IMMERSIVE, LOW-COST TRAINING WITH ENVIRONMENTAL EFFECT SDK'S

PREPARED BY FRANK KANE, CEO
SUNDOG SOFTWARE LLC
FKANE@SUNDOG-SOFT.COM
HTTP://WWW.SUNDOG-SOFT.COM/

ABSTRACT:

Immersive and physically realistic synthetic environments are no longer restricted to high-end image generators, visual systems, and game engines.   Add-ons for open-source and low-cost graphic engines are commercially available for skies, volumetric clouds, weather effects, oceans, and atmospheric scattering. These products meet or exceed the fidelity and visual quality of current technology designed primarily for the simulation and training industry. This paper examines current capabilities of real-time environmental effect SDK's, and a high level overview of the techniques employed to create virtual environments that are both realistic enough for training, and photorealistic enough for modern video games.

LOW-COST SOLUTIONS FOR ACCURATE, IMMERSIVE ENVIRONMENTAL EFFECTS

Figure 1: Sundog's Triton Ocean SDK and SilverLining Sky SDK in a Real-Time Scene

Technology may be licensed today for unlimited use in a training program for under $10,000 that provides the following capabilities at 200 Hz on consumer-grade hardware:

- 3D ocean simulation of over 65,000 simultaneous waves adhering to the JONSWAP spectra
- Physically-realistic time of day effects and skies for any time, location, and weather
- 3D volumetric clouds suitable for flight training
- Perceptual tone-mapped scattered and direct sun and moonlight simulation
- Simulation of ship wakes, water impacts, and helicopter rotor wash

Costs are low as this technology is also marketed to the larger game developer market. It is provided as software development kits (SDK's), or code libraries that developers incorporate into their own graphics engines. Popular engines such as OpenSceneGraph, Ogre, Havok, and Unity are supported – many of which are free or low-cost.

CURRENT OCEAN SIMULATION CAPABILITIES

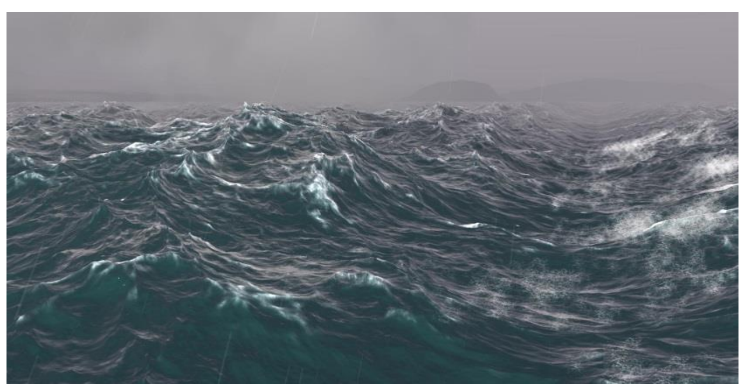

Figure 2: The Triton Ocean SDK Simulating 262,144 waves at once

Due to advances in general-purpose GPU computing, it is now possible to use the same ocean rendering techniques used in films like *Titanic* in real-time (1), at hundreds of frames per second. Recent advances in modeling and simulation have extended these entertainment-focused algorithms to be suitable for training (2). Today's environmental effect SDK's can simulate over 250,000 individual ocean waves at once using the massively parallel computing capabilities of consumer-grade video cards, using wave spectra used for maritime training such as JONSWAP (3). These spectra are capable of capturing the effects of wind fetch length on wave conditions, and accurate simulation of Beaufort scales 1-9 as well as Douglas Sea Scales that specify swell conditions independent of local wind waves. In addition, user-defined swell waves or existing wave models may be integrated with these visual systems.

Previous technology would approximate these effects by adding together a small number of sinusoidal waves, and overlaying a pattern of high-frequency waves over them, but this resulted in unrealistic motion of ships, buoys, and other objects that rely on accurately modeled wave heights over time. By modeling every wave individually across the entire frequency spectrum, buoyancy models may be developed that react to real wave conditions instead of approximating them with simple sinusoidal motion.

Modern techniques employ a mathematical tool called the inverse discrete Fast Fourier Transform (FFT) to simulate such a large number of waves at once. Technologies such as CUDA, OpenCL, and DirectX11 Compute Shaders may be used to solve these FFT's over the many cores available on current video cards, and feed the results directly to the shaders that control the final position and lighting of every point on the water surface. A real-time simulation of an infinite ocean simulating the JONSWAP spectra with 65,535 waves was observed running at over 700 frames per second using an NVidia GTX 770 graphics card using our Triton Ocean SDK. Expecting this level of fidelity without compromising 60Hz update rate requirements is reasonable today using consumer hardware.

