

물수제비 시뮬레이션을 위한 개선된 동역학 모델

이남경[†], 백낙훈^{†*}

요 약

우리는 일상에서 유체와 강체 사이에서 일어나는 상호작용을 흔히 볼 수 있다. 하지만 이를 시뮬레이션 하는 것은 많은 계산량이 필요한 어려운 작업이다. 본 논문에서는 유체와 강체 사이의 상호작용 현상 중 하나인 물수제비 현상을 실시간으로 시뮬레이션 할 수 있는 역학적 모델을 제안한다. 이를 위해 실시간에 계산 가능하면서도 이전 연구에서 고려하지 않았던 돌맹이의 회전운동을 포함하는 개선된 역학적 모델을 사용하며 공기와의 마찰로 생기는 힘들도 포함한 수식을 제안한다. 제안하는 모델을 사용하면 사용자의 다양한 입력에 대해 사실적인 물수제비 현상을 시뮬레이션 할 수 있다. 또한 이전 결과에 비해 보다 원에 가까운 파장을 만들면서 실시간 처리가 가능한 수면 모델도 제시한다. 본 논문에서 제안하는 방법은 상호작용 역학 시스템이나 게임 엔진들에 쉽게 적용할 수 있다.

An Improved Dynamics Model for Stone Skipping Simulation

Nam Kyung Lee[†], Nakhoon Baek^{†*}

ABSTRACT

We can see interactions between rigid body and fluid every day, anywhere. This kind of rigid body-fluid simulation is one of the most difficult problems in physically-based modeling, mainly due to heavy computations. In this paper, we present a real-time dynamics model for simulating stone skipping, which is a popular rigid body-fluid interaction in the real world. In comparison to the previous works, our improved dynamics model supports the rotation of the stones and also computes frictional forces with respect to the air. We can simulate a realistic result for various user input by using proposed model. Additionally, we present a water surface model to show more realistic ripples interactively. Our methods can be easily adapted to other interactive dynamics systems including 3D game engines.

Key words : stone skipping(물수제비), real-time simulation(실시간 시뮬레이션), dynamics model(동역학 모델)

1. 서 론

유체와 강체 사이에서 생기는 상호작용 중 하나인 물수제비는 고대 그리스 이래 지금까지 이어져오고 있는 놀이이다. Bocquet은 물수제비 현상을 수면과 돌 사이에서 발행하는 항력 때문이라고 설명했고, 가

장 많은 물수제비 횡수를 얻기 위한 방법을 물리학적으로 설명하였다[1]. 하지만 물수제비 현상과 같이 강체와 유체간의 상호작용에 대한 시뮬레이션을 컴퓨터 상에서 구현하는 것은 쉽지 않다. 유체 현상은 정해진 형태가 없이 다양하게 변화하고, 이를 수치적으로 해석한 나비에 스톡스 방정식 역시 비선형형으로

* 교신저자(Corresponding Author) : 백낙훈, 주소 : 대구시 북구 산격동 경북대학교 컴퓨터학부 (702-701) 전화 : 053-950-6379, FAX : 053-950-6369, E-mail : oceancru@gmail.com

접수일 : 2010년 3월 12일, 수정일 : 2010년 5월 25일
완료일 : 2010년 6월 8일

[†] 정회원, 경북대학교 IT대학 BK21 Post Doc.
(E-mail : namklee@hotmail.com)

^{†*} 정회원, 경북대학교 IT대학 컴퓨터학부 교수

※ 이 논문은 2009년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(과제번호: 2009-0088544).

로 인해 이산화하기가 힘들다. 이런 문제들로 인해 유체역학에서의 연구결과들을 컴퓨터 그래픽스에 적용하기 위하여 많은 노력들이 있었지만, 아직까지 모든 경우에 적용가능한 일반적인 접근법은 없다.

본 논문은 물수제비의 움직임을 실시간으로 시뮬레이션하는 기법을 제안한다. 이전에도 물리기반 모델링을 바탕으로 한 강체와 유체간의 상호작용에 대한 시뮬레이션이 연구되었다. 하지만 앞선 연구들은 주로 강체의 충돌로 생기는 물표면의 사실적 표현에 초점이 맞추어져 있었다[2-4]. 물수제비 현상에 대한 이전 연구는 항력을 수직항력과 수평항력으로 분해해서 적용함으로써 수식이 복잡하고, 물수제비 현상의 주된 요인 중의 하나인 회전을 고려하지 않았다[5]. 회전은 회전체의 균형을 유지하는 필수 요소로, 회전 유무에 따라 물수제비 횡수는 큰 차이가 난다.

본 논문에서는 실시간 물수제비 시뮬레이션을 위해 개선된 역학적 모델을 제안한다. 회전과 공기 저항을 고려함으로써 이전 연구에 비해 보다 사실적인 물수제비 현상을 시뮬레이션 할 수 있다. 또 물표면의 움직임을 표현하기 위해서 [6]에서 제안한 방법을 실시간처리에 맞게 수정하여 사용한다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구결과들에 대해서 알아보고, 3장에서 돌과 수면의 충돌을 설명하며, 4장에서 실시간 물표면 처리 처리 과정을 설명한다. 5장과 6장에서 각각 시뮬레이션의 결과와 결론 및 향후 연구 방향에 관하여 기술한다.

