An Interactive Virtual Stone Skipping System

Namkyung Lee
K-MAC Inc.
Daejon 305-500, Korea
namklee@hotmail.com

Nakhoon Baek *†
Kyungpook Nat'l Univ.
Daegu 702-701, Korea
oceancru@gmail.com
(corresponding author)

Abstract

Stone skipping is an example of rigid-fluid coupling, which typically need much computation. In this paper, we present a real-time method for visually plausible stone skipping simulation. Based on the Newtonian physics, an intuitive dynamics model is presented to simulate the linear and rotational motions of the stone. We also present a modified water surface model for the waves due to the bounces of the stone. For more realistic user experiences, the real-world stone is substituted by the Wii Remote connected to the host PC. User just throws the controller (strapped to his/her wrist), and our implementation shows the real-time simulation of the bouncing process of the virtual stone.

1 Introduction

Stone skipping (also known as *ducks and drakes*) is one of the traditional pastimes, experiencing typical rigid-fluid coupling phenomena. We get various results with respect to the initial conditions such as attacking angle, velocity, etc. Although it is possible to physically simulate these phenomena, we need a remarkable amount of computation to make models for interactions between the fluid and the rigid body.

In this paper, we present a real-time virtual stone-skipping simulation system, as shown in Figure 1. Our system contains a computation-efficient dynamics model to show visually plausible stone skipping motions in real time. The water waves generated by the stone on the water surface are processed by a modified water surface model from [10]. For better user experiences, the initial physical quantities are naturally set by the *Wii Remote*, the three dimensional input device for Nintendo game console *Wii*. This wireless controller has the acceleration, location and infrared sensors and a Bluetooth communication module to be directly connected to the host PC. We simulate the virtual stone skipping with this interactive device.

* Corresponding author: oceancru@gmail.com

† This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (Grant 2011-0014886).