Although recent research in 3D ocean simulation is motivated in part by popular video games, 3D ocean SDK's are available with features specifically important to training and simulation. The following requirements may be met today with off-the-shelf software development kits:

- Accurate FFT simulation of Beaufort and Douglas Sea Scale states 1-9
- Ship wakes with 3D Kelvin wake displacements, turbulent propeller wash, bow wakes, and spray effects
- Particle-based wind spray
- Foam effects at and behind wave crests
- Individual swell waves modeled in addition to wind waves
- JONSWAP and Pierson-Moskowitz wave spectra (realistic wave frequency and amplitude distributions)
- Ability to model the effects of wind speed and wind fetch length on waves
- Smooth blending and modeling of water / coastline boundaries
- Support for geocentric / ECEF coordinate systems that encompass an entire planet
- Fast height queries for powering buoyancy models
- Decal textures that move with the water surface (for kelp, oil slicks, debris, etc.)
- Rotor wash and impact effects that displace the water surface
- Reflections of objects on the water and of the environment, and simulation of refraction
- Proper handling of visibility and lighting of the water surface above, below, and at sea level

Sundog Software's Triton Ocean SDK fulfills all of these requirements, and competing technologies such as Antycip's MyOcean3D, osgOcean, Hydrax, and NVidia's WaveWorks meet various subsets of them. These other products often only work with a very specific engine or specific set of hardware, but are worth evaluating for a given development environment and set of training requirements. In addition to ocean simulation, these

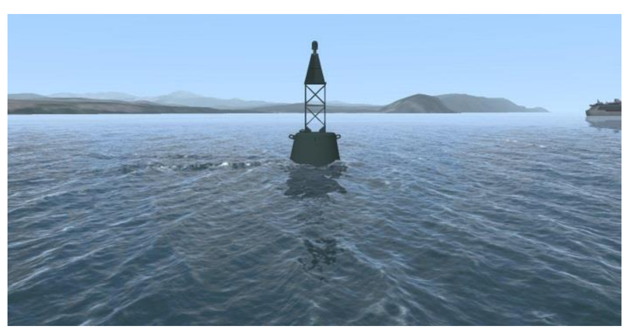

Figure 3: The Triton Ocean SDK Powering Buoy Motion

toolkits in some cases may also be used for lakes, ponds and even rivers.

DYNAMIC SKIES AND TIME OF DAY EFFECTS

Figure 4: Hosek-Wilkie vs. Preetham Sky Colors

Accurate lighting is crucial to synthetic environments intended to simulate a specific time and place. Recent computer graphics research has advanced the ability to represent realistic skies for any given time, date, location, and weather conditions. In 2012, the Hosek-Wilkie sky model was presented (4), which is powering procedural sky boxes in training and simulation today with unprecedented realism. Hosek-Wilkie extends upon the Preetham model (5) used previously, offering a more accurate representation of the sky with lower solar altitudes and in the area surrounding the sun. As seen in Fig. 4, the improvement can be striking. Today's environmental effect SDK's are capable of executing the Hosek-Wilkie model in real time at hundreds of frames per second, even in simulations that track real time and are always modifying the simulated time. One downside of the Hosek-Wilkie model is that it cannot simulate conditions past sunset, and so it must be combined with the Preetham model and models of twilight conditions in real applications that simulate a full 24 hour cycle. Our SilverLining SDK does exactly that.

Nighttime lighting in visible wavelengths is particularly challenging to properly represent (6). The actual luminance at night is several orders of magnitude less than that during the day, and linearly mapping this luminance to display values would result in pitch-black nights in synthetic environments with zero visibility. However, the human eye can adapt to low light conditions, assuming the presence of at least some moon, star, or artificial light. We employ a technique called perceptual tone mapping to capture this effect, using the Durand tone-mapping operator (7).

Correct lighting depends on a good ephemeris model, capable of capturing the correct position of the sun and moon as well as the phase of the moon. Correct positioning and movement of the visible stars and planets may also be important in some applications. This ephemeris model must balance accuracy with speed, as changing the time of day should not result in significant performance degradation.