2. 관련 연구

유체와 강체사이의 상호작용에 대한 연구는 유체 역학과 컴퓨터 그래픽스 분야에서 오랜 기간 동안 진행 되어왔다. 특히 계산 유체 역학은 컴퓨터의 계산 능력 향상과 함께 비약적인 발전을 거듭하고 있다. 이런 계산 유체 역학 기법들은 유체를 표현하는데 유용하게 사용되었지만, 계산량이 방대할 뿐만 아니라 유체의 흐름을 제어하기가 쉽지 않기 때문에 컴퓨터 그래픽스 분야에 적용하기가 쉽지 않다.

컴퓨터 그래픽스 분야에서도 Stam의 연구를 기점으로 유체 시뮬레이션에 대한 연구가 활발히 진행되고 있다[7]. Stam은 Navier-Stokes 방정식의 비선형 항인 대류항을 계산하기 위해 semi-lagrangian 방법과 암묵적 적분법을 이용하여 안정적이면서 빠른 유

체 시뮬레이션을 구현하였다. Stam의 방법은 이후 많은 연구자들에 의해 더욱 개선되었다[8-10]. 하지만 기존의 접근법들은 많은 계산시간을 필요로 하여 게임이나 가상현실과 같이 실시간 처리를 요하는 환경에서는 적용하기가 힘들다.

물수제비 현상에 대한 물리학적 접근은 Bocquet의 연구로부터 시작한다[1]. Bocquet은 알루미늄 원반을 자동 발사하는 장치와 고속 카메라를 이용하여 알루미늄 원반이 수면과 부딪치는 장면을 촬영하였다. 이를 통해 물수제비 횡수는 돌의 속도가 빠를수록 증가하며, 원반과 수면과의 각도가 20도를 이룰 때 최대의 반복 횡수를 얻을 수 있는 것을 알게 되었다. 또 원반과 수면의 충돌에서 원반의 속도가 일정량 이상이 되면 항력이 원반의 무게 이상으로 작용하여 원반이 수면 위로 튀어 오름을 수식적으로 증명하였다.

Nagahiro 등은 ODE 모델을 이용하여 최대 물수제비 횡수를 얻을 수 있는 최적의 속도와 원반과 수면의 각을 구했다[11]. 수면과 원반의 상호 작용을 계산하기 위해 Smoothed Particle Hydrodynamics (SPH)를 이용하였다. SPH 또한 많은 계산을 요구한다는 단점이 있다.

도주영 등은 물수제비 현상을 물리기반 시뮬레이션 방법을 이용하여 실시간으로 시뮬레이션 하였다[5]. 하지만 항력을 수직항력과 수평항력으로 분해해서 적용함으로써 수식이 복잡하여 직관적으로 이해하기가 힘들고 더욱이 회전을 고려하지 않았다. 회전은 물수제비 현상을 설명하는데 중요한 요소이지만 이에 대한 고려가 없어 [5]의 시뮬레이션에서 물수제비는 직선운동만 한다.

본 논문에서는 실시간으로 물수제비의 3차원 운동을 시뮬레이션 할 수 있는 개선된 역학 모델을 제안한다. 제안하는 기법은 적은 계산량으로도 적절한 사실성을 보장 할 수 있다.

3. 물수제비 현상을 위한 역학 모델

수면을 향해 던진 돌맹이는 4가지 힘의 영향을 받는다. 사람이 던지는 힘, 중력, 공기에 의한 저항력, 마지막으로 물과 충돌했을 때 수면과의 충격력이다. 물수제비 현상을 시뮬레이션 하기 위해서는 4가지 힘을 모두 포함하는 역학 모델을 사용해야 한다. 하지만 [5]에서는 공기에 의한 저항력을 고려하지 않았

다. 따라서 수면과 충돌 후 돌맹이의 자세각 변화에 따른 공기 저항력의 변화를 시뮬레이션에 반영하지 않았다. 자세각이란 돌맹이의 진행 방향과 돌맹이의 법선 벡터 사이의 각을 말한다. 또한 회전에 대한 고려도 없었다.

본 논문에서는 이들 모두를 고려한 역학 모델을 제안한다. 사람이 던지는 힘과 중력은 기존의 역학 모델에서 쉽게 모델링 할 수 있으므로 공기에 의한 저항력과 수면과의 충돌과정에서 생기는 충격력에 대해 살펴보기로 한다. 먼저 수면과의 충돌 모델부터 설명하겠다.

Bocquet은 얇은 두께의 원반과 수면 사이에서 발생에서 발생하는 충돌을 해석하는 모델을 제시했다 [1]. 하지만 본 논문에서는 [5]와 같이 삼각형 메시들로 구성된 돌맹이를 사용하여 보다 사실적인 시뮬레이션을 수행한다(그림 1.a). 또 계산량을 줄이기 위해 그림 1.b와 같이 돌맹이의 디스크 모델도 가진다. 즉 시뮬레이션 과정에서 메시 모델과 디스크 모델을 모두 채택하여 필요에 따라 사용한다. 메시 모델은 수면과의 충돌을 처리할 때 사용하고 디스크 모델은 공기 저항에 의해 생기는 힘을 고려할 때 사용한다.