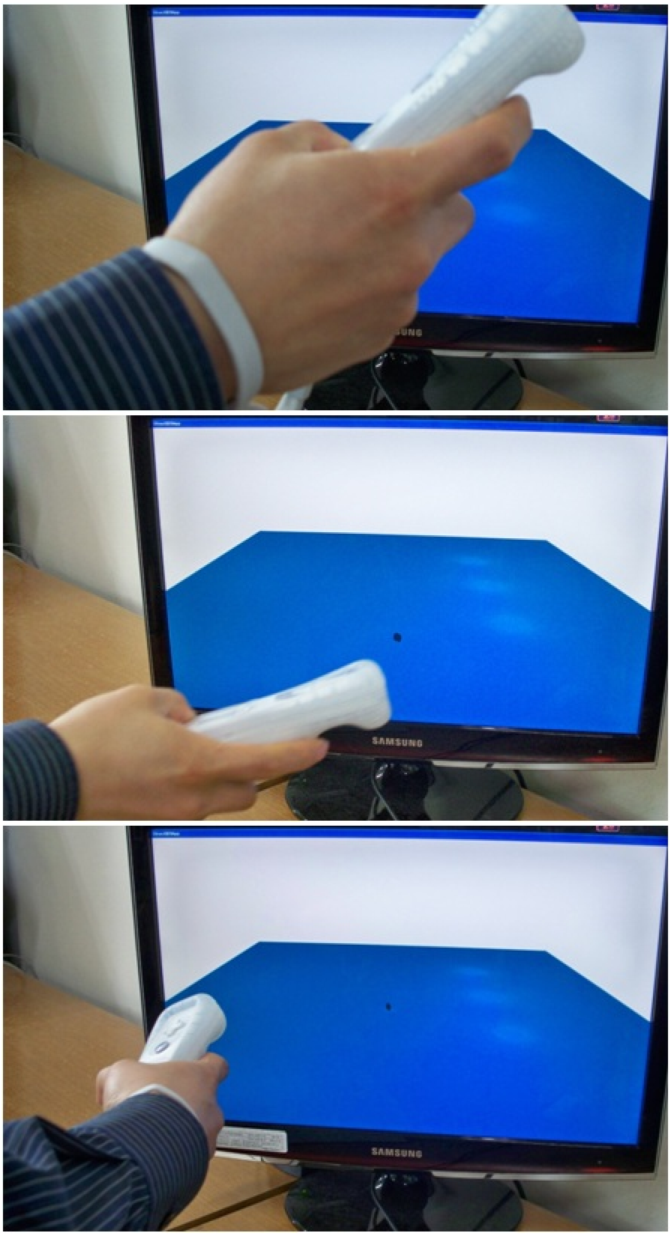


Figure 1. Our virtual stone skipping system.

2 Background

Bocquet's article[4] would be the first literature on the stone skipping. He analyzed the stone skipping as the drag forces between the water surface and the stone. He physically analyzed the suitable angle of attack for the greater number of stone bounces. Later, Rosellini et al. presented a more physical analysis of the stone skipping[14]. Nagahiro et al. used ordinary differential equations (ODE) and the smoothed particle hydrodynamics (SPH) method, to calculate the best velocity and angle of attack[12]. Both require any amount of computation.

In the field of computer graphics, typical physically-based modeling researches are focusing on the realistic representation of the water surface hit by a rigid body. Since Stam's fluid simulation method[16], many research results are available[2, 6, 17, 3]. However, most of them require much computation time, and still unsuitable for real-time applications including computer games and virtual reality programs.

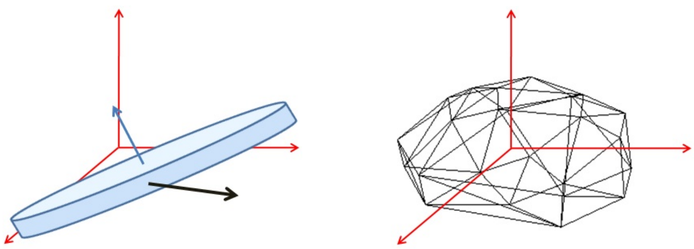
Do et al.[7] focused on the virtual simulation of stone-skipping, and calculated the vertical and horizontal drag forces for the simulation. However, they ignored the spinning of the stone and its related effects, and failed to show realistic simulations, at least for some cases.

For our user experience support, we used the *Wii Remote* as the input device. As an inexpensive three-dimensional input device, *Wii Remote* is now available for various applications including interactive whiteboard[9], motion recognition[15], virtual orchestra[5], etc. Our implementation will be another good example of direct interactions with the virtual physical world.

3 Our Method

3.1 A real-time dynamics model for stone-skipping

A well-balanced practical dynamics model is needed to achieve a realistic simulation of the stone skipping with relatively less computation. A stone thrown to the water surface is affected by four kinds of forces: throwing force, gravity, air resistance and impulsive force to the water. Using a typical physically-based simulation system[1], the throwing force and gravity can be naturally handled just as the forces on the rigid bodies.



(a) a flat disk model (b) a triangular mesh model

Figure 2. Modeling a virtual stone.

Bouncing of a stone is actually generated by the reaction force to the water surface. Bocquet modeled the flying stone simply as a flat disk, as shown in Figure 2.(a) and presents an analytic model for the collision of this disk with the surface of still water[4]. Later, Do et al.[7] refines the virtual stone with a triangular mesh, as shown in Figure 2.(b). In our simulation, we adopted a hybrid approach of using the triangular mesh model for the more precise reactions to the water surface and the simplified disk model for air resistance calculation, respectively.

At the moment of collision, its linear and angular velocity and the area of contact affects the bouncing of the stone. The linear velocity and the amount of contacting area are important factors for calculating the lift force. When the lift force is greater than the gravity, the stone moves the water surface up. The spinning of the stone supports its stabilization and also makes the curved trajectories of the stone skipping. Thus, all these terms should be integrated into the dynamics model.

