

선형 컨벌루션 기반의 물표면과 객체의 실시간 상호작용 애니메이션

Linear Convolution Based Realtime Animation of Interaction between Water Surface and 3D object

강경헌, 허기택, 김은석
동신대학교 디지털콘텐츠학과

Gyeong heon Gang, Gi Taek Hur, Eun Seok Kim
Dept. of Digital Contents, Dongshin University

요약

CG에서 물을 애니메이션하거나 다양한 특수효과를 표현하기 위해 유체역학의 기술들이 사용되고 있다. H/W 성능이 높아지면서 이전에 불가능했던 알고리즘들이 실시간으로 가능해지고 있기는 하지만, 정밀한 표현에는 여전히 많은 시간이 소요되며, 성능과 사실성 사이의 균형을 위한 다양한 기술들이 연구되고 있다. 특히 게임과 같은 문맥을 가지는 곳에서 사용자의 요구에 의해 바다나 호수같은 넓은 지역의 물표면과 객체의 상호작용을 표현하기 위해서는, 물리적 사실성을 어느 정도 희생하더라도 시각적인 사실성을 유지하는 범위에서 실행성능을 높이는 것이 우선시 된다. 본 논문에서는 물표면과 객체의 상호작용에 의한 다양한 물표면의 형태변화를 선형 컨벌루션 기법과 경계구를 이용하여 실시간으로 자연스럽게 애니메이션하는 방법을 제안한다.

Abstract

In computer graphics, fluid dynamics are used for animating and expressing the various special effects of water. Although the algorithms that were impossible through the hardware at that time become to be possible in real time, it still requires a lot of time to achieve the realistic and detailed expressions. Therefore, there are many researches on the techniques of balancing between performance and quality. Specially in environments with the game context which express the interaction between water surface and 3D objects in the wide area such as sea or lake according to user's request, it must give priority to the executive performance preserving the visual reality even though physical reality is sacrificed. In this paper, we propose a method for the realtime animation of interaction between water surface and 3D objects using the linear convolution of height fields and the bounding spheres of objects.

I. 서론

H/W의 발전과 함께 CG에서 유체를 애니메이션 하기위해 이전에 불가능했던 알고리즘들의 시뮬레이션이 가능해지고 있으며, 물, 불, 폭발, 연기, 안개 등의 유체 표현 기술이 성공적으로 적용되어지고 있다. 물은 물리적 처리에 있어 타고난 복잡함을 포함하고 있기 때문에 물리적 법칙을 기반으로 다양한 효과를 실시간 시뮬레이션 하기 매우 어렵다. 결국 게임과 같은 문맥을 가지는 환경에서 물과 상호작용하는 이펙트를 구현할 때는 품질과 성능의 교환이 불가피하며, 호수나 바다와 같은 넓은 지역의 움직임을 실시간으로 시뮬레이션 하기 위해서는 물이 가지는 많은 물리적 속성들을 실시간이 가능할 정도로 배제시키며, 시각적으로 사실성을 유지하면서 근사치를 계산하여 표현하여야 한다. 물과 객체와의 상호작용은 물리적으로 정확한 규칙을 사용한 비실시간 유체 시뮬레이션에 주로

이루어져 있으며, 정적인 대규모 액체와 객체와의 실시간 시뮬레이션에 대한 연구는 많이 이루어지지 않고 있다[1].

본 논문에서는 물표면과 객체의 상호작용을 실시간 시뮬레이션하기 위해 선형컨벌루션과 객체의 경계구를 이용한 높이 필드 변형 방법을 제안하고자 한다.

II. 관련연구

유체와 객체와의 상호작용 애니메이션을 표현하기 위한 다양한 방법들이 연구되어 왔다. 많은 연구들이 유체와 객체의 충돌 및 주변의 힘에 따른 사실적인 파동이나 물방울 생성을 위해 주로 파티클 및 볼륨 모델을 이용하였다.