돌맹이를 수면으로 던지면 돌맹이는 수면 아래로 가라앉기도 하고 수면 위로 튕겨 나기도 한다. 또 돌맹이의 속도와 입수각에 따라 여러 번 수면 위에서 물수제비를 뜨기도 한다. 물수제비 횡수에 영향을 미치는 요소는 수면과 충돌하는 순간에 돌맹이의 속도, 수면과 돌맹이의 접촉 면적 그리고 돌맹이의 회전이다. 앞의 두 요소는 돌맹이와 수면의 충돌로 발생하는 수직 항력을 결정짓는 중요한 물성이다. 수직 항력이 돌맹이의 무게 이상으로 작용할 때 돌맹이는 수면 위로 튕겨져 오른다. 또한 돌맹이의 회전은 수면 위로 튕겨 나온 돌맹이의 운동 에너지 손실을 최소화 하여 물수제비 운동을 유지하는 관건이 된다.

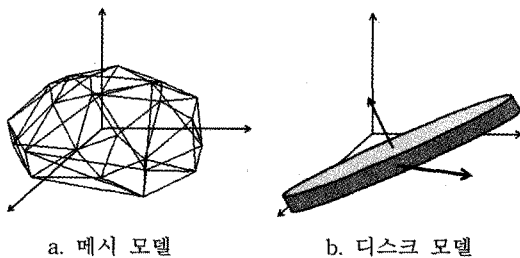


그림 1. 돌맹이 메시 모델과 대표 디스크

즉 회전은 수면과 충돌로 인하여 생기는 돌맹이의 자세각의 변화를 최소화시켜 공기에 의한 저항을 줄여 준다. 물수제비 운동을 하는 돌맹이의 궤적이 수면과 충돌할 때마다 변하는 것 또한 회전의 영향이다. 따라서 물수제비 현상을 바르게 이해하기 위해서는 이 세 요소를 모두 고려해야 한다.

이에 우리는 자연스러운 물수제비 현상을 시뮬레이션 하기 위해 돌맹이의 회전을 고려한 모델을 제안하며, 직관적으로 이해하기 쉬운 수식들을 제안한다.

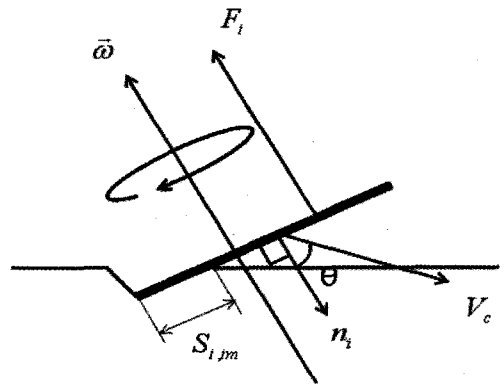


그림 2. 돌맹이의 한 메시 m_i 와 수면 사이의 충돌

그림 2는 수면과 충돌한 돌맹이 메시의 한 삼각형에 대한 묘사이다. $i(1 \leq i \leq n)$ 는 돌맹이 메시의 특정 삼각형 인덱스를 나타낸다. 그림 2에서 $S_{i,im}$ 은 수면 아래로 잠긴 면적을, \vec{F}_i 는 메시의 i 번째 삼각형에 가해지는 항력을 나타낸다. 수면과 충돌한 돌맹이가 다시 수면 위로 떠오르는 것은 항력 \vec{F}_i 때문이다. 결국 돌맹이의 모든 메시가 물수제비 운동에 영향을 주는 것이 아니라 수면 아래로 내려간 메시들만이 고려 대상이다.

본 논문에서는 \vec{F}_i 를 계산하기 위하여 뉴턴의 항력 방정식을 이용한다[12,13]. 항력이 발생하려면 돌맹이의 속도벡터와 수면사이에 각도가 90도 이하인 상태로 돌맹이가 입수해야 한다. 즉 항력은 돌맹이의 무게중심 속도인 \vec{V}_c 와 삼각형 평면의 법선벡터 \vec{n}_i 사이의 각 θ 가 90도 이하일 경우에만 발생된다. 이제 각속도 $\vec{\omega}$ 로 회전하면서 속도 \vec{V}_c 로 움직이는 돌맹이가 수면과 접촉하는 경우, 돌맹이의 삼각형들 중에서 항력 발생의 필요조건을 만족하는, 즉 θ 가 90도 이하

인 삼각형 i 에 의해 생기는 항력 \vec{F}_i 는 다음과 같이 구할 수 있다.

$$\begin{aligned} \vec{F}_i &= -(\rho \vec{V}_{i,p} \cdot \vec{n}_i S_{i,im} \vec{V}_{i,p} \cdot \vec{n}_i) \vec{n}_i \\ \vec{V}_i &= \vec{r}_i \times \vec{\omega} \\ \vec{V}_{i,p} &= \vec{V}_c + \vec{V}_i \end{aligned} \quad (1)$$

