

# 물리기반 해석을 통한 물수제비 운동 시뮬레이션

(Motion of Stone Skipping Simulation by Physically-based Analysis)

도주영<sup>†</sup>    라은철<sup>\*\*</sup>    김은주<sup>\*\*\*</sup>    유관우<sup>\*\*\*\*</sup>  
(Joo-Young Do)    (Eun Chul Ra)    (Eun-ju Kim)    (Kwan-Woo Ryu)

**요약** 물리기반 시뮬레이션 모델링 기법은 기하학적 속성만을 사용하여 모델링한 다른 시뮬레이션과는 달리 뉴턴의 제2법칙과 같은 물리 법칙을 사용하여 현실세계를 시뮬레이션 한다. 본 논문에서는 현실 세계에서 일어나는 물수제비 현상의 물리적 운동 원리를 수식기반 해석을 통해 실제와 같은 물리기반 시뮬레이션 시스템을 제안한다. 또한 본 논문에서는 물의 표면과 돌맹이의 상호작용을 표현하는데 있어서 물의 자연스러운 움직임 현상과 함께 돌맹이의 궤적을 계산하는데 중점을 두었다. 궤적은 돌맹이의 속도와 물의 항력에 의해 결정되고, 이러한 운동은 돌맹이가 물 표면에서 가라앉을 때까지 반복 계산된다. 본 논문에서 제안하는 자연스러운 물수제비 운동은 운동역학적 방법을 사용하여 실시간에 사실적으로 표현된다. 그리고 이러한 물수제비 운동을 PC 환경에서 대화형으로 재현하여 유사한 운동을 하는 3차원 모델들에도 적용할 수 있다.

**키워드** : 물리기반 시뮬레이션, 3D 렌더링, 유체역학, 물수제비

**Abstract** Physically-based simulation modeling is to simulate the real world by using physical laws such as Newton's second law of motion, while other modelings use only geometric properties. In this paper, we present a real time simulation of stone skipping by using the physically-based modeling. We also describe interaction of a stone on the surface of water, and focus on calculating the path of the stone and the natural phenomena of water. The path is decided by velocity of the stone and drag force from the water. The motion is recalculated until the stone is immersing into the water surface. Our simulation provides a natural motion of stone skipplings in real time. And the motion of stone skipplings are generated by give interactive displays on the PC platforms. The techniques presented can easily be extended to simulate other interactive dynamics systems

**Key words** : Physically-based Simulation, 3D Rendering, Fluid Dynamics, Stone Skipping

## 1. 서론

대부분의 사람들이 호수나 강물에 돌맹이를 던져서 돌맹이가 수면 위를 튀기게 하는 물수제비를 해보았거나 관찰해 보았을 것이다. 일반적으로 잔잔한 수면에 바닥이 평평한 돌을 던질 때 돌맹이가 물위로 여러 번 튀

어 오르는 현상이 나타난다. 물수제비 운동(stone skipping)은 수면위로 던져진 돌이 수면과 충돌 시에 물 표면의 항력(drag force)에 의해 수면 위로 다시 튀어 오르는 운동이다. 물 표면에 강체가 충돌하였을 때 물표면의 반응은 기존에 물리 기반 시뮬레이션 분야에서 여러 연구가 있었으나, 강체가 물과의 상호작용으로 물수제비 운동과 같은 형태를 가지는 현상에 관한 연구는 없었다. [1,2]

본 논문에서 구현한 물수제비 운동의 형태는 물리적으로 해석이 가능하다[3]. 시뮬레이션에서 돌맹이는 얇은 두께와 일정한 질량을 가진 삼각 메쉬(triangle mesh)로 이루어진 회전체이며, 3차원 벡터(vector)성분을 가진 모델로서 속도와 체공시의 자세각에 대한 정보를 가지고 있다. 자세각(attitude angle)이란 비행 중인

<sup>†</sup> 학생회원 : 경북대학교 컴퓨터공학과  
voronoi@nate.com

<sup>\*\*</sup> 정회원 : 디토소프트 기술연구소  
ramer@nate.com

<sup>\*\*\*</sup> 정회원 : 동명정보대학교 정보통신공학과 교수  
ejkim@tit.ac.kr

<sup>\*\*\*\*</sup> 종신회원 : 경북대학교 컴퓨터공학과 교수  
kwryu@bh.knu.ac.kr

논문접수 : 2004년 9월 16일  
심사완료 : 2005년 11월 22일

물체가 수평면과 이루는 각으로 가정한다. 이 각은 돌맹이와 수면의 충돌 경사각(tilt angle)으로써 항력의 중요한 인자인 물 표면과의 접촉 면적을 결정한다. 시뮬레이션을 시작할 때 물수제비 운동을 하는 돌맹이에 대한 물리적인 제약(constraint)과 각 물체의 물리적 성질을 설정함으로써 다양한 결과를 얻을 수 있다. 본 시뮬레이션은 강체(rigid body)인 돌맹이가 유체(fluid)인 물에 충돌하면서 일어나는 힘의 작용들에 대한 해석을 하여 줌으로써, 자연 현상에서 보이는 물수제비 운동을 사실감 있게 묘사한다.

기존의 물리학 분야의 연구에서 물수제비 운동은 속도와 경사각에 따른 접촉 면적과, 돌맹이의 무게, 물의 밀도에 관한 물리 수식으로 설명되었다[3]. 여기에서 고려해야 할 유체와 강체의 차이점은 강체는 모션(motion)이 변환되어도 각 부분의 속도(velocity)와 형상(shape)이 일정하지만, 유체는 각 부분의 형체가 변화되고, 속도도 달라진다는 차이점이 있다. 본 논문에서는 강체의 운동은 발사체(projectile) 운동과 물 표면에 충돌 시 유체에서 발생한 항력의 작용으로 계산하였고, 유체의 반응은 강체에서 발생하는 위치 정보를 이벤트(event)로 처리하여 O'Brien[4]의 방법을 이용하였다. O'Brien의 물 표면 모델은 수식이 복잡하고 계산량이 많은 단점이 있는데, 본 논문에서는 이를 보완하여 간략화된 물 표현식을 실시간으로 계산한다. 또한 강체와 유체의 상호작용은 에너지 보존법칙을 만족시키고, 전체적인 운동 방식이 물리법칙을 따르게 된다.