The other piece of the puzzle is simulating the amount of scattered and direct light used to illuminate the rest of the scene, based on the time of day and location (8). In our SDK, we employ the Bird simplified clear sky model (9), which provides a good tradeoff between accuracy and speed of computation. This captures the effects of Rayleigh and Mie scattering, and models things such as the amount of ozone and water vapor present. Other products, such as from JRM Technologies, provide full MODTRAN simulation where the most accurate solar radiance values are needed. Accurate lighting of a synthetic natural scene depends on things such as proper modeling of twilight, refraction of the sun near the horizon, the effect of cloud cover, ground albedo, atmospheric turbidity, and choices made in tone-mapping the final results – perhaps more so than the underlying scattering model used. These effects are all modeled in current environmental effect SDK's.

To summarize, the following requirements may be met by today's sky SDK products:

- Procedurally generated sky for any given time, location, and date, with real-time adjustment of time
- Atmospheric scattering simulation resulting in direct and scattered solar and lunar radiation
- Modeling of the effects of water vapor, ground albedo, ozone, Mie scattering, and Rayleigh scattering
- Real-time ephemeris model time with <1 degree error of sun, moon, star, and planet positions
- Modeling of the moon phase and its effect on scene lighting
- Perceptual tone-mapping to simulate human eye adaptation to low light conditions
- Continuous simulation of the day/night cycle including twilight and the effects of atmospheric refraction
- Ability to operate in arbitrary coordinate systems such as geocentric / ECEF

Sundog Software's SilverLining Sky, 3D Cloud, and Weather SDK meets these requirements. Competing products, such as osgEphemeris, trueSky by Simul, and Ephemeris by Confetti, also deserve evaluation.

CURRENT APPROACHES TO 3D VOLUMETRIC CLOUDS

While most games use pre-created sky box artwork to represent clouds in a given level, this approach is not sufficient for training applications where different weather conditions need to be simulated. A procedural approach to representing cloud cover is needed, and there are many to choose from in today's environmental effect SDK's (10).

Figure 5: The SilverLining SDK used in NASA's OBVA program

One approach is GPU ray-casting, where a 3D volumetric texture representing a cloud layer is created on demand for a desired coverage amount. The application renders a bounding volume for the cloud layer, and fragment shaders are employed to shoot rays through each screen pixel through the volume, calculating the final amount of light at each point by simulating the scattering of light through the cloud along the ray that intersects the volume. While this approach can yield visually stunning results for ground-based viewpoints, limitations in the resolution of the required 3D texture can result in loss of detail when viewed up close. This approach can also be very taxing on the fragment processor, especially if more advanced scattering models involving multiple forward scattering are employed.

Another approach is called splatting, which models clouds as a collection of high-resolution puffs overlaid on top of each other. Since individual puffs may be composed with different textures and rotated, detail is preserved in flight training applications where the camera gets near the clouds. On older graphics cards, the overdraw inherent in this approach limited its performance, but modern cards have the fill rate to handle dense cloud layers extending to the horizon at 100+ frames per second using this technique. It is not without downsides; as each puff is basically a 2D billboard facing the camera, special handling is needed so they do not reveal their existence as view angles change. This approach is well suited to partly cloudy conditions, as it renders clouds individually instead of as one giant volume for an entire cloud layer.

Light moves through clouds in very complicated ways, and a tradeoff must be sought between believable cloud representation and performance. Approaches vary from multiple scattering of light rays within a cloud to simply darkening the cloud as sunlight passes through more of it – which is the simplest technique to ensure that large clouds have dark bottoms as expected. Modeling the correct lighting of clouds is the subject of entire theses (11).

The shapes and distribution of clouds are also important to model properly. Cloud shapes may be drawn from a database of artist-created cloud models, or grown procedurally using cellular automata techniques. The latter approach has the advantage of ensuring no two clouds look the same, and enables effects such as cloud growth and dissipation over time. However, it requires extra computation at initialization time. The size of clouds also should not be constant or randomly chosen for a realistic scene; they actually follow an exponential distribution. We use the Plank model (12) to choose the distribution of our cloud sizes.

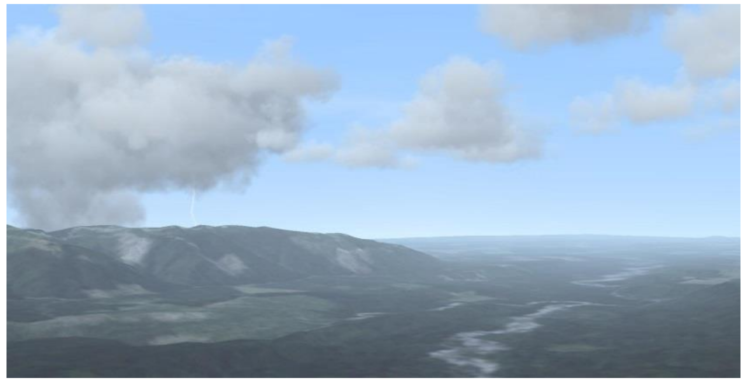

Figure 6: Cumulonimbus and cumulus congestus clouds in SilverLining

Further complicating the problem is the wide variety of cloud types present in nature. While volumetric techniques are well suited for cumulus clouds, other techniques are needed for stratus, cirrus, and similar clouds. Cumulonimbus clouds also require special attention, as they have specific shapes, sizes, and can generate lightning effects. The experience of flying through each of these cloud layer types also needs to be considered.

