Visual Simulation of Waterfalls and Other Water Phenomena

Accurate physically-based rendering of complex phenomena such as the flow of water is an unrealistic aim: computational fluid dynamics simulation is a compute-intensive task, even for simple flow regimes, and is far beyond current knowledge for complex flows such as waterfalls. Our goal here is to develop a simple yet flexible simulation system which models a rich variety of scenes and behaviours and provides the basis for visually convincing rendering. The simulation system aims to provide a convenient set of facilities for an animator to construct a scene and control the flow of water in that scene. Whereas other authors [1,2,3] have simulated flows such as rivers, we have concentrated on falling water. Our system draws on observations of a large number of real waterfalls.

A standard collection of objects such as spheres, planes and polygons are used to model terrain or other solid objects such as fountains. More elaborate terrain models as in [4,6] are intended but the current focus is on behaviour. Falling water is modelled using a three-dimensional particle system [5]: sections of falling water consist of a collection of relatively small particles which obey gravity. Each particle contains data relating to position, velocity and size (including mass). Drops may influence one another, depending on the outcome of a simple particle-particle intersection test. As two particles approach, their relative velocity is examined: if this is above a certain threshold, the particles bounce away from one another, but if they are moving slowly they coalesce into a single blob of water (Figure 1 shows a fountain with coalescing columns). Intersection tests between particles or particles and solid objects are evaluated efficiently because the velocity of the particles and the simulation time step enable a localisation optimisation to be employed. Where a particle collides with a solid object, reflection about the surface normal takes place, but the normal is perturbed by a random amount to simulate surface irregularities and roughness. The reflection velocity is dampened by a factor based on the incident angle: if the angle is near 0°, then minimum loss occurs, but if the angle is near 90° then almost all energy is lost in the collision, thus slowing down the particle. If the impact velocity exceeds a given threshold, the particle splits into two equally sized particles. Figure 2 is a typical example.

Whilst falling, water is influenced by a number of factors. It can slow down and spread out due to air resistance and have its position affected by wind, sometimes dramatically. In our system, when wind is encountered, the number of nearby particles in the plane of interest is counted. This value is used to dampen the effect of wind on the volume of water falling, on the principle that dense particles mask the effects of wind whereas sparse particles are more susceptible to being blown around. Multiple winds from multiple directions are used.

Mist/water vapour surrounds any section of falling water, contributed to by collisions with objects, deceleration, splitting, and small droplets of vapour escaping from the main volume. For example, a particle moving above a given velocity emits much smaller mist particles. Figure 3 demonstrates this effect. The behaviour of mist differs from water particles in that it is very light, thus not falling at the same rate as a normal section of water due to air resistance. Mist is particularly susceptible to influence both from the main body of water and external effects such as wind and may rise as well as fall. Mist also expands, contracts, and is even less predictable than its liquid counterpart.

Rendering the final image in our simulation is currently straightforward as our goal is *not* realistic rendering but visualisation of the dynamics of the simulation. Each particle is initially assigned a blue colour and a vector is drawn at the end of each time step based on the distance travelled in that time step, i.e. we draw velocity vectors. On striking an object the colour of the particle is changed. The sound which water makes is emulated by generating random noise when particles strike solid objects .

In our future work we plan to explore the turbulent actions of the splashdown area of a waterfall. We will also apply the ideas developed in the study of falling water to other water phenomena such as rivers and puddles. The goal is to provide an animator system which allows the construction of water-based scenes and effects based on the ideas which evolve during this research.

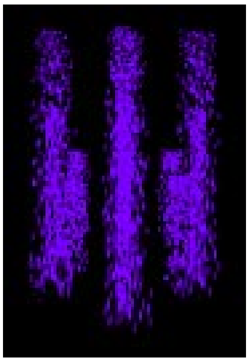

Figure 1 Fountain with Coalescing Columns (still)

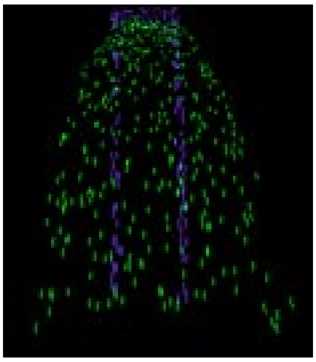

Figure 2 Water Striking Obstacle (still)

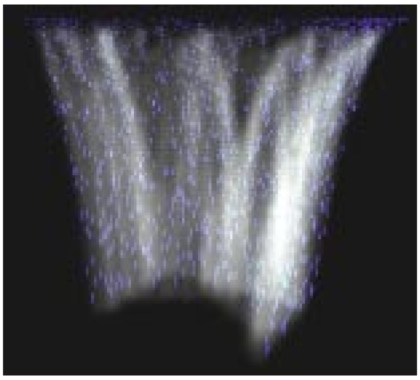

Figure 3 Horseshoe Falls with Mist (still from video)

References

1. N. Chiba, S. Sanakanishi, K. Yokoyama, I. Ootawara, K. Muraoka & N. Saito: Visual Simulation of Water Currents Using a Particle-based Behavioural Model. *Journal of Visualization and Computer Animation, Volume 6, Issue 3, July-September 1995, pages 155-171.*

2. N. Foster & D. Metaxas: Realistic Animation of Liquids. *Graphical Models and Image Processing, Volume 58, Number 5, September 1996, pages 471-483.*

3. M. Kass & G.S.P. Miller: Rapid, Stable Fluid Dynamics for Computer Graphics. *ACM SIGGRAPH Computer Graphics, Volume 24, Number 4, August 1990, pages 49-57.*

4. H. Mallinder: The Modelling of Large Waterfalls using String Texture. *Journal of Visualization and Computer Animation, Volume 6, Issue 1, January-March 1995, pages 3-10.*

5. W.T. Reeves: Particle Systems - A Technique for Modelling a Class of Fuzzy Objects. *ACM Transactions on Graphics, Vol. 2, No. 2, 1983, pages 91-108.*

6. K. Sims: Particle Animation and Rendering Using Data Parallel Computation. ACM SIGGRAPH Computer Graphics, Volume 24, Number 4, August 1990, pages 405-413.

Contact ASHLEY T. HOWES & A.R. FORREST,
Computational Geometry Project, University of East Anglia,
School of Information Systems, Norwich NR4 7TJ, UK
ah,forrest@sys.uea.ac.uk