본 논문에서 제시하는 돌맹이와 물의 충돌 시뮬레이션 모델은 크게 두 부분으로 나눌 수 있다. 돌맹이에 해당하는 강체의 운동 부분과 물 표면의 반응을 보여 주는 유체의 파동 부분이다. 특히 전자인 강체의 운동 부분에 주안점을 두고, 강체가 유체에 충돌하였을 때의 수력학적 항력(hydrodynamic drag force)에 의한 반작용(reaction)을 해석하였다[5]. 강체가 유체의 경계면에 충돌 하였을 때 유체의 반작용에 의해 강체가 수면 밖으로 튕기는 물수제비 운동 대한 컴퓨터 그래픽스 분야의 연구는 아직 없으며, 물리학 분야에서는 Bocquet가 물수제비 운동을 순수 물리학적으로 분석한 연구가 있었다[3]. 이 연구는 실제제와는 다른 몇 가지 제약을 가정하고 있다. 물수제비 운동에서 돌맹이를 평판의 형태로 가정하고, x축과 z축만을 고려하여 물수제비 운동을 단순화 시키고, 이를 통하여 돌맹이의 최대 튕김 횟수와 물수제비 운동이 가능한 돌맹이의 기울기 각도와 속도를 구했다. 평판으로 가정한 돌맹이는 정사각형의 면이나 원형의 면이고, 높이는 무시 가능할 정도로 얇은 것으로 가정하였다. 이런 돌맹이의 형태는 실제 돌맹이의 모양과 다르고, 물과 접촉하는 면적에 대해서도 실제와

는 차이가 있다. 실제 돌맹이는 등근 형태로 두께도 얇지 않고, 표면도 고르지 않다. 또한 x, z축에서만 힘을 고려하고 y축에서 작용하는 힘도 고려하지 않고 있다. 따라서 본 논문에서는 이 연구에서 제안한 돌맹이에 대한 유체 항력의 물리적 해석을 기본으로 돌맹이의 형태와 x, y, z축에 대한 모든 힘에 대한 고려를 통하여 자연스러운 물수제비 운동의 시뮬레이션이 가능하도록 했다.

실시간의 자연스러운 물수제비 시뮬레이션 구현을 위해 첫 번째 단계에는 초기에 주어진 운동 에너지와 위치 에너지 값에 중력가속도가 작용하여, 시간에 대한 위치 변화량에 따라서 돌맹이가 발사체 운동을 해석하며, 두 번째는 위치 정보로부터 물 표면과 돌맹이의 충돌(collision)여부를 판단하고 이를 처리하는 단계로 이루어진다. 충돌 처리는 돌맹이와 수면사이의 접촉에 따른 반작용력을 구하는 것으로서, 접촉 면적과 접촉 시의 속도 벡터(vector)에 따라 항력이 발생한다. 마지막으로 충돌에 의한 항력 발생으로 변화된 위치 에너지와 운동 에너지를 구한다. 이 세 단계를 돌맹이가 물 위를 튕겨 나가지 못하고 물 속에 잠길 때까지 반복한다. 구현에서 핵심 부분인 물과 돌맹이의 접촉에서 발생하는 항력 계산과 항력의 적용 과정을 벡터를 이용한 운동 방식적으로 처리하였다. 이를 통해 실시간에 자연스러운 물수제비 운동 애니메이션을 시뮬레이션(simulation)의 결과로 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 돌맹이와 물의 충돌과정을 기술하고, 3장에서는 전체적인 시스템의 개요를 설명하고, 물수제비 운동 시뮬레이션의 3단계 과정에 대하여 기술하며, 4장에서는 제시된 시뮬레이션의 실시간 실행 결과를 보여준다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대하여 기술한다.

## 2. 물수제비 운동의 물리적 해석

물수제비 운동은 유체역학에서의 수력학적 항력(hydrodynamic drag force)에 기인하며, 이 힘에 의해 돌맹이와 물의 충돌에서 돌맹이가 물위로 튀어 오르는 현상이 발생한다[3].

Bocquet는 돌맹이와 물의 충돌에서 돌맹이의 속도가 일정량 이상이 되면 물의 항력은 돌맹이의 무게(mass) 이상으로 작용하여 돌맹이가 물위로 튕겨 오를을 물리적 수식으로 증명하였다. 이 수식은 에너지 보존 법칙을 만족함으로써 계산 유체 역학(computational hydro-mechanics)에 적용하여 시뮬레이션 할 수 있다. 그림 1은 Bocquet가 제안한 이상적인 돌맹이와 수면의 충돌 모델이다. 무게가  $M$ 이고 얇은 두께를 가진 납작한 돌맹이가 잔잔한 수면위로 던져지면 수면과 충돌하게 되

는데, 수면과 돌맹이 바다 사이의 경사각(tilt angle)은  $\theta$ 이다.  $V$ 는 돌맹이의 속도이며, 최소한의 비행이 가능한 속도로 가정한다. 물 표면과 돌맹이의 충돌에서 발생하는 반작용력을 구하기 위해서는 레이놀즈 수(Reynolds number)<sup>1)</sup>를 결정하여야 한다. 크기가  $a$  cm 일 때 레이놀즈 수는  $Re = Va/\nu$ 로 정의되는데,  $\nu$ 는 동역학 점성도(kinematic viscosity)를 말한다. 물에서  $\nu \approx 10^{-6} m^2/s$ 이며, 따라서 이 모델의  $Re \approx 10^5$ 이다. 충돌과정에서 수면과 충돌하는 돌맹이는 전체 면적에서 일부분만 참여하게 되므로 수면과 접촉하는 돌맹이의 면적  $S_{im}$ 에 비례하여 항력이 발생한다. 충돌 면에 수직으로 작용하는 형태항력과 수평으로 작용하는 마찰항력은 아래와 같다.

$$F = \frac{1}{2} C_l \rho_w V^2 S_{im} n + \frac{1}{2} C_f \rho_w V^2 S_{im} t \quad [3] \quad (1)$$

수식 (1)에서  $C_l$ 은 형태항력에 의한 충돌 면에서 돌맹이를 들어올리는 힘의 양력 계수(lift coefficient)이고,  $C_f$ 는 마찰항력에 의해 충돌 면에서 돌맹이의 속도를 감소시키는 마찰 계수(friction coefficient)이다.  $\rho_w$ 는 물의 밀도이며, 값은  $1000 kg/m^3$ 이다. 속도 성분  $V$ 는 초기 운동에너지와 위치에너지가 수면과의 충돌 시에 가진 속도 성분으로써 최초로 주어진  $V_{x0}$ 와  $V_{z0}$ 가 이 값을 결정한다. 단  $V_{z0} \ll V_{x0}$ 로 가정한다.