식 (1)에서 $\vec{V}_{i,p}$ 는 회전 속도와 선속도의 합인 실제 속도를, ρ 는 물의 밀도를, \vec{r}_i 는 돌맹이 중심으로 부터의 메시 i 까지의 거리를 나타낸다. 식 (1)의 항력을 이용하여 토크를 구할 수 있다. 즉 삼각형 평면이 받는 토크 $\vec{\tau}_i$ 는 평면의 항력 \vec{F}_i 와 돌맹이 중심으로 부터의 평면의 거리벡터 \vec{r}_i 의 외적으로 표현된다.

$$\vec{\tau}_i = \vec{F}_i \times \vec{r}_i \quad (2)$$

식 (1)과 (2)를 이용하여 항력 발생 조건을 만족하는 삼각형의 힘과 토크를 계산한 뒤 이를 다 합하여 돌맹이가 받는 힘과 토크의 총량을 계산할 수 있다.

$$\vec{F} = \sum_{i=1}^n \vec{F}_i, \quad \vec{\tau} = \sum_{i=1}^n \vec{\tau}_i \quad (3)$$

이제 돌맹이가 공기 중에서 움직일 때 발생하는 힘에 대해 고려해보자. 돌맹이와 공기 사이의 충돌로 인해 운동량의 변화가 생긴다. 이는 돌맹이가 수면과 충돌하면서 생기는 항력과 같은 방법으로 뉴턴의 항력 방정식을 이용하여 양력과 항력을 구할 수 있다. 수면 충돌의 경우와 같이 매 시간 돌맹이의 모든 삼각형에 대해서 고려해야 하지만, 계산량을 줄이기 위해서 돌맹이가 두께가 없는 디스크라고 가정하고 계산을 간소화한다. 따라서 디스크의 법선벡터만 있으면 식 (1)과 (2)를 이용하여 공기와의 충돌로 생기는 힘을 계산할 수 있다.

$$\begin{aligned} \vec{F}_{air} &= -(\rho \vec{V}_p \cdot \vec{n} S_{disk} \vec{V}_p \cdot \vec{n}) \vec{n} \\ \vec{V}_r &= \vec{r} \times \vec{\omega} \\ \vec{V}_p &= \vec{V}_c + \vec{V}_r \\ \vec{\tau}_{air} &= \vec{F}_{air} \times \vec{r} \end{aligned} \quad (4)$$

식 (4)에 S_{disk} 는 디스크라고 가정한 돌맹이의 면적이며, \vec{n} 은 디스크의 법선벡터이다. 이제 물수제비에 가해지는 4가지 힘을 모두 구했다. 즉 시뮬레이션 과

정에서 계속적으로 돌맹이에 적용되는 힘은 중력과 식 (4)에서 계산된 공기역학적 힘 \vec{F}_{air} 이다. 만일 돌맹이가 수면과 충돌한 경우에는 식 (3)의 힘도 계산해서 돌맹이에 가한다.

위 식들을 적용하여 시뮬레이션한 결과를 렌더링 하기 위해서는 돌맹이의 위치와 방향이 필요하다. 즉 식 (3)과 (4)의 힘을 두 번 적분하여 위치와 방향을 구할 수 있다. 하지만 계산의 편의를 위해서 힘 F 를 직접 사용하기 보다는 운동량 P 를 이용하여 위치와 방향을 계산한다. 따라서 그림 3처럼 강체 시뮬레이터는 시뮬레이션 시작 시간에 돌맹이의 초기 위치 x , 방향 q , 운동량 P 그리고 각운동량 L 을 입력으로 주면 상미분 솔버를 이용하여 다음 시간 간격에서의 돌맹이의 위치, 방향, 운동량량, 각운동량을 구해준다. 이 때 식 (3)과 (4)는 입력 F 를 계산하는데 사용된다. 상미분 솔버의 자세한 과정은 [14]를 참고하라.

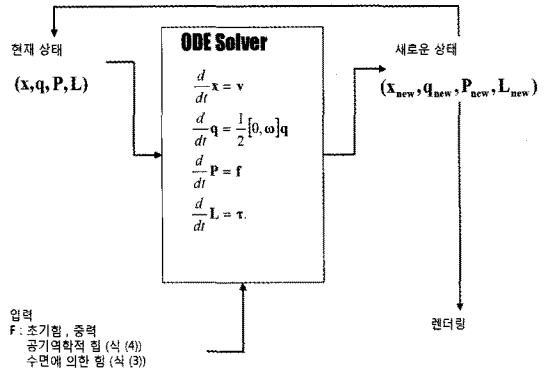


그림 3. 강체 시뮬레이션의 처리 과정

4. 수면모형

돌맹이의 물수제비 운동으로 인해 생기는 물표면의 변화도 우리가 고려해야 할 요소이다. 본 논문의 시뮬레이션은 실시간성을 위해 [6]의 표면 변위 평가 방식을 사용하여 물 표면의 변화를 모델링한다. 이 방식은 2차 파동방정식을 근거로 하며 2차 파동 방정식은 다음 식으로 표현된다.

$$\frac{\partial^2 z}{\partial t^2} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - \mu \frac{\partial z}{\partial t} \quad (5)$$

식 (5)에서 상수 c 는 단위 시간 당 거리이다. 이것은 파동이 매질을 통해 전파되는 속력을 의미한다.