96년 Foster와 Metaxas는 유체 전체 환경을 사각 그리드로 나누어 유체 영역과 장애물이 포함된 영역을 구분하여 유한차분을 이용한 Navier-Stokes방정식의 해를 구함으로써 상호작용을 고려하였다[2]. 99년 Stam은 유체를 파티클로 모델링

하고 파티클의 움직임을 역추적하여 이동 속도를 계산하고, 객체를 속도 집합의 경계조건(boundary condition)으로 다룸으로써 상호작용을 구현하였다[3]. 2001년 Foster와 Fedkiw는 파티클과 Level Set을 이용하여 물을 표현하고, 객체의 탄젠트를 따라 유체가 자유롭게 움직이도록 함으로써 상호작용을 표현하였다[4].

그러나, 유체를 파티클 또는 격자로 표현되는 볼륨으로 표현하는 경우 계산상의 비용이 크기 때문에 실시간 시물레이션에 적용하기 어렵다는 단점이 있다. 따라서 실시간 물표면 모델은 sin파와 같은 정현파의 합성을 통한 물 표면만을 표현하는 wave모델을 사용하고 있다.

이러한 wave Model로는 파동 형태 표현을 위해 Pelin Noise[5]와 같은 절차적인 noise를 사용하거나, Gerstner Model[6]을 이용하여 모델링하고 물표면에 파동이론을 적용하여 시물레이션한다. 물표면 메쉬와 객체와의 상호작용에 관련된 연구로는 2002년 Lvoiscah[7]의 컨벌루션을 기반으로 하여 배가 지나갈 때의 물표면 곡형 변형 방법이 있다.

III. 선형 컨벌루션 기반의 물표면 애니메이션

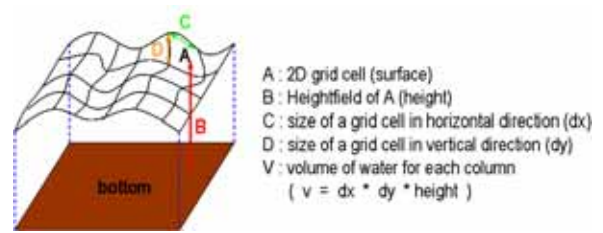
바다나 호수 등 대규모의 정적인 유체의 가장 역동적인 상황은 물체와의 상호작용에 의해 물 표면에 발생하는 물결 및 파도의 변화일 것이며, 유체 시물레이션에서 가장 관심을 일으키는 부분이기도 하다. sin파와 같은 주기 함수들의 계산에 의한 파동 모델은 연속함수이기 때문에, 물 표면과 객체의 상호작용에 의해 지역적으로 급격한 변형을 생성하기 어렵다는 문제점을 갖는다. 따라서, 본 논문에서는 sin파와 같은 주기 함수들의 계산에 의한 파동생성이 아닌, 물표면을 모델링하는 메쉬 정점들의 높이 값을 저장한 높이필드(Hieght Field)에 선형 컨벌루션을 적용시킴으로써 물 표면의 움직임을 생성한다.

1. 높이 필드(Height Field)

높이 필드는 연산량의 감소를 위해 완전한 3D가 아닌 2.5D로써 물표면을 표현한 모델로서, 빠른 2D 계산과 적은 리소스의 사용, 그리고 쉽게 렌더링할 수 있다는 등의 이점을 제공한다. [그림 1]은 높이 필드의 구성요소를 보여주며, 각 정점의 높이를 변경함으로써 효과적으로 물표면의 애니메이션을 생성할 수 있다[8].

물결의 움직임은 시간에 따라 일정 방향으로 이동함으로써 나타난다. 즉, 현재 물표면 메쉬의 높이 필드는 이전 시간의 주변의 물결 높이에 영향을 받으며, 자연스러운 물표면 움직임 생성을 위해서는 시간 t-1과 t에서의 높이 필드가 모두 필요하다. 따라서, 본 논문의 물표면 애니메이션은 연속된 시간에 대

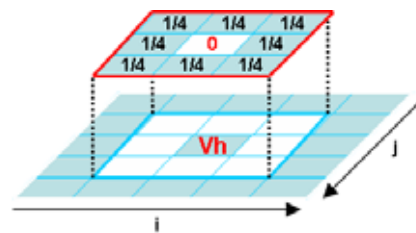
한 두 개의 높이 필드를 사용한다.