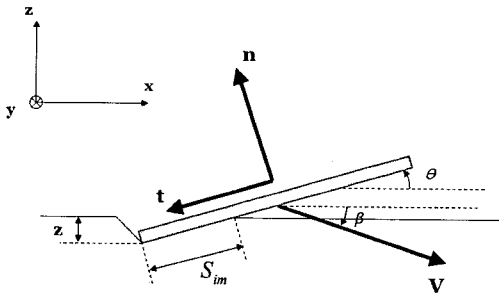


그림 1 돌맹이와 물 표면의 충돌 [3]

수식 (1)에서의 양력은 오직 물에 부분적으로 잠긴 표면에 의해서만 발생한다고 가정한다. 돌은 스스로 양력을 가진 균형 잡힌 형태가 아니며, 완전히 물에 잠긴 경우에도 양력이 발생하지 않는다. 그러므로 오로지 속도 성분과 물과 돌맹이 사이의 기울기 각에 의하여 양력이 결정된다. 위의 가정과 수식 (1)에 따라서  $x$ 축과  $z$ 축 각각에 대한 힘으로  $F$ 를 유도해 낼 수 있다. 이때

의 힘은 속도의 시간에 대한 미분과 중량의 곱으로 나타난다.

$$M \frac{dV_x}{dt} = -\frac{1}{2} \rho_w V^2 S_{im} (C_l \sin \theta + C_f \cos \theta) \quad [3] \quad (2)$$

$$M \frac{dV_z}{dt} = -Mg + \frac{1}{2} \rho_w V^2 S_{im} (C_l \cos \theta - C_f \sin \theta) \quad [3] \quad (3)$$

수식 (2)와 (3)은 돌맹이의 운동 방정식으로써 항력 성분과 함께  $z$ 축에 대한 중력가속도  $g$ 가 첨가되어 있다. 초기에 돌맹이에 주어진 운동에너지와 위치에너지는 물수제비 운동 동안에 소비된 에너지와 같다. 따라서 수식 (2)는 에너지 보존의 법칙을 만족한다. 수식 (3)에서 돌맹이가 튀어 오르기 위한 최소 속도를 돌맹이의 무게와 수면으로부터 발생하는 양력으로부터 유도할 수 있다. 운동 에너지가 수식 (2)에서 발생하는 마찰항력으로 인하여 점차적으로 사라진다. 여러 번의 충돌에 의하여 발생한 마찰 항력은 속도를 감쇄시켜, 속도가 튀어 오르기 위한 최소 속도 이하로 떨어지게 한다. 최소 속도 이하로 감속이 된 돌맹이는 수면 아래로 내려가게 된다.

### 3. 물수제비 운동 시뮬레이션

본 논문의 전체적 시뮬레이션의 진행은 그림 2와 같다. 먼저 초기화(initialize) 작업에서 물체의 모델링 정보와 운동에너지(kinetic energy), 위치에너지(potential energy) 등을 입력 받고 돌맹이의 발사체 운동에서 돌맹이의 위치 계산을 시작한다.

그림 2에서 설명한 바와 같이 프로그램 절차는 크게 3단계로 구분한다. 돌맹이의 위치를 계산하여 돌맹이의 입수형태와 위치를 결정하는 단계, 충돌을 검사하는 단계, 충돌 반응을 처리하는 단계로 되어있다. 이 과정은 재귀적으로 진행이 된다.

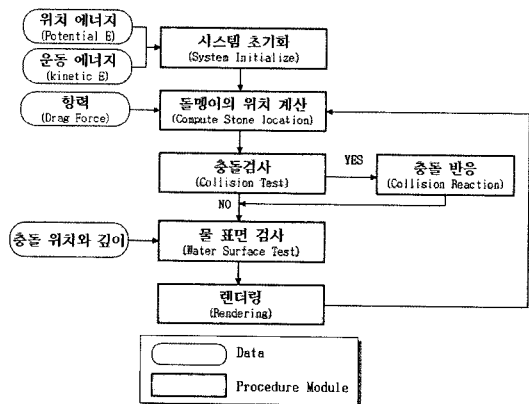


그림 2 시뮬레이션 흐름도

1) 유동(flow)하는 유체 내에 물체를 놓거나 관(tube) 속을 유체가 흐를 때에 그 흐름의 상태를 특징짓는 수치

3.1 돌맹이의 위치 계산과 충돌 검사

사용자가 입력한 위치와 속도에 따라서 돌맹이의 위치 계산을 시작한다. 돌맹이가 발사체 운동을 하기 때문에 주어진 속도 벡터에 중력가속도를 받아서 낙하 운동을 한다[6]. 시뮬레이션에서 공기저항은 생각하지 않기로 한다. 위치 정보  $pos$ 는 속도  $\vec{V}=(V_x, V_y, V_z)$ 에 의해 결정되며 수식은 아래와 같고 충돌이 일어날 때 까지 아래 수식에 따르는 운동을 한다.

$$pos_x = pos_x + V_x \cdot elapsedtime \quad (4)$$

$$pos_y = pos_y + V_y \cdot elapsedtime \quad (5)$$

$$V_z' = V_z - g \cdot elapsedtime \quad (6)$$

$$pos_z = pos_z + V_z \cdot elapsedtime + \frac{V_z' - V_z}{2} \cdot elapsedtime \quad (7)$$

$$V_z = V_z' \quad (8)$$

위치가 계산되면 충돌 계산과 모델링을 위해 그림 3의 (a)에 나타난 돌맹이의 외형을 이루는 좌표들 ( $C_1 \sim C_n$ )을 자세각(Attitude Angle)에 따라 회전(rotation) 시켜주고  $pos$ 에 따라서 좌표 변환(coordinate translation)을 하여 각 점의 현재 위치를 계산한다[7]. 좌표 변환에 쓰이는 돌맹이 모델은 그림 3과 같다. 그림 3(a)의 점들은 메쉬(mesh)를 구성하는 좌표 값들이며, 이들은 돌맹이의 외형을 나타내는 기본 정보로써 회전과 좌표 변화에 의해서 실제 돌맹이의 위치를 계산하는데 이용한다. 그림 3(b)는 돌맹이의 외형을 렌더링(rendering) 하고 항력을 계산하는 기본단위로 이용되는 삼각 메쉬(triangle mesh)들의 정보이다.

시뮬레이션을 시작할 때 입력된 자세각 정보는  $x, y, z$  각 축에 대하여 돌맹이가 기울어진 정도이다. 각 축에 대하여 돌맹이 모델의 각 점  $C_1 \sim C_n$ 을 회전 시켜서 자세각 정보가 반영된  $C_1' \sim C_n'$ 점의 위치를 찾아낸다. 자세각을 적용한 모델의 모습은 그림 4와 같다.

