex v_{si} after a *vsplit* transformation.

As the 1-ring neighbourhood of v_{si} can change dynamically in the synthesis phase, caused by the local refinement of the mesh in the proximity of the acoustic footprint, the active vertices in the 1-ring neighbourhood of v_{si} are not guaranteed to have the same configuration as when *ecol*$(v_{si}, v_{ti}, v_{ui}, v_{li}, v_{ri})$ was performed during the analysis phase. From the dual perspective however the vertices v_{li} and v_{ri} will always exist and are the only vertices in the 1-ring neighbourhood of v_{si} whose dual pieces are consecutively adjacent to both $\mathbb{D}$ (v_{ti}) and $\mathbb{D}$ (v_{ui}). To find v_{li} and v_{ri} on-the-fly during simulation their corresponding vertices in the original mesh $\hat{M}$, termed the *fundamental cut vertices* of v_{si}, must have previously been located and stored during the analysis phase. In the analysis phase two faces $f_l = \Delta(v_{ti}, v_{ui}, v_{li})$ and $f_r = \Delta(v_{ti}, v_{ui}, v_{li})$ are collapsed by *ecol*$(v_{si}, v_{ti}, v_{ui}, v_{li}, v_{ri})$. Referring to the branching structure of the vertex hierarchy H if each internal node vertex constructing f_l is relabelled as the active ancestor of the vertex at the leaf node of the same branch it follows that these leaf node vertices will also form a triangle in $\hat{M}$. The face f_l active in M^i is therefore represented as $(\hat{v}_{ti}, \hat{v}_{ui}, \hat{v}_{li})$ in $\hat{M}$. The dual piece of $\hat{v}_{li}$ has adjacency to the dual pieces of both $\hat{v}_{ti}$ and $\hat{v}_{ui}$ as they form the same triangle in $\hat{M}$. Given the previous rule, $\mathbb{D}$ $(v_q) \subset$ $\mathbb{D}$ (v_p) if v_p is an ancestor of v_q in H, the active ancestors of the corresponding leaf node vertices constructing f_l will share the same relationship.

The fundamental cut vertices of v_{si}, the active vertices of the collapsed triangle in M^i and the corresponding leaf node vertices representing the same triangle in $\hat{M}$ are stored after each *ecol* performed in the analysis phase. The cut vertices v_{li} and v_{ri} of v_{si} can then be located for a *vsplit* during the synthesis stage by ascending the vertex hierarchy from the leaf nodes containing each fundamental cut vertex until an active vertex in the current mesh is located. An *ecol* during the synthesis stage simply involves locating the vertices opposite to the edge $e_{ti/ui}$ as these vertices complete the triangles to be collapsed and already have dual pieces adjacent to $\mathbb{D}$ (v_{ti}) and $\mathbb{D}$ (v_{ui}).

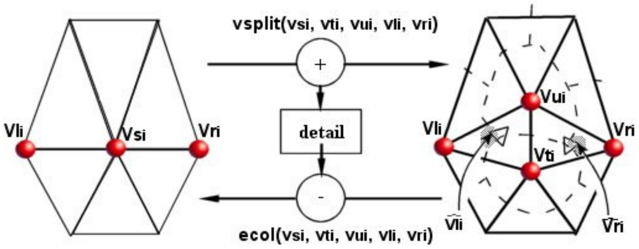
vsplit(vsi, vti, vui, vli, vri)
+
detail
-
ecol(vsi, vti, vui, vli, vri)
Vli
Vsi
Vri
Vui
Vti
Vli
Vri
Vli
Vri

Fig. 4, Definitions for *vsplit* and *ecol* transformations

3.3 View Dependant Refinement Implementation

For acoustic simulation the selective refinement process is controlled both by the geometry of the frustum bounding the ray bundle and the range of the sonar. When the mesh is first loaded, the radius of the sphere centered at each vertex and bounded by the region on the mesh enclosing that vertex and all its descendants in stored. During simulation the parameters of the sonar view frustum are re-calculated at each ping and each active vertex within range is interrogated to determine whether its bounding sphere lies completely outside the view frustum; if the refinement query returns true (descendents of the vertex lay within the view frustum but are not active in the present mesh) then the vertex is split (if it is not already a leaf node vertex), otherwise an edge collapse is attempted.

