

FFT를 이용한 깊은 바다 시뮬레이션

*양은주⁰ **김은주 *유관우

*경북대학교 컴퓨터공학과

**동명정보대학교 정보통신공학과

*ab1713@hanmail.net⁰, **dmswnakdmf@hanmail.net *kwryu@hosanna.net

FFT of Deep Ocean Simulation

*Eun-Ju Yang⁰ **Eun-Ju Kim *Kwan-Woo Ryu

*Dept. of Computer Engineering, Kyungpook National University

**Dept. of Information & Communication Engineering, Tongmyong University of Information Technology

요약

본 논문에서는 해양학의 관찰을 통한 통계학적 모델과 영상처리 기법을 이용하여 바다 표면을 나타내는 방법을 제시한다. 파도를 구성하는 파들의 합을 모두 구하지 않고, 높이필드를 사용하여 FFT로 처리하는 것은 전체적인 파도를 한꺼번에 빠르고 쉽게 모델링 할 수 있다는 장점을 가진다. 또한 통계적 모델을 적용할 때, 영상처리에 사용하는 필터를 사용하여 보다 자연스러운 물결을 표현하는 방법을 제시한다. 그리고 주파수 영역에서 매그니튜드 값의 효율적인 사용법을 제시하여 파도의 애니메이션을 나타낸다.

1. 서론

바다를 사실처럼 나타내는 것은 컴퓨터 그래픽스에서 어렵고 복잡한 작업 중에 하나이다. 실제에 가깝게 모델링한 바다는 게임의 재미와 흥미를 높여주고 영화의 흥미진진한 효과를 극대화한다.

바다의 모델링에 관한 연구는 크게 두 분야로 나뉜다. 한 분야는 파장의 1/20 이하인 수심을 가지는 얕은 바다에 대한 경우로서, 파도는 해저지형과 밀접한 관련을 가지고 있기 때문에 트로코이드(trochoid)를 사용한다 [1-3]. 다른 한 분야는 파장의 반 이상이 수심인 깊은 바다에 대한 것으로 이는 해저지형의 변화에 영향을 받지 않으므로 해양학에서 연구된 자료들을 토대로 한 통계적 분포를 이용한다[4-6].

본 논문에서는 후자에서 설명한 깊은 바다의 모델을 구성하는 방법을 제시한다. 바다 표면은 높이필드(height field)를 이용하여 표현하고, 가우스 확률변수로 높이필드를 초기화한다. 그 후, 이것은 FFT(Fast Fourier Transformation) 과정을 거쳐서 주파수 영역으로 변환되고, JONSWAP(Joint North Sea wave project) 스펙트럼과 Lowpass 필터를 적용한 뒤, IFFT(Inverse FFT)를 사용하여 최종적인 높이필드를 구성한다. 바다의 움직임은 높이필드의 주파수 영역에서 매그니튜드(magnitude)값을 이동(shift)시키고 다시 IFFT 처리하는 과정을 반복하면서 이루어진다.

2. 이전 연구

사실적으로 깊은 바다를 모델링하는 것에 관한 연구는 꾸준히 진행되어 왔다. 초기에는 프랙탈로 바다를 표현하였고, Perlin은 노이즈를 이용하여 먼 거리에서 바라본 바다를 표현했다. 그러나 이러한 방법은 사실적인 표현

에 있어서 미흡한 점이 많았다. Gary A.와 Mastine은 처음으로 푸리에(Fourier)변환 방법과 해양학의 연구들을 토대로 하여 깊은 바다를 표현했다[3]. 이들의 시도는 깊은 바다에 대한 최근의 여러 연구에 중요한 토대가 되고 있다. Simon Premoze은 Gary A.와 Mastine의 연구를 기반으로 하여, 0~255 사이의 랜덤값 대신 파도의 분포가 가우스 확률 분포를 따른다는 사실에 의해 가우스 확률 변수 값을 초기값으로 하여, 좀더 개선된 광학적(optical) 효과를 제시했다[4]. 본 논문에서는 이에 추가적으로 영상처리에서 사용되는 필터를 적용하는 방법과 매그니튜드의 이동에 의한 파도의 애니메이션 방법을 제안하여 좀더 자연스러운 파도의 효과를 표현하였다.