자세각이 적용된 각 점의 위치  $C_1' \sim C_n'$ 는  $pos$ 정보가 합산되지 않은 값으로 중심이 원점에 위치한다. 중심을  $pos$ 로 좌표변환 시켜주면 각 점들이 실제 돌맹이가 위

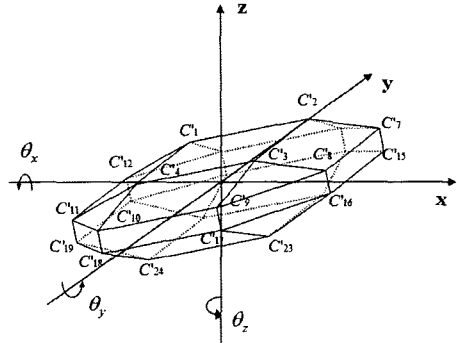
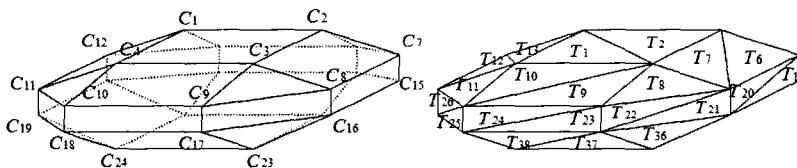


그림 4 자세각이 적용된 돌맹이 모델

치한 곳의 정보를 가진다. 돌맹이 모델에서 각 축의 자세각으로 회전을 하고, 현재의 위치로 좌표변환 된 각 점으로 충돌 검사(Collision Test)를 수행하게 된다. 물수체비 운동에서 충돌 감지(Collision Detection)는 강체와 강체사이의 충돌 감지와는 다르다[8]. 두 강체의 충돌에서는 강체의 형태가 변하거나 부서지는 현상이 나타나지만, 강체와 유체의 충돌에서는 강체가 유체와 충돌에 의하여 변형되지 않는다. 또한 유체의 강체에 대한 항력을 계산하기 위해서는 강체와 유체의 접촉면이 있어야 하므로, 계산된 각 점의  $z$ 축 좌표가 수면과 접촉이 되어 있거나 아래에 있으면 충돌을 감지하고 충돌 반응을 한다.

3.2 충돌 반응(Collision Reaction)

충돌 반응은 본 논문의 핵심적인 내용이다. 왜냐하면 수면과 돌맹이의 충돌이 일어나면 수력학적 항력이 발생하고 항력의 크기에 따라 돌맹이가 수면위로 튀어 오르거나, 가라앉기 때문이다. 충돌 반응은 충돌시의 돌맹이 위치를 물 모델에 이벤트로 전달함으로써 물 표면에 충돌에 따른 반작용이 일어나도록 하고, 충돌 시에 발생하는 물로부터의 항력을 구하고 이를 돌맹이 모델에 작용시켜 주어 물 표면에 충돌에 따른 물수체비 운동이 일어난다. 물에 의한 항력을 받는 돌맹이 모델은 다수의 삼각형들로 이루어진 다면체이다. 삼각형들을 이루는 각



(a) 메쉬를 구성하는 각 점의 좌표값

(b) 모델을 구성하는 삼각메쉬

그림 3 돌맹이 모델

점의 위치는 돌맹이의 위치를 계산하는 과정에서 모두 구해졌다. 면을 구성하는 삼각형들 중에서 수면과 접촉하고 있는 것들의 면적과 삼각형 면에 수직 벡터(normal vector)를 구한다. 면적과 수직 벡터를 이용하여 각 삼각형 면에 작용하는 수직 형태 항력과 수평 성분의 마찰 항력을 계산한다[9]. 마지막으로 전체 돌맹이에 작용된 항력은 각 삼각형에 주어진 항력의 합과 같다. 삼각형은 3차원 공간상에서 세변으로 이루어진 면(surface)을 가지게 되는데, 이 면이 물과 접촉하기 위해서는 3개의 점 중 하나 이상의 점이 수면 아래에 위치한다. 접촉면을 구하기 위해서는 삼각형의 접촉 상태를 확인하여야 한다. 한 점이 수면 아래에 있는 경우와 두 점이 수면 아래에 있는 경우, 세 점이 수면 아래에 있는 경우를 각각 다른 방법으로 접촉 면적을 구하게 된다. 세변의 길이로 삼각형의 넓이  $S$ 를 구하는 헤론의 공식(Heron's formula)을 이용하여 삼각형의 세변의 길이가  $a, b, c$ 일 때, 다음과 같이 구한다.

$$s = (a + b + c) / 2 \tag{9}$$

$$S = \sqrt{s(s-a)(s-b)(s-c)} \tag{10}$$

세 점이 모두 수면 아래에 있는 그림 5(a)의 경우에는 삼각형의 전 면이 물과 접촉하고 있으므로 세변의 길이  $|C_1'' - C_2''|, |C_2'' - C_3''|, |C_3'' - C_1''|$ 을 이용하여 삼각형의 면적을 구한다. 두 점이 수면 아래에 있는 그림 5(b)의 경우는  $C_4'', C_5'', C_6''$ 를 세 점으로 하는 삼각형 면적을 구하고, 선분  $\overline{C_4''C_5''}$ 의  $z=0$ 인 점  $P_1$ 과 선분  $\overline{C_4''C_6''}$ 에서  $z=0$ 인 점  $P_2$ 를 구하여,  $C_4'', P_1, P_2$ 를 세 점으로 하는 삼각형 면적을 구한다. 이 삼각형의 면적을  $\Delta C_4''C_5''C_6''$ 의 면적에서 빼주면 그림 5(b)의 삼각형과 수면의 접촉 면적이 된다. 한 점만 수면 아래에 있는 그림 5(c)의 경우는  $\overline{C_7''C_8''}$ 의  $z=0$ 인 점  $P_3$ 과 선분  $\overline{C_8''C_9''}$ 에서  $z=0$ 인 점  $P_4$ 를 구하여, 수면과 접촉면인  $\Delta C_8''P_3P_4$ 의 면적을 구한다.

물과 삼각형의 충돌을 확인하고 면적을 구할 때에 수면 아래에 있는 점의 위치와 깊이 정보를 물 모델에 깊이 설정 함수에 파라미터(parameter)로 넘겨준다. O'Brien의 방법을 이용한 물 모델은 돌맹이가 접촉하고 있는 부분에서의 깊이를 표면 모델에 적용하여 주면 접촉한 부분의  $x, y$ 좌표에 따라 높이를 검사 하고 충돌에 따른 물의 깊이를 처리하여 동심원 모양의 물결을 생성한다[4]. 삼각형과 물이 접촉한 면적  $S_{im}$ 을 구하고 삼각형이 가진 속도 성분을 분석한다. 속도 성분을 삼각형의 수직 벡터 방향 성분과 면과 평행한 수평 성분의 속도로 나누어 준다. 삼각형의 세 점을 외적(cross method)을 이용하여 수직 벡터를 구한다. 그림 6의 (a)와 (b)는 수직 벡터와 속도 벡터의  $z$ 축에 대한 방향각  $\theta$ 와  $x$ 축에 대한 방향각  $\phi$ 를 나타내고, (c)는 두 벡터 사이의 사이각  $\zeta$ 를 나타낸 그림이다.