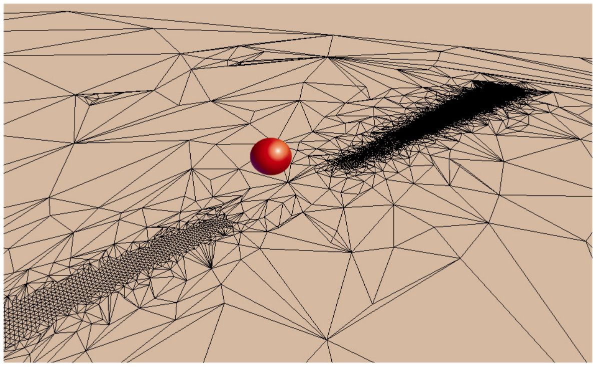

Fig. 5, Selective refinement for a sidescan sonar at a position indicated by the spherical object.

Multiple sonars (e.g. a bundle of obstacle avoidance sonars in AUV vehicle control simulation) can be implemented by locally refining the mesh for each sonar at the start of a ping while maintaining the required flags to preserve the footprints of the previous sonar(s).

4 Acoustic Beam Propagation Model

The theoretical basic underlying all mathematical models of acoustic propagation is the wave equation. Using certain simplifying assumptions from ray theory the wave equation reduces to the Helmholtz equation [10]:

$$\nabla^2 \phi + k^2 \phi = 0 \qquad (6)$$

where $k = (\omega / c) = (2\pi / \lambda)$ is the wavenumber and λ the wavelength. Rather than incrementally tracing each Helmholtz ray through the water column, incurring intersection tests at each step, it was found that over the operating range of high frequency sonar the faces intersected by a Helmholtz ray and the corresponding straight-line ray deviate little. When the simulation is first initialised, the Plücker Coordinates of all edges supporting the mesh $\hat{M}$ are stored. For a line passing through points p and q its Plücker coordinates are the determinants of each 2x2 minor of the matrix [11]:

$$\begin{bmatrix} p_x & p_y & p_z & 1 \\ q_x & q_y & q_z & 1 \end{bmatrix} \qquad (7)$$

Given two directed lines, the permuted inner product of their Plücker Coordinates will indicate which side one directed line is on with respect to the other. As the edges of a triangle are directed (to determine the direction of the face normal), a ray must pass the same side of each edge to intersect a triangular face (checking against two edges first can often provide a quick rejection test).

For this reason straight line rays are first launched with the same launch angle as the Helmholtz rays and intersection tests performed using the pre-stored Plücker Coordinates. Intersecting rays exceeding the range of the sonar are subsequently discarded. The intersection test for each Helmholtz ray then uses these faces as starting points, expanding outwards until the intersection is located. Once the intersecting facet is located the grazing angle is determines from the face normal.

5 Seafloor Scattering Model

To calculate the amount of energy reflected back to the transducer by the seafloor water-sediment interface Jackson's backscattering model is used [12]. The Kirchhoff approximation, composite roughness and sediment volume models are interpolated to account for the various contributions to bottom backscattering at high frequencies. The bottom backscattering strength $S_b(\theta)$ is the dB equivalent of the scattering cross section, determined from:

$$S_b(\theta) = 10\log_{10}[\sigma_r(\theta) + \sigma_v(\theta)] \qquad (8)$$

$\sigma_v(\theta)$ is the dimensionless backscattering cross-section per unit solid angle per unit area due to a portion of the acoustic energy penetrating the interface sediment and being reflected back by heterogeneities included in the sediment volume. $\sigma_r(\theta)$is the dimensionless backscattering cross-section per unit solid angle per unit area due to the interface roughness and is calculated by the Kirchoff approximation at incidence close to the vertical to account for specular backscattering and by the composite roughness model at grazing incidence.