▶▶ 그림 1. 높이 필드구성

2. 선형 컨벌루션(Linear Convolution)

컨벌루션은 일반적으로 영상처리에서 선명화, 경계선 검출 등의 영역 기반 처리를 하기위해 사용되는 기법으로, 주어진 위치의 값을 구하기 위해 이웃 값들의 가중치 평균을 이용하는 방법이다[9]. 본 논문에서는 연속적인 물결의 움직임(파동)을 생성하기 위해 [그림 2]와 같은 선형 컨벌루션을 이용하였다.



▶▶ 그림 2. 파동 생성을 위한 선형 컨벌루션 영역 필터

선형 컨벌루션을 이용하여 물을 애니메이션 하기위해서는 파동의 지속여부를 조절하기 위한 댐핑(Damping) 인자가 필요하다. 댐핑인자는 0에서 1사이의 실수값을 가지며 생성된 파동이 얼마나 빨리 소멸되는지를 결정한다[10]. 댐핑인자가 0인 경우 파동은 다음시간에 즉시 사라지게 되고, 1인 경우 물의 파동은 절대로 사라지지 않고 계속 유지된다.

선형 컨벌루션을 이용한 파동 애니메이션 생성하기 위한 방법은 다음과 같다. 먼저, 현재시간 t의 높이필드에 영역필터 마스크를 적용하여 임시변수에 저장하고(식1), 시간 t-1의 높이 값을 차감한 후 댐핑인자 Dp를 곱해주어 시간 t+1의 높이값을 저장한다(식2). 모든 정점에 대하여 선형 컨벌루션 과정이 끝나면 높이필드 값을 업데이트한다.

$$Buff_{(x,z)} = \sum_{i=1}^{-1} \sum_{j=1}^{-1} (Vh_{(x+i,z+j)}(t) \times C(i,j)) \quad (1)$$

$$Vh_{(x,z)}(t+1) = (Buff_{(x,z)} - Vh_{(x,z)}(t-1)) \times Dp \quad (2)$$

C(i,j)는 파동 애니메이션을 생성하기 위한 선형 컨벌루션의 영역필터로서 다음과 같은 값을 갖는다.

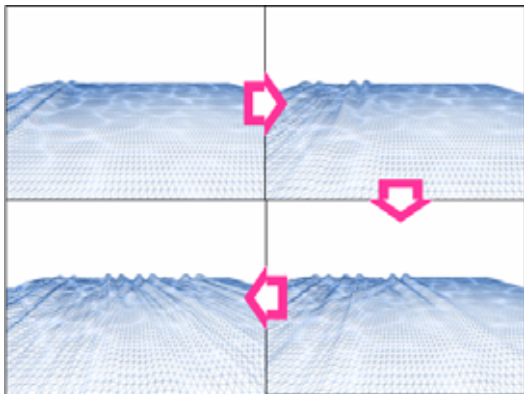
$$C(i,j) = \begin{cases} 0, & \text{if } i=j=0 \\ 1/4, & \text{otherwise} \end{cases}, \quad i,j \in -1,0,1$$

[식 2]를 이용한 높이필드의 업데이트 알고리즘은 다음과 같다. 이 때 xmin와 zmin는 물표면을 나타내는 메쉬의 정점들의 x,z의 최소값을, xmax와 zmax는 최대값을 나타낸다.

```

for(x=xmin; x<xmax; x++)
  for(y=ymin; y<ymax; y++){
    식1과 2를 이용하여  $Vh_{(x,z)}(t+1)$ 를 계산
     $Vh_{(x,z)}(t-1) = Vh_{(x,z)}(t)$ 
     $Vh_{(x,z)}(t) = Vh_{(x,z)}(t+1)$ 
  }
    
```

[그림 3]은 좌측에서 발생한 물결이 시간의 변화에 따라 선형 컨벌루션에 의해 우측으로 이동하는 모습을 보여준다. 물표면의 파동 애니메이션은 선형 컨벌루션 과정을 통해 이뤄지며, 물표면에서 발생하는 이펙트는 높이필드를 재정의 하는 방법에 따라 다양하게 생성할 수 있다.