삼각형 면에 수직 벡터인  $\vec{N}$  방향의 속도 벡터의 크기  $|\vec{V}_n|$ 과 면에 평행하는 속도 벡터의 크기  $|\vec{V}_t|$ 를 구하여, 면에 수직으로 작용하는 수직 형태 항력과 수평으로 작용하는 마찰 항력을 구한다. 속도의 크기는 수식 (11)과 같다.

$$|\vec{V}| = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{11}$$

속력 벡터  $\vec{V}$ 와  $\vec{N}$ 의 사이각 $\zeta$ 을 이용하여,  $|\vec{V}_n|$ 과  $|\vec{V}_t|$ 를 구한다.  $\zeta$ 는 벡터의 내적 공식으로 계산한다. 내적의 공식은 아래의 수식 12와 수식 13의 두 가지 방법으로 계산한다.

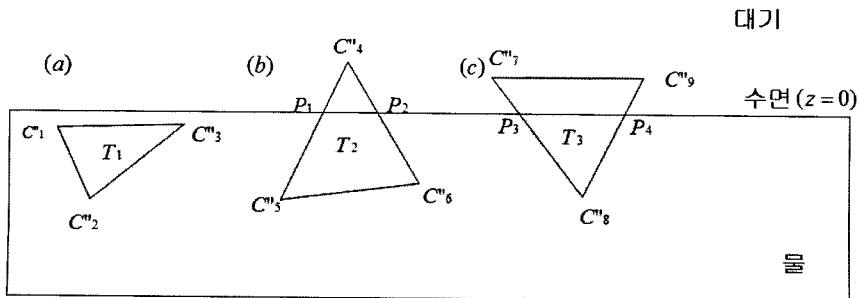
$$\vec{V} \cdot \vec{N} = |\vec{V}| |\vec{N}| \cos \zeta \tag{12}$$

$$\vec{V} \cdot \vec{N} = V_x N_x + V_y N_y + V_z N_z \tag{13}$$

수식 (13)에서 내적을 구하고  $|\vec{V}|, |\vec{N}|$ 를 구하여  $\cos \zeta$ 를 유도한다.

$$\cos \zeta = (\vec{V} \cdot \vec{N}) / (|\vec{V}| |\vec{N}|) \tag{14}$$

$$|\vec{V}_n| = |\vec{V}| \cos \zeta \tag{15}$$



(a) 세 점이 수면 아래에 있는 경우 (b) 두 점이 수면 아래에 있는 경우 (c) 한 점이 수면 아래에 있는 경우  
그림 5 삼각형과 물의 충돌

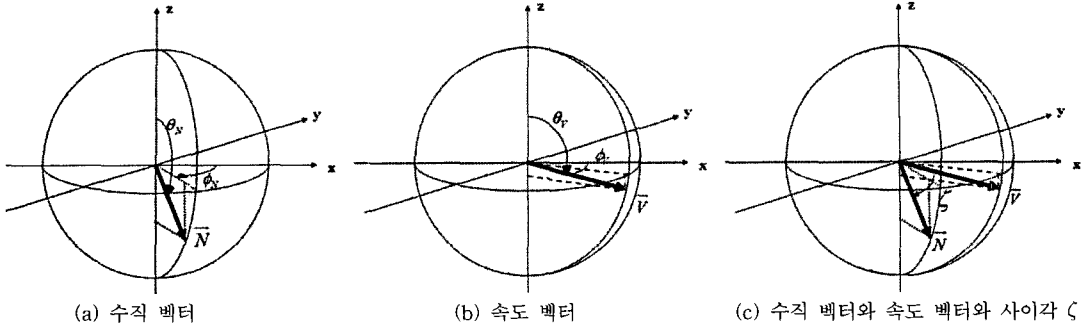


그림 6 삼각형의 수직 벡터와 돌맹이의 속도 벡터의 방향각 성분

면에 수평인 속도 성분  $\vec{V}_t$ 의 크기  $|\vec{V}_t|$ 를 수식 (15)에서 구한  $|\vec{V}_n|$ 과 수식 (11)에서 구한  $|\vec{V}|$ 로부터 피타고라스의 정리에 따라 계산한다.

$$|\vec{V}_t| = \sqrt{|\vec{V}|^2 - |\vec{V}_n|^2} \quad (16)$$

삼각형 면에 수직 성분 속도의 크기와 수평 성분 속도의 크기, 그리고 접촉 면적을 이용하여 수직 형태 항력과 마찰 항력을 구한다. 수식 (5)를 이용하여 본 논문에서 시뮬레이션의 핵심인  $\vec{n}$  방향으로 작용하는 수직 형태 항력  $F_n$ 과,  $\vec{t}$  방향으로 작용하는 대한 마찰 항력  $F_t$ 를 본 연구에서 유도하였다.

$$F_n = \frac{1}{2} C_{\rho_w} |\vec{V}_n|^2 S_{im} \quad (17)$$

$$F_t = \frac{1}{2} C_f \rho_w |\vec{V}_t|^2 S_{im} \quad (18)$$

수식 (17)과 (18)에서 계산된 항력  $F_n$ 과  $F_t$ 는 벡터 값이 아니므로, 돌맹이에 항력을 가해 주기 위해서는 방향 성분  $\vec{n}$ 과  $\vec{t}$ 가 필요하다.  $\vec{n}$ 는 삼각형의 수직 벡터  $\vec{N}$ 의 역 방향 단위 벡터이며,  $\vec{t}$ 는  $\vec{V}$ 가 삼각형 면에 투영된  $\vec{V}_t$ 의 역방향의 방향성을 가진 단위 벡터이다.  $\vec{n}$ 은 다음과 같이 계산된다.

$$\vec{n} = (-N_x, -N_y, -N_z) \quad (19)$$

$\vec{t}$ 는  $\vec{V}_t$ 를 이용하여 계산 되는데,  $\vec{V}_t$ 는

$$\vec{V}_t = \vec{V} - (|\vec{V}_n| \cdot \vec{N}) \quad (20)$$

이다.  $\vec{V}_t$ 를 이용하여  $\vec{t}$ 의 각 성분을 구한다.

$$t_x = -(V_x - N_x \cdot |\vec{V}_n|) / |\vec{V}_t| \quad (21)$$

$$t_y = -(V_y - N_y \cdot |\vec{V}_n|) / |\vec{V}_t| \quad (22)$$

$$t_z = -(V_z - N_z \cdot |\vec{V}_n|) / |\vec{V}_t| \quad (23)$$

수식 (11)에서 수식 (23)까지를 이용하여 삼각형 면에 대한 항력 벡터  $\vec{F}_D$ 를 구한다.

$$\vec{F}_D = F_n \cdot \vec{n} + F_t \cdot \vec{t} \quad (24)$$

수식 (3-21)에서 구한  $\vec{F}_D$ 는 아래의 그림 9와 같이 나타난다. 그림 7은 삼각형의 수직 벡터와 속도 벡터의 성분과 삼각형에 작용한 항력을 설명한다.