3. 바다의 모델링

3.1. 파도 모델

중력파중 하나인 풍파(wind wave)는 바람 에너지가 바다에 전달되어 생성되며, 이것의 성장은 풍속, 바람의 지속시간 그리고 풍역대(바람의 방향이 변하지 않고 중단없이 부는 영역)에 의해 이루어진다. 이를 세 요소가 충분히 오랫동안 지속되면 바다는 완전히 다 발달하게 되고, 이를 다 자란 바다(fully developed wind

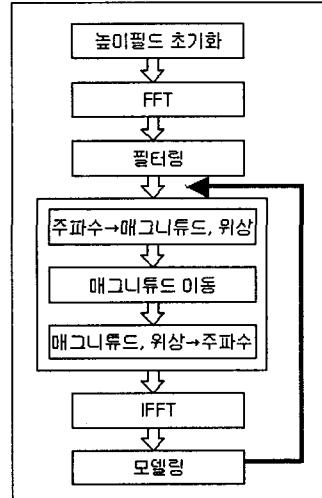


그림 1. 모델링 과정

sea)라 한다. Pierson과 Moskowitz는 5년 동안 다 자란 바다에서 파도의 변화를 관찰하여 1000개 이상의 데이터를 수집하였으며, 이를 기반으로 스펙트럼 모델을 제시했다.

$$F_{PM}(f) = \frac{ag^2}{(2\pi^4)f^5} e^{-\frac{4}{5}\left(\frac{f_m}{f}\right)^4} \quad (1)$$

식 (1)에서 f 는 주파수, a 는 Phillips 상수, g 는 중력가속도 그리고 f_m 은 최고 주파수(peak frequency)로 바람의 속도와 직접적인 관계가 있다.

$$f_m = \frac{0.13g}{U_{10}} \quad (2)$$

식 (2)의 U_{10} 은 바다 표면의 10m 위에서 측정된 바람의 속도를 나타낸다.

그러나 이 모델은 바람의 속도만을 다루는 한계가 있으므로, 본 논문은 이러한 점을 수정하여 파도의 진행방향까지 고려한 JONSWAP 스펙트럼을 사용한다.

$$F_J(f, \theta) = F_{PM}D(f, \theta) \quad (3)$$

$D(f, \theta)$ 는 바람의 방향 θ 에 대해 파도가 진행하는 방향을 나타낸다. 이것은 다음의 관계에 의해,

$$D(f, \theta) = \frac{1}{N_p} \cos^{2p}\left(\frac{\theta}{2}\right) \quad (4)$$

로 나타낸다. 식 (4)에서

$$p = 9.77 \left(\frac{f}{f_m}\right)^\mu \quad (5)$$

$$\mu = \begin{cases} 4.06 & \text{if } f < f_m \\ -2.34 & \text{otherwise} \end{cases} \quad (6)$$

이다. 표준(normalization) 상수는,

$$N_p = \frac{2^{1-2p} \pi \Gamma(2p+1)}{\Gamma^2(p+1)} \quad (7)$$

으로 정의하고, Γ 는 감마함수이다. 이 모델은 바람 방향에 대한 효과뿐만 아니라, 최고 주파수에서 좀더 뾰족한 윤곽을 표현한다.

통계적 모델은 타당한 수학적 접근을 위해 높이필드를 사인과 코사인의 합으로 분해하며, 이를 위하여 본 논문에서는 이를 빠르게 구하는 방법인 FFT를 사용한다.

본 논문에서 제안된 모델링 방법은 크게 2단계에 의해서 진행된다. 첫째는 물 표면의 높이를 결정짓는 높이필드의 처리이다. 그림 1의 높이필드 초기화, FFT 그리고 필터링이 이 단계에 포함된다. 높이필드가 결정되면 이를 이용하여 애니메이션을 하는 것이 두 번째 단계로,

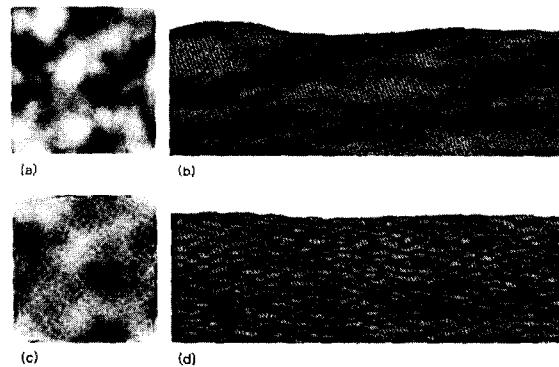


그림 2. 필터링 처리 전(a, b)과 필터링 처리 후(c, d)

그림 1에서 보는 것처럼 매그니튜드(magnitude)값의 변화, IFFT 그리고 모델링 과정을 반복하였다.