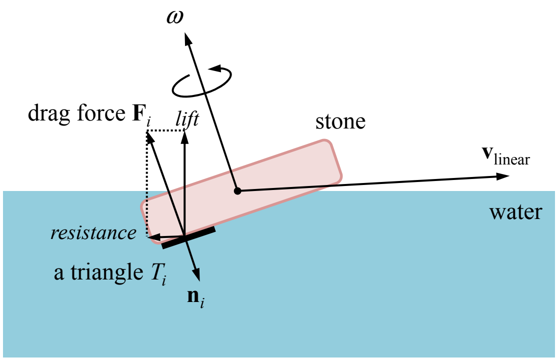


Figure 3. Collision between the water surface and a triangle.

Figure 3 shows the collision between the water surface and a triangle T_i from the virtual stone mesh. The *drag force* $\mathbf{F}_i$ is derived from the contacting area S_i. Thus, only the triangles under the water surface generate the drag force, major source of the bounce. Actually, the drag force can be decomposed into the lift force and the resistance, as shown in Figure 3. Since the stone spins, the linear velocity at the triangle T_i can be calculated as:

$$\mathbf{v}_i = \mathbf{v}_{\text{linear}} + \mathbf{r}_i \times \omega, \tag{1}$$

where $\mathbf{v}_{\text{linear}}$ is the linear velocity of the stone, $\mathbf{r}_i$ is the average rotational radius for the triangle and ω is the rotational velocity of the stone. Based on the Newton's drag force equation, the drag force $\mathbf{F}_i$ is calculated as[8, 11]:

$$\mathbf{F}_i = -\rho_{\text{water}} (\mathbf{v}_i \cdot \mathbf{n}_i)^2 S_i \ \mathbf{n}_i, \tag{2}$$

where ρ_{water}, $\mathbf{n}_i$, and S_i are the material constant for water, the normal vector, and contacting area of the triangle T_i, respectively. With this drag force, the torque τ_i on the triangle T_i can be expressed as the cross product of:

$$\tau_i = \mathbf{F}_i \times \mathbf{r}_i. \tag{3}$$

Integrating all the drag forces and torques on triangles, we get the total force and torque on the stone:

$$\mathbf{F}_{\text{drag}} = \sum_i \mathbf{F}_i, \tag{4}$$

and

$$\tau_{\text{drag}} = \sum_i \tau_i. \tag{5}$$

We also calculate the drag forces and lift forces with respect to the air. To minimize the computational burdens, we approximated the stone as a thin circular disk, as Bocquet did[4]. The Newtonian equations as shown in Equations (2) and (3) are also used for this calculation, with the air factors rather than the water surface. Through approximating the stone as a disk, we get more simplified equations of:

$$\mathbf{F}_{\text{air}} = -\rho_{\text{air}} (\mathbf{v}_{\text{disk}} \cdot \mathbf{n}_{\text{disk}})^2 S_{\text{disk}} \ \mathbf{n}_{\text{disk}}, \tag{6}$$

and

$$\tau_{\text{air}} = \mathbf{F}_{\text{air}} \times \mathbf{r}_{\text{disk}}, \tag{7}$$

for the whole stone. Here, $\mathbf{v}_{\text{disk}}$, $\mathbf{n}_{\text{disk}}$, S_{disk}, and $\mathbf{r}_{\text{disk}}$ are the linear velocity, normal vector, total surface area, and average radius of the simplified disk, respectively.

The gravity and air-related forces are consistently applied to the stone, over the whole simulation process. The throwing force should be applied only at the very first throwing time. Impact forces due to Equation (2) are applied to the stone at the stone-water collision time.