Today's commercially available 3D cloud SDK's can meet the following requirements:

- Representation of cumulus congestus, cumulus mediocris, cirrus fibratus, nimbostratus, broken stratus, cirrus, stratocumulus, cumulonimbus, and cirrocumulus cloud types to the horizon at any coverage
- 3D volumetric cumulus clouds that may be flown through while maintaining 60Hz performance
- Splatting and GPU ray-casting rendering techniques
- Crepuscular rays ("God rays" or shafts of light resulting from sunlight shining through broken clouds.)
- Lightning and precipitation effects
- Real-time cloud growth and dissipation
- Cloud motion and shapes influenced by simulated wind conditions
- Ability to handle whole-Earth databases using geocentric / ECEF coordinate systems
- Ability to represent localized cloud cover as well as clouds that extend infinitely as the camera moves
- Realistic cloud lighting at all times of day

In addition to our SilverLining Sky, 3D Cloud, and Weather SDK that fulfills these requirements, competing products such as TrueSky by Simul and Ephemeris by Confetti are also worth evaluating.

MORE TO EXPLORE

We invite you explore the works cited, as well as our website at http://www.sundog-soft.com/ to learn more about the broad topic of real-time environmental effects and how Sundog Software is furthering the state of the art. It is a rapidly evolving field that is quickly converging on a combination of cinematic quality, game performance, and simulation fidelity – all available at low cost.

WORKS CITED

1. *Simulating Ocean Water.* **Tessendorf, J.** s.l. : Addison-Wesley, 1999. SIGGRAPH Course Notes.

2. Triton 2.1 Supports JONSWAP, Pierson-Moskowitz Wave Models & Swells. *Sundog Software.* [Online] March 20, 2013. [Cited: February 5, 2014.] http://sundog-soft.com/sds/2013/03/triton-2-1-supports-jonswap-pierson-moskowitz-wave-models-swells/.

3. *Directional Wave Spectra Observed during JONSWAP 1973.* **D.E. Hasselmann, M. Dunckel, J.A. Ewing.** 1980, Journal of Physical Oceanography, Vol. 10.

4. *An Analytic Model for Full Spectral Sky-Dome Radiance.* **L. Hosek, A. Wilkie.** 4, 2012, Proceedings of ACM SIGGRAPH 2012, Vol. 31.

5. *A Practical Analytic Model for Daylight.* **A.J. Preetham, P. Shirley, B. Smits.** July 1999, Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 91-100.

6. *A Physically-Based Night Sky Model.* **H. Jensen, F. Durand, J. Dorsey, M. Stark, P. Shirley, S. Premoze.** 2001. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. pp. 399-408.

7. *Interactive Tone Mapping.* **F. Durand, J. Dorsey.** June 26-28, 2000, Proceedings of the Eurographics Workshop on Rendering Techniques 2000, pp. 219-230.

8. **Kane, F.** Physically-Based Outdoor Scene Lighting. [ed.] E. Lengyel. *Game Engine Gems.* Sudbury : Jones and Bartlett, 2011, Vol. 1, 15, pp. 219-226.

9. **R.E. Bird, C.Riordan.** *Simple Solar Solar Spectral Model for Direct and Diffuse Irradiance on Horizontal and Tilted Planes at the Earth's Surface for Cloudless Atmospheres.* Solar Energy Research Institute. Golden, CO : s.n., 1984. SERI/TR-215-2436.

10. **Kane, F.** Modeling, Lighting, and Rendering Techniques for Volumetric Clouds. [ed.] E. Lengyel. *Game Engine Gems.* Natick : A K Peters, 2011, Vol. 2, 2, pp. 21-44.

11. **Bouthors, Antoine.** *Real-Time Realistic Rendering of Clouds.* s.l. : Grenoble Universites, 2008. PhD dissertation.

12. *The Size Distribution of Cumulus Clouds in Representative Florida Populations.* **Plank, V.** 8, 1969, Journal of Applied Meteorology, pp. 46-67.