3.2. 높이필드의 처리

해저지형은 평평하다고 가정하고, 초기의 바다 표면은 x-z평면에 어떠한 물결도 일지 않는 물 표면처럼 ($y=0$) 평면 형태의 격자 그리드를 구성한다. 그리드의 교차점 (x_i, z_i) 의 y_i 값은 해저지형과 물의 표면사이의 거리로 나타내고, 높이필드 (h_x, h_z) 좌표에 주어진 각각의 값은 대응되는 y_i 값으로 매핑시켜 바다 표면을 표현한다.

다 자란 바다에서 실험 측정 결과, 물 표면의 위상과 기울기 등은 가우스 분포(Gaussian distribution)를 가진다는 사실을 확인하였다. 따라서 본 논문에서 높이필드는 가우스 변수 생성기(Gaussian random number generator)를 사용하여 초기화해 주었다. 그림 2의 (a)는 128×128 의 높이필드를 영상화해서 표현한 것으로 흰색은 강도(intensity)가 255인 경우이며 검은색은 강도의 값이 0인 경우이다. (b)는 높이필드의 강도 값을 3D 그리드의 y 값으로 매핑하여 모델링한 결과이다. (b)의 결과만으로 파들의 전체적인 윤곽이 그려지긴 하지만, 실제 파도라고 하기에는 부족하다. 그러므로 파도를 좀더 자연스럽게 표현하기 위해 해양학에서 연구한 통계적 분포를 적용한다.

본 논문에서 바다의 모델링을 위한 높이필드는 적절한 필터링을 거쳐서 계산된다. 그것을 위해 우선 FFT처리를 하여 높이필드를 주파수 영역, 즉, 실수와 허수로 표현하고, JONSWAP 스펙트럼으로 필터링을 한다. 그러면 $\cos^2(\theta)$ 에 의해 바람의 방향을 반영하게 되고, 이때 최고 주파수(f_m)의 근처에서 바다의 그럴듯한 외형을 얻는다. 필터링 결과가 좀더 사실에 가까운 파도의 효과를 위해서 영상처리에서 사용되는 lowpass 필터를 적용하여 이웃한 높이간에 갑작스런 차이를 줄였다. 그 후, IFFT를 하여 다시 공간 영역(spacial domain)의 높이필드로 변환한다. 그림 1의 (c)는 필터링 처리가 끝난 높이필드를 영상화하여 표현한 것이고 (d)는 3D상 그리드로 모델링 한 것이다.

3.3. 바다의 애니메이션

파도 모델이 구성된 후, 그것의 높이를 일정 시간 간격마다 변화를 주어, 넘실거리는 파도를 표현한다. 이를 위해 우선 주파수 영역의 매그니튜드와 위상을 처리한다.

$$\text{Magnitude} = \sqrt{\text{Real}^2 + \text{Imaginary}^2} \quad (8)$$

$$\text{Phase} = \text{Arctan}(\text{Imaginary}/\text{Real}) \quad (9)$$

높이필드는 그림 1에서와 같이 시뮬레이션의 초기에 단 한번 FFT와 필터링 처리를 한다. 그 후 주파수 영역에서 매그니튜드의 값을 이동(shift)시키고, 변경된 매그니튜드와 본래의 위상을 다시 실수와 허수로 복원을 하여 처리를 시간 간격마다 반복하도록 한다. 복원은 아래와 같은 방법으로 처리한다.

$$\text{Real} = \sqrt{\frac{\text{Magnitude}^2}{1 + \tan^2(\text{Phase})}} \quad (10)$$

$$\text{Imaginary} = \sqrt{\text{Magnitude}^2 - \text{Real}^2} \quad (11)$$

최종적으로 IFFT를 하여 바뀐 값으로 높이필드를 수정한다.