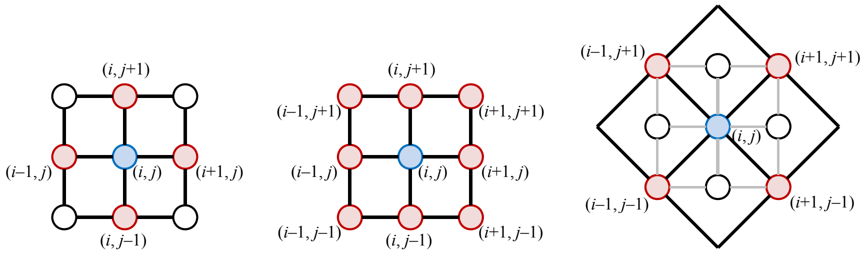
3.2 A water surface model

We also need to model the water surface and simulate the waves caused by the stone skipping. Although we already have a few photo-realistic water surface models[18], they require much computation. Our goal is to achieve the real-time processing on ordinary PC's with mid-tier graphics cards, and thus, we used an enhanced version of a simple height field model[10].

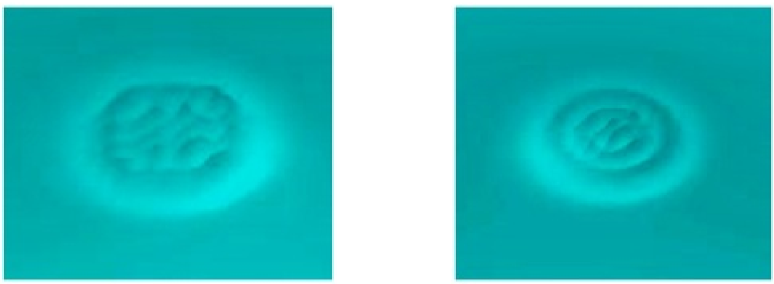
Letting the water surface be the xy-plane, its height changes are assigned to the z-axis. The displacement terms can be expressed as the following second-order differential equation with respect to the time:

$$\frac{\partial^2 z}{\partial t^2} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - \mu \frac{\partial z}{\partial t}, \tag{8}$$

where c is a constant for the propagation speed, μ is the viscosity coefficient, and $-\mu(\partial z / \partial t)$ is the damping force, respectively.



(a) 4 neighbors (b) 8 neighbors (c) diagonal neighbors



(c) simulated with 4 neighbors (d) simulated with 8 neighbors

Figure 4. Wave simulation models for the stone bounce.

Equation (8) is actually a kind of damped wave equation. Although we can get an analytic solution for Equation (8), it will require too much computation to achieve real-time performance. Alternatively, we use a numerical approximation of the wave propagation on the height field. Discretizing with respect to time t,

$$\frac{z_{i,j}^{t+1} - 2z_{i,j}^{t} + z_{i,j}^{t-1}}{(\Delta t)^2} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) - \mu \frac{z_{i,j}^{t+1} - z_{i,j}^{t}}{2\Delta t}, \tag{9}$$

where $z_{i,j}^t$ is the height at the grid point (i, j) at time t. Then, the second-order differential term $(\partial^2 z/\partial x^2) + (\partial^2 z/\partial y^2)$ is discretized.

In the typical central difference approximation, four neighbors are engaged as shown in Figure 4.(a). Thus, $(\partial^2 z/\partial x^2) + (\partial^2 z/\partial y^2)$ is approximated as:

$$\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = \frac{1}{d^2} \left[z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} - 4z_{i,j} \right],$$

where d is the distance of those square grid points. Time parameter t used as a superscript is omitted for simplicity.