매체가 받는 장력이 커질수록 파동의 속력이 빨라지고, 매체의 밀도가 높아질수록 파동의 속력이 느려지는 것과 같다. μ 는 음이 아닌 상수로, 유체의 점성(viscosity)을 의미하며 유체의 표면에서 파동이 잦아들기까지 걸리는 시간을 결정한다. 점성에 의한 감쇠력(damping force)에 관여한다. 식 (5)과 같은 점성에 따른 감쇠를 가진 2차원 파동 방정식은 변수들을 분리하여 해석적으로 풀 수 있다. 그러나 그 방법은 상당히 복잡하여서 실시간 시뮬레이션에서 활용하기엔 계산량이 너무 많다. 따라서 유체 표면 위에서 파동의 전파를 높이 필드(Height Field)로 모델링하는 수치적 방법을 사용한다. 높이 필드는 시간 t 에서 좌표 (i, j) 에 대한 높이 $z(i, j, t)$ 로 표현한다. 식 (5)의 편미분을 중앙 차분 근사로 표현하면 다음과 같다.

$$\begin{aligned} & \frac{z(i, j, t+1) - 2z(i, j, t) + z(i, j, t-1)}{t^2} \\ &= c^2 \frac{z(i+1, j, t) - 2z(i, j, t) + z(i-1, j, t)}{d^2} \\ &+ c^2 \frac{z(i, j+1, t) - 2z(i, j, t) + z(i, j-1, t)}{d^2} \\ &- \mu \frac{z(i, j, t+1) - z(i, j, t-1)}{2t} \end{aligned} \quad (6)$$

식 (6)에서 상수 d 는 인접 좌표 사이의 거리이다. 현재 시간의 변위 $z(i, j, t)$ 와 이전 시간의 변위 $z(i, j, t-1)$ 을 알고 있으면, 시간 t 가 흐른 뒤의 변위 $z(i, j, t+1)$ 을 알 수 있다. 식 (6)을 $z(i, j, t+1)$ 에 대하여 풀면 다음 식을 얻을 수 있다.

$$\begin{aligned} z(i, j, t+1) &= \frac{4 - 8c^2t^2/d^2}{\mu t + 2} z(i, j, t) + \frac{\mu t - 2}{\mu t + 2} z(i, j, t-1) \\ &+ \frac{2c^2t^2/d^2}{\mu t + 2} [z(i+1, j, t) + z(i-1, j, t) + z(i, j+1, t) + z(i, j-1, t)] \end{aligned} \quad (7)$$

식 (7)을 사용하여 높이 필드의 각 정점의 높이 변위를 계산할 때, 각 방향의 상수들은 미리 계산해 두면 실제 시뮬레이션 때 계산을 피할 수 있다. 따라서 각 정점마다 총 세 번의 곱셈과 다섯 번의 덧셈만 수행한다.

이때 식 (7)은 그림 4a와 같이 각 정점의 상하좌우 4방향의 인접 정점의 높이 변이만을 참조한다. 이 경우는 수면의 파장이 완전한 동심원의 모습을 그리지 못하는 문제점이 생길 수 있어, 그림 4b와 같이 대각선 방향의 인접 정점에 대한 높이 변이를 참조하는 항을 추가하여 식 (8)로 개선하였다.

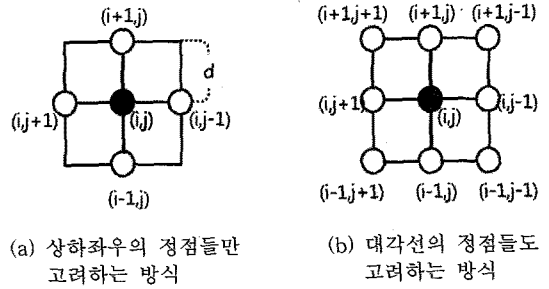
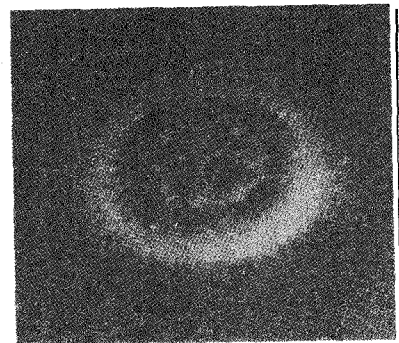


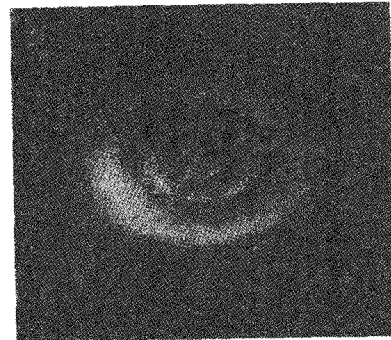
그림 4. 높이 필드

$$\begin{aligned} z(i, j, t+1) &= \left(-\frac{4}{\mu t + 2} - \frac{10}{3}\xi\right) z(i, j, t) + \frac{\mu t - 2}{\mu t + 2} z(i, j, t-1) \\ &+ \frac{2\xi}{3} [z(i+1, j, t) + z(i-1, j, t) + z(i, j+1, t) + z(i, j-1, t)] \\ &+ \frac{1\xi}{3} [z(i+1, j+1, t) + z(i-1, j-1, t) + z(i-1, j+1, t) + z(i+1, j-1, t)] \end{aligned} \quad (8)$$