▶▶ 그림 3. 컨벌루션을 이용한 파동 애니메이션

IV. 물표면과 객체의 상호작용

물의 파동은 호이겐스(Huygens)의 원리에 따라 파원으로 부터 발생하는 구면파 형태의 2차파형이므로, 높이값을 구의 형태로 변형시켜주면 파동의 움직임을 애니메이션 할 수 있다. 본 논문에서는 객체와 물표면과의 상호작용을 위해 객체의 경계구(bounding sphere)를 정의하고 경계구의 크기와 위치에 따라 물표면의 높이필드값 $Vh_{(x,z)}(t)$ 를 변형시킴으로써 상호작용을 표현한다. [그림 4]는 객체의 경계구가 물표면을 향해 움직일 때 물표면과의 충돌에 의해 물표면이 변형된 모습을 보여준다.



▶▶ 그림 4. 경계구에 의한 물표면 변형

1. 가속 인자(Acceleration Factor)

동일한 크기를 가진 객체가 물표면에 충돌할 때, 물표면은 경계구가 가진 무게와 속도에 따라 변형이 달라질 것이다. 경계구의 크기와 위치는 현재 시간 t의 물표면의 높이필드값에 영향을 준다. 그러나 파동 애니메이션은 선형 컨벌루션을 통해 t에서의 높이값과 t-1에서의 높이값을 조합하여 t+1의 높이값을 계산하므로, t-1에서의 높이필드값 역시 조정이 필요하다. 또한 t-1에서의 높이값은 객체의 이동 특성을 반영하여야 한다. 본 논문에서는 이러한 특성을 고려하여 객체의 무게와 속도에 비례하는 가속인자를 정의하고 높이필드에 적용하였다.

가속인자 AF는 t-1의 높이필드값을 변형시켜 대체되어 애니메이션시 객체의 움직임에 따른 물표면의 변화를 유도할 수 있게 하며, 경계구가 물의 현재 높이필드 $Vh_{(x,z)}(t)$ 를 변형시킬 때, 이전 시간의 높이필드 $Vh_{(x,z)}(t-1)$ 을 변형시키는 식은 다음과 같다.

$$Vh_{(x,z)}(t-1) = Vh_{(x,z)}(t) - Vh_{(x,z)}(t) \times AF \quad (3)$$

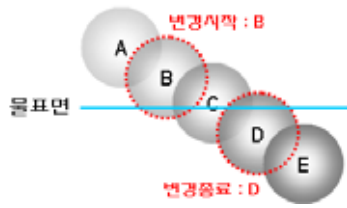


▶▶ 그림 5. AF값에 의한 t+1에서의 물표면 변화

[그림 5]는 경계구가 물 표면에 충돌하였을때, AF값에 따른 물표면의 변화를 보여준다. AF는 실수값을 가지며 0을 기준으로 양의 값을 가지면 물의 파동 변화가 커지고, 음수 값을 가지면 파동 변화의 감쇄가 빨라져 원래의 높이로 빨리 복원된다. AF는 파동에 대한 반발력을 정의하는 것이고, Dp는 파동의 수명을 정의하는 것으로서, AF값에 의한 감쇄는 Dp와는 다르게 나타난다.

2. 움직이는 객체에 의한 물표면의 변형

[그림 5]은 경계구가 수직방향으로 떨어졌을 때를 기준으로 시뮬레이션한 결과로서, 움직임이 수직이 아닌 방향성을 가진 객체의 경우 이를 고려하여 높이필드를 변경하여야 한다. 높이필드를 변경하는 시점은 [그림 6]과 같이 경계구가 물표면에 닿는 순간부터 완전히 잠길 때까지로 정의할 수 있다.



▶▶ 그림 6. 객체에 의한 물표면의 높이필드 변경

경계구의 위치는 시간에 따라 변화하게 되며, [그림 6]처럼 수면과 충돌 시작시 높이필드를 변형하기 시작한다. 즉, 변경 시작 시점 B에서의 물표면 높이를 기준으로 경계구가 완전히 잠기게 되는 시점인 D 이후에 높이 필드 변형을 종료하여야 한다. 지금까지의 방법들을 이용하여 물표면과 객체의 상호작용을 고려한 물표면 변형 알고리즘은 [표 1]과 같다.