In contrast, we integrated eight neighbors as shown in Figure 4.(b), and thus we added extra terms for the diagonals as shown in Figure 4.(c). Now, in spite of a bit more computation, $(\partial^2 z/\partial x^2) + (\partial^2 z/\partial y^2)$ is approximated as:

$$\begin{aligned} \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} &= \frac{1}{d^2} \left[z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} - 4z_{i,j} \right] \\ &+ \frac{1}{(\sqrt{2}d)^2} \left[z_{i-1,j-1} + z_{i-1,j+1} + z_{i+1,j-1} + z_{i+1,j+1} - 4z_{i,j} \right]. \end{aligned} \tag{10}$$

Combining Equations (9) and (10),

$$\begin{aligned} \frac{z_{i,j}^{t+1} - 2z_{i,j}^{t} + z_{i,j}^{t-1}}{(\Delta t)^2} &= c^2 \Bigg(\frac{1}{d^2} \left[z_{i-1,j} + z_{i+1,j} + z_{i,j-1} + z_{i,j+1} - 4z_{i,j} \right] \\ &+ \frac{1}{(\sqrt{2}d)^2} \left[z_{i-1,j-1} + z_{i-1,j+1} + z_{i+1,j-1} + z_{i+1,j+1} - 4z_{i,j} \right]. \Bigg) \\ &- \mu \frac{z_{i,j}^{t+1} - z_{i,j}^{t}}{2\Delta t}. \end{aligned}$$

We get the final difference equation as follows:

$$\begin{aligned} z_{i,j}^{t+1} &= \left(\frac{4}{\mu \Delta t + 2} - \frac{10}{3}\xi \right) z_{i,j}^{t} + \frac{\mu \Delta t - 2}{\mu \Delta t + 2} z_{i,j}^{t-1} \\ &+ \frac{1}{2}\xi \left[z_{i+1,j}^{t} + z_{i-1,j}^{t} + z_{i,j+1}^{t} + z_{i,j-1}^{t} \right] \\ &+ \frac{1}{4}\xi \left[z_{i+1,j+1}^{t} + z_{i-1,j-1}^{t} + z_{i-1,j+1}^{t} + z_{i+1,j-1}^{t} \right], \end{aligned} \tag{11}$$

where ξ is used as the substitution for $(2c^2(\Delta t)^2/d^2)/(\mu \Delta t + 2)$.

In the case of relatively sparse grids, the difference equation with four neighbors may show some irregular shapes, as shown in Figure 4.(c). Using Equation (11), more circular shapes are generated, as shown in Figure 4.(d). Although the eight neighbor difference equation needs a little more computation, the total amount of computation is reduced through using sparser grid. Notice that from the view point of the stone, the actual grid is easy to be sparse even with high grid resolutions, since the stone skips over the expanse of water.

3.3 User interface

Wii remote, also known as *Wiimote*, is the three-dimensional input device for Nintendo's game console *Wii*[13]. A *Wii remote* has acceleration and motion sensors for three directions and ultra-red sensors, as shown in Figure 5.

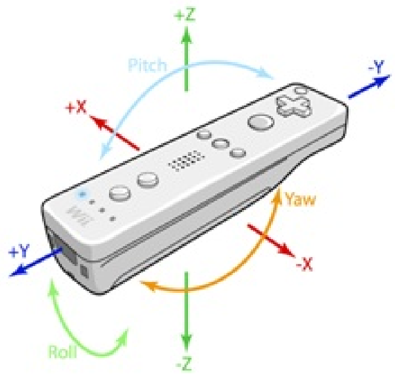


Figure 5. The *Wii remote* controller.