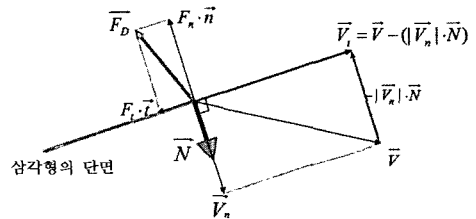


그림 7 삼각형의 속도 벡터와 항력 벡터

위 과정을 통해 단일 삼각형에 작용하는 항력을 벡터로 구했다. 돌맹이는 다면체이므로 물과 충돌이 일어나는 삼각형이 여러 개 있다. 충돌하는 여러 개의 삼각형에서 계산된 항력 벡터들의 합을 구함으로써 돌맹이 전체에 주어진 항력의 합을 구한다. 충돌이 일어나지 않는 삼각형의  $\vec{F}_D$ 는 (0,0,0)으로 계산한다. 항력 벡터의 합  $\vec{DragForce}$ 는 그림 3(b)의 각 삼각형( $T_0 \sim T_n$ )에 작용한 항력의 합으로 구하고, 그림 8과 같이 설명할 수 있다.

$$\vec{DragForce} = \sum_{x=T_1}^{T_n} \vec{F}_D \quad (25)$$

$$\vec{DragForce} = (DragForce_x, DragForce_y, DragForce_z) \quad (26)$$

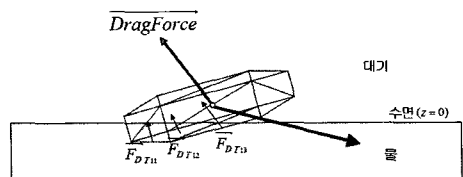


그림 8 돌맹이의 전체 항력

각 축에 대해 주어진 힘  $DragForce_x$ ,  $DragForce_y$ ,  $DragForce_z$ 는 뉴턴의 제2법칙  $F=ma$ 에 따라 각축에 대한 가속도

$$a_x = DragForce_x / M, \quad (27)$$

$$a_y = DragForce_y / M, \quad (28)$$

$$a_z = DragForce_z / M \quad (29)$$

로 유도된다. 수식 (27)~(29)의 가속도를 돌맹이의 운동에 적용한다. 충돌 시에는 돌맹이의 위치 정보를 계산하는 수식 (4)에서 수식 (8)를 아래의 수식으로 대체하여 계산한다.

$$V'_x = V_x + a_x \cdot elapsedtime \quad (30)$$

$$pos_x = pos_x + V_x \cdot elapsedtime + \frac{V'_x - V_x}{2} \cdot elapsedtime \quad (31)$$

$$V_x = V'_x \quad (32)$$

$$V'_y = V_y + a_y \cdot elapsedtime \quad (33)$$

$$pos_y = pos_y + V_y \cdot elapsedtime + \frac{V'_y - V_y}{2} \cdot elapsedtime \quad (34)$$

$$V_y = V'_y \quad (35)$$

$$V'_z = V_z - g \cdot elapsedtime + a_z \cdot elapsedtime \quad (36)$$

$$pos_z = pos_z + V_z \cdot elapsedtime + \frac{V'_z - V_z}{2} \cdot elapsedtime \quad (37)$$

$$V_z = V'_z \quad (38)$$

충돌에서 발생하는 항력을 계산하고 이를 돌맹이의 운동에 적용하여 일정한 속도와 자세각을 가진 돌맹이가 물위를 튀어 오르는 물수제비 운동이 가능하다. 아울러 물표면 모델에 돌맹이의 충돌에 해당하는 이벤트를 전달하여 줌으로써 충돌로 인한 동심원 형태의 물결을 만들어 주었다.

### 3.3 물 표면의 처리

돌맹이가 물의 표면과 충돌하여 튕겨 올라가거나, 아래로 가라앉으면 수면에는 동심원 모양의 파동이 발생한다. O'Brien은 부피 모델, 표면 모델과 스프레이 모델을 제안하였는데 본 논문에서는 부피 모델만을 수정하여 간단하게 물 표면을 구현하였다. 부피 모델은 물의 부피(volume)를 모델링 하며, 각 칼럼(column)들이 연결된 사각격자(rectilinear grid)의 모양을 취하고 있다 [4]. 각 칼럼은 독립적으로 생각하며, 칼럼 사이의 유체의 이동은 수직 수평으로 연결된 가상의 파이프(pipe)를 통해서 이루어진다. 그림 9는 부피 모델을 위에서 내려 본 그림으로 각 칼럼의 연결을 설명한다. 각 칼럼의 대각선 방향으로 연결된 컨트롤 포인트(control point)는

이웃 칼럼의 표면 상태를 확인하고 파이프 보다는 작은 량의 수량의 이동의 통로가 된다.

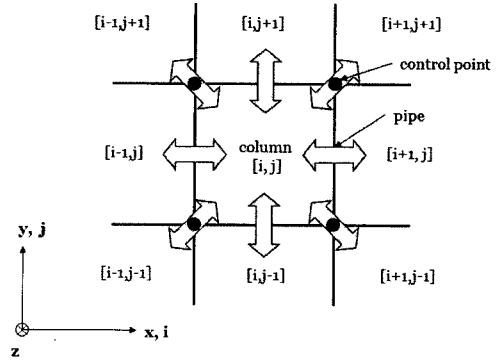


그림 9 각각의 수직칼럼과 8개의 방향으로 연결된 가상 파이프와 사각 격자로 나누어진 물 모델[4]

$[i,j]$ 칼럼의 높이  $h_{ij}$ 가 칼럼 사이에서 유체를 주고받는 기준 값이 된다. 각 칼럼은 파이프와 컨트롤 포인트를 이용하여 이웃에 위치한 각 칼럼들에서부터 유체를 받는다. 수식 (39)에 따라 전체 부피 모델은 질량 보존이 된다.

$$\forall ij : \forall kl \in \eta_{ij} : Q_{ij \rightarrow kl} = -Q_{kl \rightarrow ij} \quad (39)$$

$\eta_{ij}$ 는  $[i,j]$ 칼럼의 이웃에 위치한 칼럼들이며,  $Q_{ij \rightarrow kl}$ 는  $[i,j]$ 칼럼에서  $[k,l]$ 칼럼으로 흘러간 흐름의 크기로 비압축성 유체이기 때문에 파이프의 한쪽 끝의 흐름의 크기와 다른 쪽 끝의 흐름의 크기는 같고, 부호는 반대이다.