[표 1] 물표면 변형 알고리즘

```

While(Simulation)
{
    Step1. if(특정경계구가 최초로 수면에 닿았는지 체크)
        fixHeight[][] = 수면의 경계 저장
    for each class of bounding sphere
    {
        Step2. if(경계구가 fixHeight[][] 아래로 잠기면)
            Goto Step7.
        Step3. if(fixHeight[][]와 경계구 표면이 충돌하면){
            Step4. 경계구 형태에 따라  $Vh_{(x,z)}(t)$  변경
            Step5. 정의된 무게,속도비례에 따라 AF계산
            Step6.  $Vh_{(x,z)}(t-1)$  변경
        }
        else Goto Step7.
        Step7. 경계구의 다음 위치 계산과 이동
    }
    Step8. Height Field에 선형 컨벌루션 적용
}
    
```

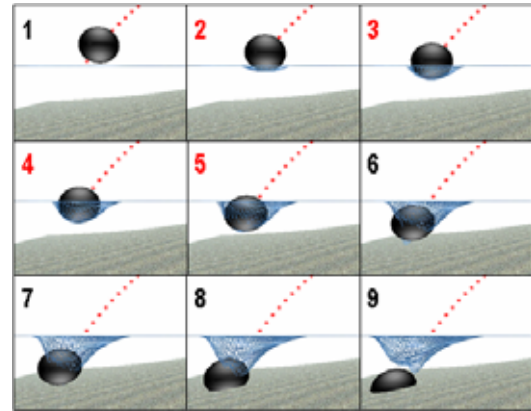
V. 실험 결과

본 논문에서는 제안한 방법들을 이용하여 포탄 발사와 물고기 점핑을 시뮬레이션 하였다.

1. 포탄 발사 시뮬레이션

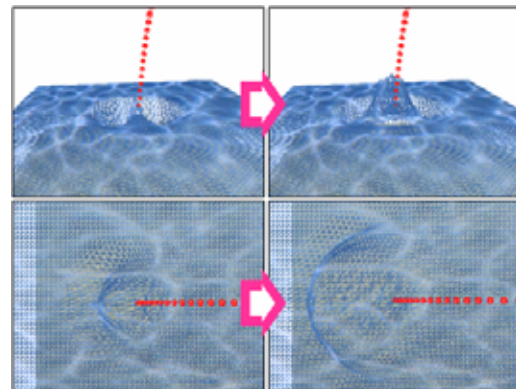
사용자는 발사되는 지점을 향해 조향각과 발사되는 힘, 포탄의 크기를 지정하여 발포하게 되며, 가속인자 AF값은 0으로 고정하고 댐핑인자 Dp는 0.95로 설정하였다. [그림 7]은 포탄이 발포되어 수면에 도달하였을 때의 물표면을 변형시킨 결과로서, 2,3,4,5번에서 물표면과 포탄과의 상호작용에 의해 높이

필드가 변형된 후 선형 컨벌루션이 진행되면서 생성한 물표면 변화를 보여준다.



▶▶ 그림 7. 포탄발사 시뮬레이션 결과

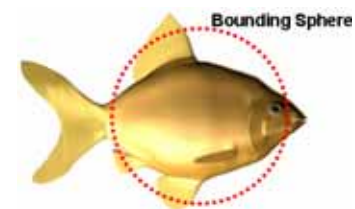
[그림 8]은 [그림7]의 시뮬레이션을 다른 각도에서 바라본 것으로서, 포탄이 물에 떨어지면서 물웅덩이가 생겼다가 물기둥으로 솟아오르는 모습과, 포탄의 진행 방향으로 이동하는 파도의 형태까지 자연스럽게 표현되었다.



▶▶ 그림 8. 포탄발사 시뮬레이션 결과

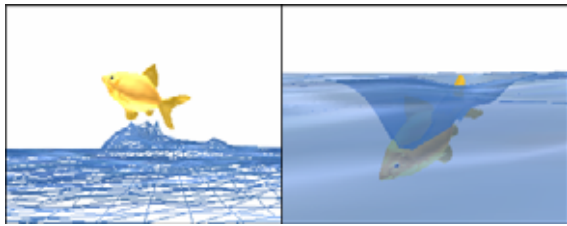
2. 물고기 점핑 시뮬레이션

기본 구형으로 표현되는 포탄보다 좀 더 복잡한 형태의 객체와 물표면의 상호작용 시뮬레이션을 위해 물고기 객체를 이용하여 수면에서 점핑하는 시뮬레이션을 수행하였다. [그림9]는 시뮬레이션에 사용된 물고기 객체와 물고기 객체의 경계구 설정을 보여주고 있다.