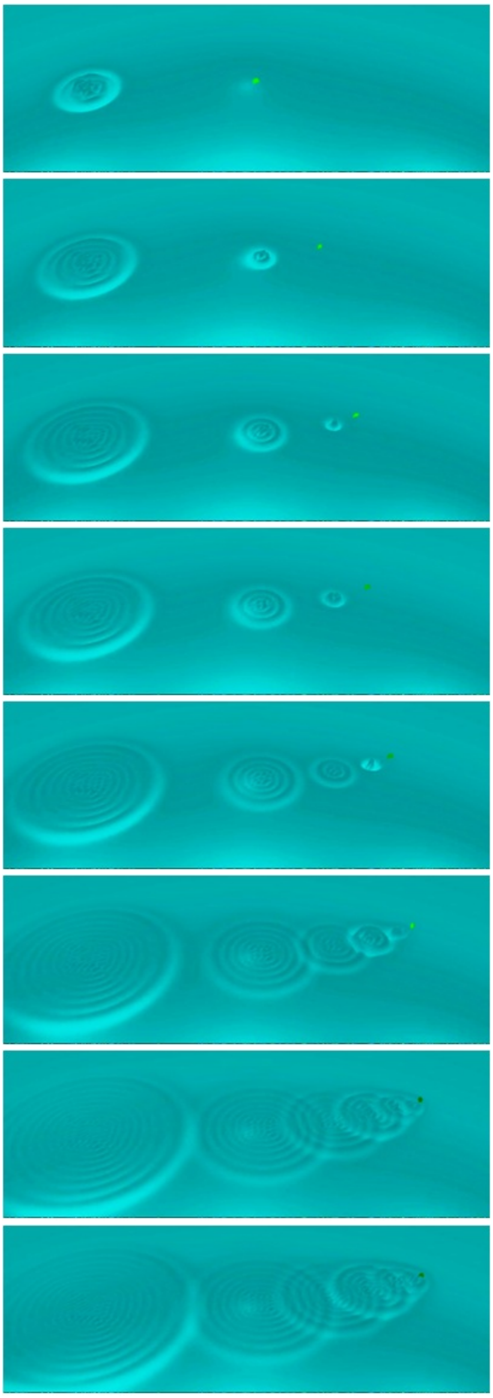
Our system interprets the sensor values of the *Wii remote* as the physical quantities of the virtual stone, during the stone throwing process. The location and acceleration values of the *Wii remote* are used for those of the virtual stone. At the first stage, the user takes the throwing motion with the *action button* pressed. At the time of throwing it, the action button will be released and the location, orientation, linear and angular acceleration terms are sent to the simulation system as the initial physical quantities. Through tracing these physical quantities of the virtual stone, we can calculate the velocity and the attack angle with respect to the water surface, and generates the bouncing motion. A rubber strap attaches the *Wii remote* to the user's wrist, to prevent real collisions. This kind of user interface based on the actual throwing of the *Wii remote* finally achieved more natural user experience.

4 Example System

Our virtual stone-skipping system is implemented on an Intel Core2 6300 1.86GHz PC with a GeForce 9600 graphics card and DirectX 9.0c libraries. For user interactions, a Nintendo *Wii remote* wireless controller (ADXL 330) is also used. This controller uses the Bluetooth protocol and reads out its data to the host PC with a maximum of 100Hz update rate. To handle the communications with the *Wii remote*, we use the *WiiYourself!* library[19].

A virtual stone represented by a mesh of 44 triangles was used for the experiment, with a height field over the 500×500 rectangular grid. Our system shows more than 90 frames per second, including all the simulations and real-time renderings with the light sources and textures. Figures 6 and 7 are simulation results from our system. Figure 6 shows a sequence of snapshots for a typical stone-skipping simulation.

Figure 7 demonstrates the effect of the spinning of the stone. Without any rotation on the stone, it bounces just along to the moving direction, as shown in Figure 7.(b) and also

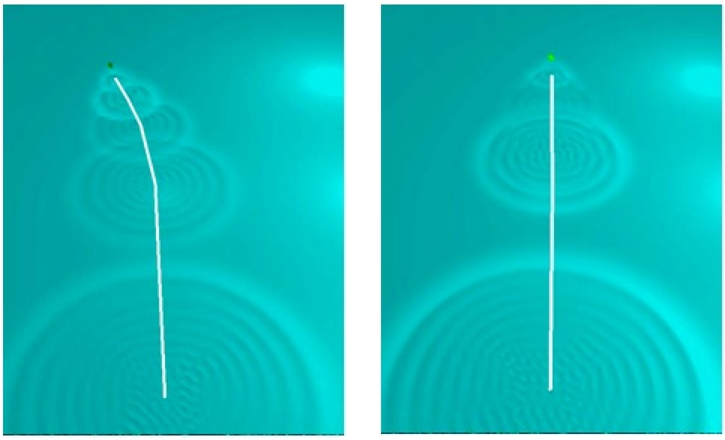


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)