위 식에서 ξ 는 $(2c^2t^2/d^2)/(\mu t + 2)$ 를 의미한다. 식 (8)에서 상하좌우의 정점들에는 가중치 2/3를, 대각선의 정점들에는 가중치 1/3을 주는데 이는 대각선의 정점들과 상하좌우의 정점들의 거리비인 $\sqrt{2}:1$ 임을 고려하여 준 수치이다. 그림 5a는 식 (7)을 적용하여 표현된 물결의 모양이며 그림 5b는 식 (8)를 사용



(a) 상하좌우의 정점들만 고려하는 방식



(b) 대각선의 정점들도 고려하는 방식

그림 5. 수치적 방법으로 표현된 물결

하여 얻는 물결의 모양이다. 대각선 인접점점의 참조항의 추가로 인하여 식 (7)보다 계산량은 늘어났지만, 적은 양의 계산증가로 파장의 모습을 원에 가까운 모습으로 만들 수 있다.

식 (6)을 식 (8)로 바꾸어 쓸 경우, 수치적 불안정성이 문제가 될 수 있다. 속력 c 가 너무 빠르거나 시간 간격 t 가 너무 긴 경우, 식 (8)를 반복 계산하다보면 발산할 수도 있다. 이를 위해 현재 시간에서 높이의 변화가 생긴 정점들은 다음 시간 간격 후에 높이가 0이 되는 쪽으로 움직인다고 하면 변위는 시간이 지날수록 점점 줄어든다고 가정할 수 있다. 이 가정을 만족시키는 속도 c 와 시간 간격 t 은 다음과 같은 식으로 찾을 수 있다.

$$0 < c < \frac{d}{2t} \sqrt{\mu t + 2}$$

$$0 < t < \frac{\mu + \sqrt{\mu^2 + 32c^2/d^2}}{8c^2/d^2} \tag{9}$$

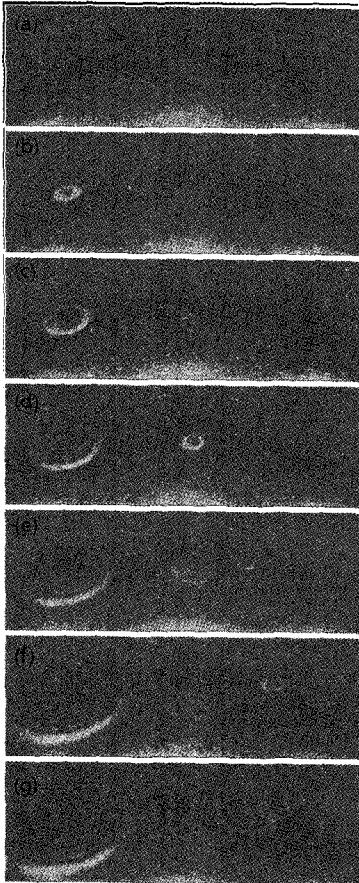


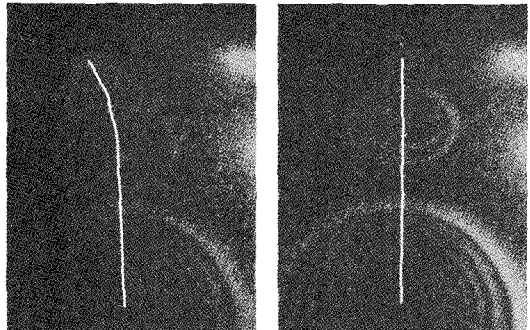
그림 4. 물수제비 운동 시뮬레이션 렌더링 결과

5. 시뮬레이션 결과

본 논문의 물수제비 운동 시뮬레이션은 DirectX9.0c를 기반으로 Intel Core 2 6300 1.86GHz CPU와 GeForce 9600 512MB 비디오카드, 2046MB RAM의 사양을 가진 시스템에서 구현되었다. 44개의 삼각형 메시로 구성된 하나의 돌맹이 객체와 500×500의 정점으로 구성된 높이 필드를 사용하여 성능을 실험하였다. 그 결과는 돌맹이와 물표면의 광원 및 텍스처(texture)에 대한 실시간 렌더링까지 포함하여 평균 90fps 이상의 성능을 나타내었다. 본 논문의 시뮬레이션은 기존에 고려되지 않았던 회전을 고려함으로써 돌맹이의 회전으로 인하여 발생하는 물수제비 운동의 변화를 보여주었다. 아래의 그림들은 그러한 시뮬레이션이 구동되는 모습들이다. 그림 4와 그림 5는 물수제비 현상을 본 논문에서 제안하는 기법으로 구현된 시뮬레이션의 결과이다. 그림 4는 시뮬레이션이 실행되는 것을 연속적으로 본 장면들이며 돌맹이와 수면의 충돌반응으로 인하여 돌맹이가 물수제비 운동을 하는 것을 관찰할 수 있다.