6 Experimental Results

The resulting synthetic images faithfully reproduce artefacts characteristic of certain seafloor geometry's e.g. sand ripples. A sample image generated from the rippled fractal seafloor model of fig. 2 is displayed in fig. 6. The images appear visually realistic with the familiar highlighting followed by shadowing of the ripples having a distribution and form spatially consistent with the geometry of the simulated seafloor mesh. The simulator was run on a 2.5 GHz Pentium IV system. Initial performance results indicate a significant improvement in computational efficiency due to the integration of the selective refinement scheme, with processing times of 1-3 pings per second being achievable for large-scale complex terrain meshes consisting of 1.5-2 million faces. Simulation rates increase greatly for small-scale complex meshes leading the authors to believe that partitioning the initial mesh into clusters of progressive meshes has the potential to further improve performance characteristics.

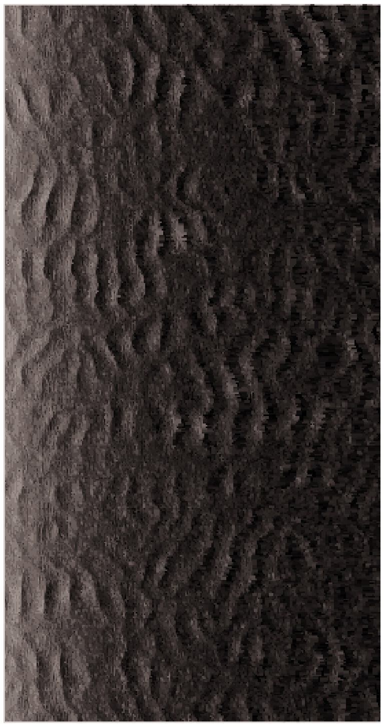

Fig. 6, Synthetic sidescan image of a rippled seafloor

7 Conclusions and Future Work

In this paper we presented the successful integration of an optimised scheme to perform view dependant selective refinement of polygonal meshes with an existing sidescan sonar simulator. As the number of active faces in mesh during simulation is orders of magnitude less than the scale of the complete high resolution mesh, the computation bottleneck involved in discovering the faces within the footprint of the acoustic beam is significantly reduced. Due to the resulting performance increases, integrating the sonar simulator with an existing AUV simulator developed within the research group, is currently being investigated. This integration will enable on-the-fly sensor data generation while flying the vehicle and thereby the investigation of the performance of sonar based vehicle control strategies.

References

[1] Hunter, A.J., Hayes, M.P., Gough, P.T., *Simulation of multiple-receiver, broadband interferometric SAS imagery,* OCEANS 2003, vol.5, 2003, pp. 2629- 2634

[2] Bell, J.M., Linnett, L.M., *Simulation and analysis of synthetic sidescan sonar images,* Inst. Elect. Eng. Radar, Sonar and Navigation, vol. 144, 1997, pp. 219-226.

[3] Elston, G.R.; Bell, J.M.; *Pseudospectral time-domain modeling of non-Rayleigh reverberation: synthesis and statistical analysis of a sidescan sonar image of sand ripples,* Oceanic Engineering, IEEE Journal of, Vol.29, 2004, pp. 317- 329

[4] Guillaudeux, S., Maillard, E., *Seabed simulations with fractal techniques,* OCEANS 1998, Vol.1, 1998, pp. 433-437

[5] Briggs,K. B., *Microtopographical roughness of shallow-water continental shelves*, Oceanic Eng., IEEE Journal of, vol. 14, 1989, pp. 360–367.

[6] De Floriani, L., Magillo, P., *Multi-resolution Mesh Representations - Models and Data Structures*, European School on Multiresolution Geometric Modeling, 2001.

[7] Hoppe H., *View-dependant refinement of progressive meshes*, Computer Graphics & Interactive Techniques, 1997, pp. 189-198.

[8] Kim J., Lee S., *Truly selective refinement of progressive meshes,* No Description On Graphics Interface Conference, 2001, pp.101-110.

[9] http://www.openmesh.org.

[10] Etter P. C., *Underwater Acoustics Modelling and Simulation: Principles, Techniques and Applications*, Spon Press, 2003.

[11] Amanatides J, Choi K., *Ray tracing triangular meshes,* 8th Western Computer Graphics Symposium, 1997, pp 43-52.

[12] Mourad, P.D., Jackson, D.R., *High frequency sonar equation models for bottom backscatter and forward loss,* Oceans 1989, 1989, pp. 1168-1175.