Figure 6. A sequence of stone-skipping simulation.

mentioned in the previous work[7]. With the same linear velocity, our system integrates the rotational motion of the stone, and it generates naturally curved trajectory of the stone skipping, as shown in Figure 7.(a). As shown in Figure 7, the spinning stone makes much more bounces even with the same linear velocity.

Figure 8 shows the number of bounces with respect to the angular velocity. As shown



(a) with spinning
(from our system)

(b) without spinning
(from the previous work[7])

Figure 7. The curved trajectory of a spinning stone.

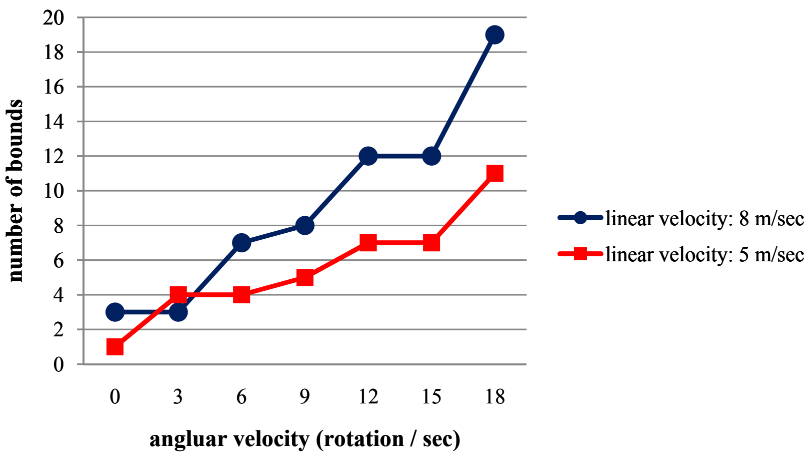


Figure 8. The number of bounces with respect to the spinning speed.

in this comparison, the spinning of the stone is one of the most important factors for the stone skipping. Figure 9 shows the momentum changes when thrown at the speed of 8 m/sec. Without spinning, the stone bounced only three times, and its momentum rapidly decreased for each bounce. In contrast, for the spinning stone, the momentum gradually decreases, and makes more bounces.

5 Conclusions and Future Work

In this paper, we presented a real-time virtual experience system for stone-skipping. To reproduce the stone-skipping, we derived a restrained dynamics model for flying stones and also a wave propagation model on the water surface. Based on these specialized physically-based modeling techniques, we accomplished visually plausible interactive simulations at more than 90 frames per second.

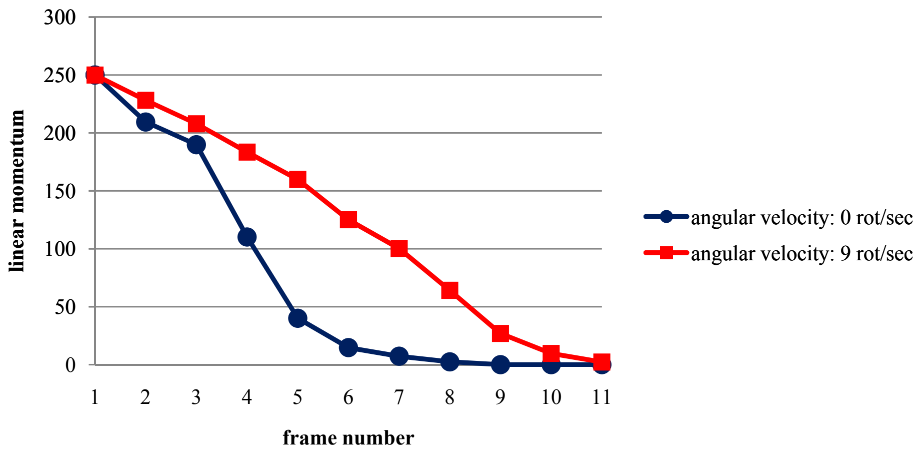


Figure 9. Momentum change with and without spinning.