그림 5는 동일 속력으로 돌맹이를 던질 때, 회전 유무로 인한 물수제비 운동의 차이를 보여준다. 그림 5.a는 회전의 영향으로 수평방향으로 휘어진 돌맹이의 궤적을, 그림 5.b는 회전이 없는 경우 돌맹이의 궤적을 보여준다. 그림 5에서 확인 할 수 있는 것처럼 동일한 속도로 움직인다 하더라도 회전이 더해지면 더 많은 물수제비 횃수를 얻을 수 있다.

그림 6은 카메라를 수면 위와 옆 쪽에 배치시키고 시뮬레이션한 결과이다. 물수제비 운동을 하는 돌맹이 위치의 변화를 연속적으로 나타내었다.



(a) 제안하는 방법의 결과 (b) 이전 연구의 결과

그림 5. 회전의 영향으로 휨 돌맹이의 궤적

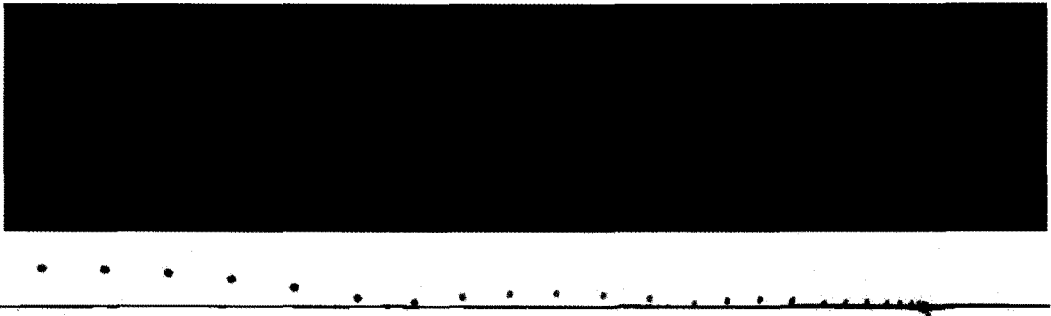


그림 6. 물수제비 운동을 하는 돌맹이의 위치 변화

그림 7은 각속도의 변화에 따른 물수제비의 횡수를 나타낸 그래프이다. 가로축은 각속도를 나타내면 세로축은 물수제비의 횡수를 나타낸다. 그림 7에서 확인 할 수 있는 것처럼 회전이 없는 경우와 회전이 있는 경우의 물수제비의 횡수는 많은 차이를 보이고 있다. 따라서 회전이 물수제비 운동의 중요한 요인 중 하나임을 알 수 있다.

그림 8은 8 m/sec의 속도로 돌맹이를 던질 때 회전의 유무와 운동량의 변화에 대한 그래프이다. 그림 8에서 가로축은 프레임을 세로축은 운동량 성분 중 x 성분을 나타낸다. 회전이 없는 경우 3회 물수제비

운동을 하는데 수면과 충돌 후 운동량이 급격히 감소함을 알 수 있다. 반면 회전이 있는 경우는 운동량이 완만히 감소함을 확인 할 수 있다. 따라서 더 많은 물수제비 운동을 하였다.

6. 결 론

본 논문은 회전이 고려된 물수제비 운동을 실시간 3차원 물리기반 모델링 기법으로 구현하였다. 사실적인 물수제비 운동을 시뮬레이션하기 위해서 공기 저항으로 생기는 영향과 회전을 포함하는 개선된 역학 모델을 제안했다. 회전은 수면과 충돌 후 돌맹이의 자세 변화를 최소화시켜 공기 저항으로 인한 운동량 감소를 줄여 준다. 따라서 회전이 없을 때보다 회전이 더해지면 더 많은 물수제비를 뜰 수 있다.

돌맹이와의 충돌로 인한 수면의 형태변화를 실시간으로 표현하기 위해서는 이전보다 간단한 수치적 모델을 적용하였다. 수면을 구성하는 정점의 높낮이 변위를 계산할 때 반복되는 과정을 간소화하고 미리 처리 할 수 있는 부분을 계산해 둬으로써 계산량을 줄였다. 이 결과로 줄인 계산 시간을 기존의 수면 모델을 개량하는데 사용하여 적은 계산량의 증가로 보다 사실적인 물표현 효과를 얻었다.

하지만 돌맹이의 운동과 수면의 반응에 관련한 수식 간략화에도 불구하고 CPU에 의존하는 렌더링은 시뮬레이션 성능 향상에 제약을 가져왔다. 따라서 GPU를 활용한 렌더링 시간 단축과 물리 계산의 병렬화는 이후의 시뮬레이션의 성능 향상을 위한 중요한 연구 과제이다. 특히 최근에 물리기반 모델링에 관한 처리를 하드웨어 레벨에서 지원하는 GPU들이 발표되었다. 이것은 상호작용이 가능한 수면 반응을 GPU만 활용하여 표현하거나 또는 기존 CPU의 부담