또한 유체의 물리적 성질을 만족하기 위하여 부피가 음수가 되는 것은 물리적 의미에서 맞지 않으므로, 모든 칼럼 안의 부피는 양수로 유지되어야 한다. 수식 (40)로 쓸 수 있다.

$$V_{ij}^{t+\Delta t} \geq 0 \Leftrightarrow V_{ij}^t \geq -\Delta V_{ij}^t \quad (40)$$

계산중에 칼럼의 부피가 음수가 될 수 있으며, 이러한 경우 위의 두 가지 조건을 통해서 격자 안의 모든 칼럼이 양의 부피를 가질 때까지 계산을 반복해야 한다.

실시간 처리를 위해서 수식 (40)을 모든 칼럼에서 양의 부피를 가질 때까지 여러 번 반복하지 않고, 수식 (41)과 같이 하여서 한번에 계산할 수 있도록 했다. 즉, 칼럼  $[i,j]$ 에서 부피의 변화량이 음수가 되는 경우 그때의 칼럼  $[i,j]$ 의 부피의 변화량을 '0'으로 설정함으로써 여러 번의 반복을 피하고, 칼럼의 부피가 0보다 크거나 같게 결정된다.

$$\text{if} (V_{ij}^t + \Delta V_{ij}^t < 0) \\ \{ V_{ij}^t + \Delta V_{ij}^t = 0 \} \quad (41)$$

따라서 수식 (41)로 여러 번의 반복계산을 피할 수

있으며, 한번 만에 안정적인 시뮬레이션 결과들을 얻을 수 있다. 위의 두 가지의 조건이 만족되어지면 칼럼 [i,j]에서 칼럼의 높이는 칼럼[i,j]의 볼륨의 변화율을 칼럼의 단위 면적으로 나누어 더함으로써 계산할 수 있다.

#### 4. 시뮬레이션 결과

물수제비 운동 모델은 OpenGL로 구현하여, centrino mobile 1.6Ghz CPU, 512MB 메모리 그리고 128MB의 그래픽카드 환경에서 실행하였다. 물표면의 텍스처 렌더링(texture rendering)을 포함하여 실시간에 20 frames/sec의 애니메이션이 가능하였다.

시뮬레이션 시에 돌맹이의 속도 벡터, 자세각 그리고 초기위치를 설정함으로써 다양한 돌맹이의 운동 궤적을 볼 수 있다. 애니메이션으로 나타나는 돌맹이의 운동 추이는 실제 물수제비 운동에 유사하다. 아래에 그림 10과 그림 11은 물수제비 운동 시뮬레이션을 제어하기 위한 시뮬레이터를 보여준다.

본 시뮬레이터를 통해 제공될 돌맹이의 조건을 사용자에게 의해 직접 제어할 수 있어 보다 사실적인 시뮬레이션 결과를 얻을 수 있다. 그림 12는 물수제비 시뮬레이션의 결과를 자세히 볼 수 있다.

그리고 아래의 그림 13은 시뮬레이션 결과를 측면에서 보았을 경우의 부분적 운동 궤적을 나타내고 있다.

#### 5. 결론 및 향후 연구 계획

실제 물수제비 운동은 돌맹이의 무게, 던지는 속도, 돌맹이 제공시의 자세각, 물과의 충돌시의 입사각과 의

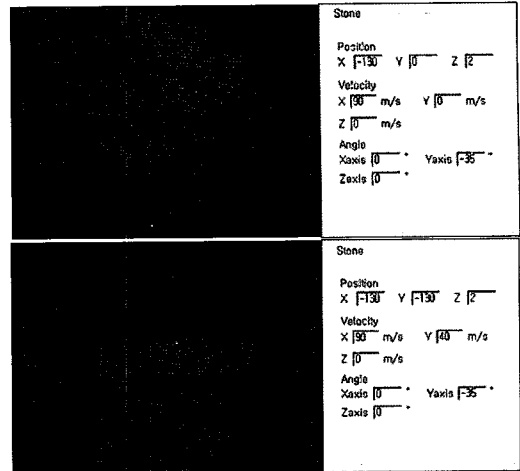


그림 11 물수제비 운동 모션

형에 따라 다양한 운동 추이를 보인다. 본 논문에서는 위의 사항들이 고려된 실시간에 수행 가능한 자연스러운 물수제비 운동 모델을 제시하였다. 이를 유체인 물과 강체인 돌맹이 사이의 상호작용으로 인한 수력학적 항력으로 재해석하고, 이를 기반으로 하여 자연 현상 속에서 물리 법칙의 지배를 받는 물리기반(Physically-based) 시뮬레이션을 구현하였다. 실제 돌맹이 모델의 충돌 반응을 계산하여 냈으며, 충돌 이후에 여러 차례 물위를 튀는 동작을 실시간에 재현했다.