▶▶ 그림 9. 물고기 객체의 경계구 설정

물고기 객체는 수면 아래에서 점핑하여 수면 위로 올라간 뒤 다시 수면 아래로 들어가는 점핑 동작을 수행하는데, 수면으로 올라갈 때는 경계구의 위쪽부분을 높이필드 $Vh_{(x,z)}(t)$ 에 적용하였고, 내려갈 때는 아래쪽부분을 적용함으로써 물표면과의 상호작용을 구현하였다. 물고기는 포탄에 비해 가벼워 물과동의 변형이 빨리 감소하도록 가속인자 AF는 -0.3, 물과동은 더욱 빨리 소멸되도록 댐핑인자 Dp는 0.90으로 설정하였고 시뮬레이션의 결과는 [그림 10]과 같다.



▶▶ 그림 10. 물고기 점핑 시뮬레이션 결과

[그림 9]의 물고기 객체를 모두 포함하는 경계구를 설정할 경우 실제 물고기 객체가 아닌 영역을 많이 포함하여 물표면과의 상호작용이 어색하게 일어날 수 있기 때문에 물고기의 일부만을 포함하는 경계구를 설정하였다. 객체의 경계구를 객체 형태에 적합한 사이즈가 다른 여러 개의 경계구 집합으로 설정하여 물표면 변형을 수행한다면 객체의 영향에 따라 물표면을 좀 더 세밀하게 표현할 수 있을 것이다.

VI. 결론 및 향후 연구

본 논문에서는 움직이는 객체를 하나의 경계구로 정의하고, 물 표면에 객체가 충돌하였을 때 물과동 변형의 감쇄 및 증대를 고려한 가속인자(AF)를 추가함으로써 객체와 물표면의 상호작용에 의한 자연스러운 물표면 변형 방법을 제안하였다. 제안된 방법은 먼저 발생한 파가 뒤이어 발생한 파와 충돌하여 파형을 상쇄시키면서 진행 방향으로 파동이 이동하는 것처럼 보이게 되는 호이젠스의 원리에 따라 선형 컨벌루션을 이용하여 물표면의 파동 애니메이션을 생성하였으며, 객체의 경계구를 이용한 높이필드의 변형을 통해 객체와의 상호작용에 의한 수면 밑이나 수면위의 모습도 자연스럽게 표현할 수 있도록 해준다.

향후 연구로는 객체의 크기와, 무게, 속도, 및 재질 특성에 따른 가속인자 설정 방법을 연구하고, 세밀하고 사실적인 물표면 애니메이션을 생성하기 위한 복잡한 형태를 갖는 객체에 대응하는 경계구 생성을 위한 공간분할 기법과 다양한 주변환경에 따른 상호작용 인자를 추가함으로써 사실적인 실시간 물표면 애니메이션을 생성하고자 한다.

■ 참고 문헌 ■

- [1] 표순형, 구분기, "CG유체 표현기술 동향", 전자통신동향분석 제 20권 제4호, 2005
- [2] FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical Models and Image Processing* 58, 5, 471-483.
- [3] STAM, J. 1999. Stable fluids. In *Proceedings of SIGGRAPH99, Computer Graphics Proceedings, Annual Conference Series*, 121 - 128.
- [4] FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquid. In *Proceedings of ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series*, 23-30.
- [5] Ken Perling, Improved Noise reference implementation SIGGRAPH 2002 paper,
- [6] A. Fournier and W. Reeves, A Simple Model of Ocean Waves, in *Proc. of SIGGRAPH '86, Aug. 1986*, pp.75-84
- [7] Loviscach, J. A Convolution-Based Algorithm for Animated Water Waves. *Eurographics 2002 Short Pa-per Presentations*, pp. 381-389, 2002.
- [8] <http://gamedev.net/reference/articles/article2138.asp>
- [9] <http://gamedev.net/reference/articles/article915.asp>
- [10] Mason McCuskey, SPECIAL EFFECTS GAME Programming With DirectX, Premier Publising, 2001