그러나 이렇게 복원하는 과정에서 발생하는 오차가 누적되면 높이필드의 결과에 치명적인 오류를 초래하기 때문에, 초기 필터링 과정을 거치고 난 후, 주파수 영역의 값을 따로 저장해둔다. 반복에 의해 수정된 주파수 값을 계속 사용하지 않고, 매 반복마다 이동시킬 매그니튜드 값의 양을 누적한다. 그래서 초기에 저장해둔 주파수 값에 누적된 매그니튜드값 만큼 이동시켜서 오차의 누적으로 인한 심각한 오류를 피하도록 하고, 이러한 사실은 실험을 통해서 확인했다. 또한 저장해둔 초기의 주파수 값은 실수와 허수의 복원식에 의해서 모두 양의 값을 가지게 되어 주파수 영역에서 높이필드의 결과에 중대한 영향을 미치기 때문에, 저장된 초기 주파수의 값을 참고하여 그들의 부호와 복원된 부호를 같도록 처리해 준다.

4. 실험 및 결론

앞에서 제시한 높이필드 처리와 애니메이션 방법을 사용하여 Pentium III, 256MB 메모리 그리고 64MB의 그래픽카드 상에서 OpenGL로 구현하여 실험하였다. 높이필드는 128×128 크기를 사용하였으며, x-z 평면에서 크기는 576×576 이다. 다른 바다로 하더라도 바람의 속도에 따라 파도의 모양이 크게 바뀌므로, $15 \sim 20 \text{m/s}$ 바람 속도가 적당하며 본 논문에서 속도는 15m/s , θ 는 45° 로 두었다. 그림 3은 파도의 움직임을 애니메이션하는 과정을 캡쳐한 것이다.

여기서 제시한 모델링 방법은 이미 많이 사용하고 있는 영상처리 기법을 응용한 것이기 때문에 쉽고 빠르게

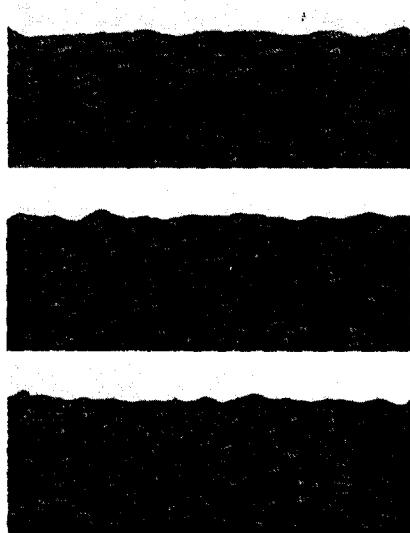


그림 3. 파도의 움직임

모델링 할 수 있을 뿐만 아니라 다른 연구들에 비해서 랜더링하는 것 또한 비교적 간단하다. 모델링 과정에서 이미 파도의 잔물결이 표현되기 때문에 랜더링에서 범프맵(bump map)을 사용하는 수고를 덜 수 있다. 또, 이때 기존의 JONSWAP 스펙트럼의 요소를 적절히 수정하면 또 다른 바다의 파도현상을 모델링 할 수 있을 것이다.

그리고, 애니메이션을 할 때, 본 논문에서 제시한 매그니튜드의 이동대신, 실제 파도의 움직임에서처럼 위상을 이동시키는 방법을 고려한다면, 좀 더 사실적인 움직임을 표현할 수 있을 것이다.

5. 참고문헌

- [1] Alain Fourier and Willian T. Reeves, A Simple Model of Ocean Waves, Computer Graphics (SIGGRAPH '86 proceedings), volume 20, pp.75-84, 1986.
- [2] Darwyn R. Reacheck, Modeling Waves and Surf, SIGGRAPH '86, volume 20, number 4, pp.18-22, 1986.
- [3] Sébastien Thon, Jean-Michel Dischler and Djamel Ghazanfarpoor, Ocean waves synthesis using a spectrum-based turbulence function, IEEE Computer Graphics and Applications, 2000.
- [4] Gary A. Mastine, Peter A., Wetterberg and John F. Mareda, Fourier Synthesis of Ocean Scenes, IEEE Computer Graphics and Applications, pp.16-22, 1987.
- [5] Simon Premoze and Michael Ashikhmin, Rendering Natural Waters, Computer Graphics Forum, volume 20, number 4, pp.189-199, 2001.
- [6] Damien Hinsinger, Fabrice Neyret and Marie-Paule Cani, Interactive Animation of Ocean Waves, SIGGRAPH 2002, pp. 161-166, 2002.