향후 연구 과제로는 물수제비 운동에 돌맹이가 각속도를 가지고 z축에 대해서 회전하는 스핀(spin)을 처리

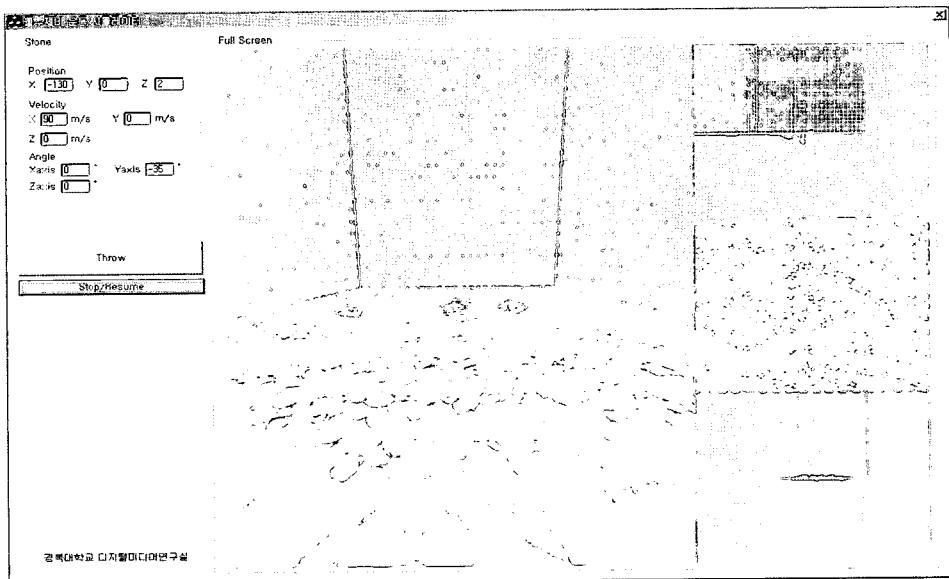


그림 10 물수제비 운동 시뮬레이터



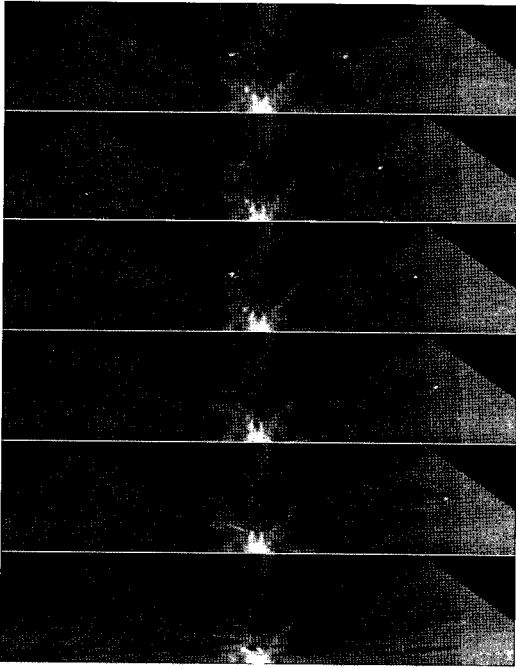


그림 12 물수제비 운동 시뮬레이션 결과

하는 것과 충돌 시 작용하는 항력을 토크(torque)로 처리하여 회전력에 따른 다양한 반응을 추가할 수 있을 것이며, 물 표면에 효과를 보다 세밀하게 표현할 수 있을 것이다. 이번 시뮬레이션에서는 돌맹이의 충돌 검사를 수면에 대해서 했지만, 충돌 검사 알고리즘을 개선하면 흐르는 물이나 일렁이는 물에서 물 표면의 상태에 따른 반작용의 차이를 고려하여 줄 수 있을 것이다. 뿐만 아니라 돌맹이가 체공비행(endurance flight)하는 시간 동안에 바람의 영향이나 스핀으로 인한 자이로스코프 효과(gyroscope effect)를 구현하여 공기역학에 의한 항력까지 적용 할 수 있다. 향후 연구 과제들의 접목으

로 물수제비 운동은 공기역학, 유체역학, 동역학 등 다양한 물리현상이 적용되는 운동 모델이 된다. 이를 이용하여 수면을 향해 발사된 탄환의 굴절 궤적을 구할 수 있으며, 수상스키와 같은 다양한 시뮬레이션을 할 수도 있다. 또한 유체역학을 위한 물리 엔진 개발에 이용하여 유체에서 일어나는 다양한 자연 현상에 대한 시뮬레이션에 활용할 수 있다.

### 참 고 문 헌

- [1] P. M. Hubbard, "Collision Detection for Interactive Graphics," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 1, No. 3, pp. 218-230, 1995.
- [2] A. Wiktkin and D. Baraff, "Physically based modeling," *Proceedings of SIGGRAPH '97 Course Notes*, pp. D32-D34, 1997.
- [3] L. Bocquet, "The physics of stone skipping," *American Journal of Physics*, Vol. 71, Issue 2, pp. 150-155, 2003.
- [4] J. F. O'Brien and J. K. Hodgins, "Dynamic Simulation of Splashing Fluids," *Computer Animation '95*, pp. 198-205, 1995.
- [5] L. N. Long and H. Weiss, "The Velocity Dependence of Aerodynamic Drag: A Primer for Mathematicians," *American Mathematical Monthly*, Vol. 106, No. 2, pp. 127-135, Feb., 1999.
- [6] T. C. Papanastasiou, *Applied fluid mechanics*, PTR Prentice Hall, 1994.
- [7] E. Angel, *Interactive Computer Graphics*, Addison-Wesley, 2000.
- [8] B. M. Das, A. Kassimali and S. Smi, *동역학*, 보문당, 2000.
- [9] J. Dingliana and C. O'Sullivan, "Graceful Degradation of Collision Handling in Physically Based Animation," *EUROGRAPHICS*, Vol. 19, No. 3, 2000.

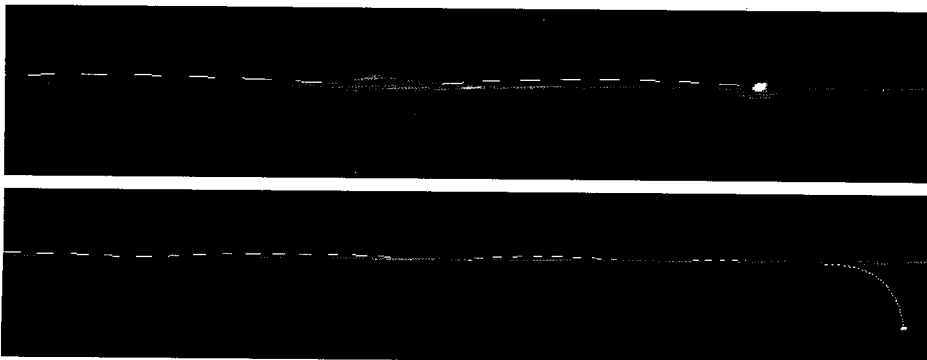


그림 13 측면에서 본 물수제비 운동의 부분 궤적(상)과 전체 궤적(하)



#### 도 주 영

1999년 경성대학교 전산통계학과(전산학, 이학사). 2001년 경성대학교 컴퓨터학과(전산학, 석사). 2001년~현재 경북대학교 컴퓨터공학과(공학박사 수료). 관심분야는 컴퓨터 그래픽스, 멀티미디어, 물리기반 렌더링 등



#### 라 은 철

2002년 경북대학교 컴퓨터공학과(전산공학, 공학사). 2004년 경북대학교 컴퓨터공학과(전산공학, 석사). 2004년~현재 디토소프트 기술연구소 팀장. 관심분야는 컴퓨터 그래픽스, 멀티미디어, 모바일 프로그래밍 등



#### 김 은 주

1984년 경북대학교 전자공학과(전산공학, 공학사). 1986년 경북대학교 전자공학과(전산공학, 공학석사). 2003년 경북대학교 컴퓨터공학과(공학박사). 2000년~현재 동명정보대학교 공과대학 정보통신공학과 전임강사. 관심분야는 멀티패러다임 프로그래밍, 병렬 알고리즘, 컴퓨터 그래픽스 등



#### 유 관 우

1980년 국립경북대학교(공학사 - 전자공학). 1982년 한국과학기술원(공학석사 - 전산공학). 1990년 메릴랜드대학교(공학박사 - 전산공학). 1991년~현재 경북대학교 컴퓨터공학과 교수. 관심분야는 멀티패러다임 프로그래밍, 병렬 알고리즘, 컴퓨터 그래픽스 등