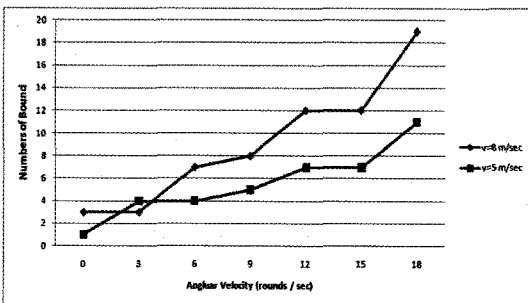


그림 7. 회전 정도에 따른 바운드 수

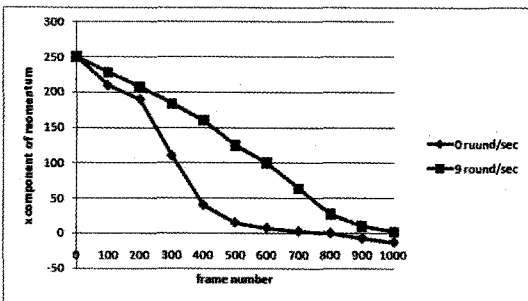


그림 8. 회전 유무에 따른 운동량의 변화

을 크게 줄여줄 수 있음을 의미한다. 시뮬레이션의 성능향상을 위해 이러한 기술들의 활용은 새로운 과제이다.

본 논문의 시뮬레이션에서 시스템 성능에 가장 큰 영향을 미친 요소는 수면의 크기였다. 수면을 처리하는데 필요한 계산량은 수면 크기에 제곱에 비례하여 증가하기 때문이다. 이러한 제약은 바다나 강같이 넓은 환경을 무대로 하는 게임이나 시뮬레이션에서 시스템 성능의 저하가 없이 본 논문의 결과를 적용하는 것을 어렵게 한다. 그러나 이러한 문제는 새로운 GPU들의 성능을 활용하여 시뮬레이션의 실시간 성능을 보다 더 향상시키는 것으로 해결될 수 있을 것이다. 또한 그러한 결과는 본 논문의 제안을 적극적으로 활용할 수 있는 것과 더불어 더욱 사실적인 수면의 표현도 병행할 수 있을 것을 기대한다.

참 고 문 헌

- [1] L. Bocquet, "The physics of stone skipping," American Journal of Physics, Vol. 71, Issue 2, pp. 150-155, 2003.
- [2] T. Takahashi, H. Ukei, A. Kunimatsu, and H. Fujii, "The simulation of fluid-rigid body interaction", Proc. of SIGGRAPH '02, pp. 266-266, 2002.
- [3] M. Carlson, P. J. Mucha and G. Turk, "Rigid Fluid: Animating the Interplay between Rigid Bodies and Fluid", Proc. of SIGGRAPH '04, pp. 377-384, 2004.
- [4] C. Batty, F. Bertails, and T. Bridson, "A Fast Variational Framework for Accurate Solid-Fluid Coupling", Proc. of SIGGRAPH '07, pp. 99, 2007.
- [5] 도주영, 라은철, 김은주, 유관우, "물리기반 해석을 통한 물수제비 운동 시뮬레이션", 정보과학회논문지, 제33권, 제3·4호, pp. 147-156, 2006.
- [6] E. Lengyel, Mathematics for 3D Game Programming & Computer Graphic, Game Development Series, 2002.
- [7] J. Stam, "Stable fluids," Proc. of SIGGRAPH '99 pp. 121-128. 1999.
- [8] M. Carlson, P. J. Mucha, and G. Turk, "Rigid fluid: animating the interplay between rigid bodies and fluid" ACM Trans. Graph., Vol. 23, No. 3, pp. 377-384, 2004.
- [9] H.-M. Hong and C.-H. Kim, "Discontinuous fluids," ACM Trans. Graph., Vol. 24, No. 3, pp. 915-920, 2005.
- [10] O.-Y. Song, H. Shin, and H.-S. Ko, "Stable but nondissipative water," ACM Trans. Graph., Vol. 24, No. 1, pp. 81-97, 2005.
- [11] S. Nagahiro and Y. Hayakawa, "Theoretical and Numerical Approach to Magic Angle of Stone Skipping," Physics Review Letters, Vol. 94, No. 174501, 2005.
- [12] D. Halliday and R. Resnick, Fundamentals of Physics, John Wiley & Sons, 2005.
- [13] L. N. Long and H. Weiss, "The Velocity Dependence of Aerodynamic Drag: A Primer for Mathematicians," American Mathematical Monthly, Vol. 106, No. 2, 1999, pp. 127-135.
- [14] A. Witkin and D. Baraff, Physically Based Modeling, SIGGRAPH 2001 Course Notes.



이 남 경

- 1998년 경북대학교 컴퓨터공학과
학사
- 2000년 경북대학교 컴퓨터공학과
석사
- 2007년 경북대학교 컴퓨터공학과
박사
- 2008년~2009년 경북대학교 전자

전기컴퓨터학부 강의초빙교수

2010년 현재 경북대학교 BK21 Post Doc.

관심분야: 물리 기반 시뮬레이션, 실시간 렌더링



백 낙 훈

- 1990년 한국과학기술원 전산학과
졸업 (학사)
- 1992년 한국과학기술원 전산학과
(공학석사)
- 1997년 한국과학기술원 전산학과
(공학박사)
- 2004년~현재 경북대학교 IT대학

컴퓨터학부 교수

관심분야: 모바일 그래픽스, 리얼타임 그래픽스