For better user experiences, we established a completely perceptible interface with *Wii remote*. Extracting all required physical quantities from the user motions, we accomplished more immersed experience. Our relatively *inexpensive implementation of a perceptible simulation system* would be expected to be used for other application areas. Currently, we are working on the better water surface model and user experience, to achieve more realistic systems.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(Grant 2011-0014886).

References

[1] David Baraff. Physically based modeling: Rigid body simulation. SIGGRAPH Course Note, 2001.

[2] Christopher Batty, Florence Bertails, and Robert Bridson. A fast variational framework for accurate solid-fluid coupling. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 100, New York, NY, USA, 2007. ACM.

[3] Markus Becker, Hendrik Tessendorf, and Matthias Teschner. Direct forcing for lagrangian rigid-fluid coupling. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):493–503, 2009.

[4] Lyderic Bocquet. The physics of stone skipping. *American Journal of Physics*, 71(2):150–155, 2003.

[5] Bernd Bruegge, Christoph Teschner, Peter Lachenmaier, Eva Fenzl, Dominik Schmidt, and Simon Bierbaum. Pinocchio: conducting a virtual symphony orchestra. In *ACE '07: Proceedings of the international conference on Advances in computer entertainment technology*, pages 294–295, New York, NY, USA, 2007. ACM.

[6] Mark Carlson, Peter J. Mucha, and Greg Turk. Rigid fluid: animating the interplay between rigid bodies and fluid. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 377–384, New York, NY, USA, 2004. ACM.

[7] Jooyoung Do, Namkyung Lee, and Kwan Woo Ryu. Realtime simulation of stone skipping. *International Journal of Computers*, 4(1):251–254, 2007.

[8] D. Halliday and R. Resnick. *Fundamentals of Physics*. John Wiley & Sons, 2005.

[9] Johnny Chung Lee. Hacking the nintendo wii remote. *IEEE Pervasive Computing*, 7(3):39–45, 2008.

[10] Eric Lengyel. *Mathematics for 3D Game Programming and Computer Graphics, Second Edition*. Charles River Media, Inc., Rockland, MA, USA, 2003.

[11] Lyle N. Long and Howard Weiss. The velocity dependence of aerodynamic drag: A primer for mathematicians. *The American Mathematical Monthly*, 106(2):127–135, 1999.

[12] Shin-ichiro Nagahiro and Yoshinori Hayakawa. Theoretical and numerical approach to "magic angle" of stone skipping. *Phys. Rev. Lett.*, 94(17):174501, May 2005.

[13] Nintendo Wii. http://www.nintendo.com/wii, 2010.

[14] Lionel Rosellini, Fabien Hersen, Christophe Clanet, and Lyderic Bocquet. skipping stones. *Journal of Fluid Mechanics*, 543:137–146, 2005.

[15] Thomas Schlömer, Benjamin Poppinga, Niels Henze, and Susanne Boll. Gesture recognition with a wii controller. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 11–14, New York, NY, USA, 2008. ACM.

[16] Jos Stam. Stable fluids. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[17] Tsunemi Takahashi, Heihachi Ueki, Atsushi Kunimatsu, and Hiroko Fujii. The simulation of fluid-rigid body interaction. In *SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications*, pages 266–266, New York, NY, USA, 2002. ACM.

[18] Natalya Tatarchuk. Artist-directable real-time rain rendering in city environments. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pages 23–64, New York, NY, USA, 2006. ACM.

[19] WiiYourself! Project. http://wiiyourself.gl.tter.org/, 2010.