EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Hydraulic Erosion Using Smoothed Particle Hydrodynamics

P. Krištof[†1], B. Beneš[1], J. Křivánek[2,‡] and O. Šťava[1]

[1]Purdue University, USA
[2]Cornell University, USA and Charles University in Prague, Czech Republic

Abstract

This paper presents a new technique for modification of 3D terrains by hydraulic erosion. It efficiently couples fluid simulation using a Lagrangian approach, namely the Smoothed Particle Hydrodynamics (SPH) method, and a physically-based erosion model adopted from an Eulerian approach. The eroded sediment is associated with the SPH particles and is advected both implicitly, due to the particle motion, and explicitly, through an additional velocity field, which accounts for the sediment transfer between the particles. We propose a new donor-acceptor scheme for the explicit advection in SPH. Boundary particles associated to the terrain are used to mediate sediment exchange between the SPH particles and the terrain itself. Our results show that this particle-based method is efficient for the erosion of dense, large, and sparse fluid. Our implementation provides interactive results for scenes with up to 25,000 particles.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Physically based modeling—; Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Animation—;

1. Introduction

The influence of external factors such as wind and water is an important element that determine the morphology of natural terrains. It has been recognized in Computer Graphics (CG) that erosion simulation is essential for realistic terrain modeling. Similar to the wide variety of erosion phenomena in reality, there are many erosion algorithms in CG. The existing techniques range from slippage simulation [BYM05, LM93] to water [BTHB06, MKM89] and wind erosion [ON00]. Large amounts of effort has been devoted to the simulation of hydraulic erosion as it has the greatest influence on the terrain's appearance at different scales. However, hydraulic erosion algorithms involve fluid simulation and this makes them relatively slow and makes interactive simulations difficult. Recent GPU-oriented approaches to hydraulic erosion simulation were presented [MDH07, SBBK08] that allow for fast and interactive simulation. However, they are restricted to 2.5D water simulations. The existing full 3D Eulerian hydraulic erosion algorithms [BTHB06, WCMT07] target small scale phenomena and are far from being interactive.

In this paper, we present a visual hydraulic erosion simulation based on fully 3D water dynamics solved using the Smoothed Particle Hydrodynamics (SPH) as can be seen in Figure 8. To the best of our knowledge, this is the first attempt to couple SPH and erosion simulation. Certain properties make SPH a good candidate for erosion simulation: SPH allows for large terrain simulation as it devotes effort only to the areas with fluid, it accounts for 3D features, and it has low memory requirements.

Our erosion simulation algorithm is inspired by the physically-based approach by Wojtan *et al.* [WCMT07], who used an Eulerian approach to water dynamics. However, we have coupled the erosion simulation with a Lagrangian approach based on SPH. The eroded sediment is tracked at the SPH particles; its movement, due to the water flow and a secondary velocity field, i.e., gravity for denser sediment, is driven by means of both implicit and explicit advection. Unlike the implicit advection, which follows the flow of SPH particles, the explicit advection requires a transfer of sed-

† {pkristof | bbenes | ostava}@purdue.edu
‡ jaroslav@graphics.cornell.edu

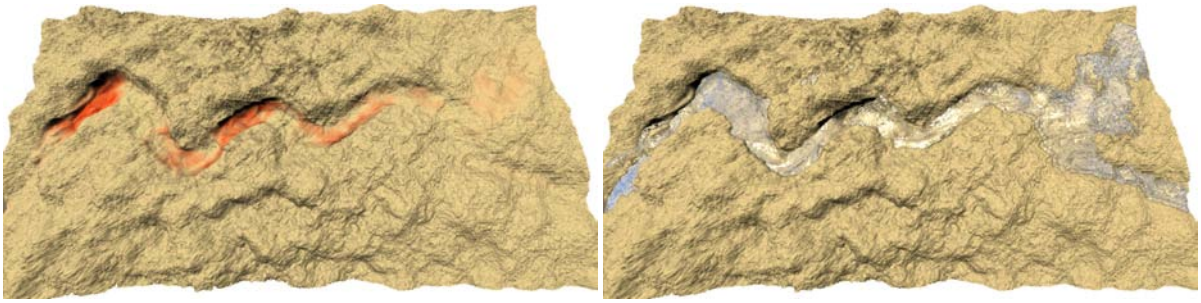

Figure 1: *Long term erosion acting on a surface. The running fluid (right) forms a river. The left image shows the corresponding erosion/deposition change of the soil in false colors.*

iment concentrations between the particles. To address the explicit advection within the context of SPH, we introduce a donor-acceptor advection scheme, which causes the sediment to advect only from the donor to the acceptor particle in the direction of the secondary velocity field, such as gravity.

The principal contributions of our paper are: (1) using SPH for erosion simulation that implies lower memory usage and better performance for large scale simulations as compared to 3D Eulerian approaches, (2) mass preserving explicit sediment advection within the SPH framework without the need for new types of particles, and (3) particle-based implementation of erosion and deposition models that allows for dynamic changes of the terrain shape.

Next we describe the related work on SPH, erosion and multi-phase flows, which are inherent to simulations of hydraulic erosion. An introduction to fluid simulation and SPH is given in Section 3. The description of erosion using SPH, follows in Section 4. Results and implementation are described in Section 5 and the paper is concluded in Section 6.

2. Previous Work

Fluid Simulation. The simulation of moving water is the basis for all hydraulic erosion methods. Because of the computational complexity of full 3D fluid solvers, a simplified 2.5D solution is often used in order to simulate water movement over large terrains. The most popular 2.5D methods are based on the Shallow Water Equation [KM90] and the Virtual-Pipes Model [OH95]. However, the simulation of 3D water requires a fully 3D fluid solver. Solvers based on the Eulerian method discretize fluid properties on a grid fixed in space [FM96, FSJ01]. While these solvers can be relatively fast and robust, their main disadvantage is the non-adaptive computational grid, which makes them unsuitable for the simulation of sparse water flows in a large domain, as for the scene in Figure 8. On the other hand, solvers based on the Lagrangian method, represent the simulation domain by particles following the water flow. Thus, the simulation focuses computation only in regions where it is needed. One of the most popular Lagrangian methods used in CG is SPH [DC96, MCG03, Mon05], which has been extended in many ways, including handling fluid-solid interactions [SSP07] or adaptive particles [APKG07]. Kipfer and Westermann showed SPH can be effectively used for simulation of rivers in [KW06] . To the best of our knowledge, our paper is the first attempt to simulate erosion in the SPH context. The complete description of the fluid simulation in CG can be found in the comprehensive survey [Bri08].

Erosion. Simulation of erosion has been a subject of active research in CG for the last two decades. In one of the first papers on the topic [MKM89], the authors applied thermal and hydraulic erosion to account for the local and global features of fractal terrains. Thermal weathering is simulated by a displacement according to the slope of the terrain. Hydraulic erosion is based on water and soil relocation to lower altitudes by employing an *ad hoc* gradient-based advection. Chiba *et al.* [CMF98] introduced hydraulic erosion directed by the water velocity field. The amount of eroded material depends on the flow and its energy. Sediment deposits when the dissolved material reaches the maximum transportable quantity. Beneš *et al.* [BF02] separated the erosion process into four independent steps that can be ran in any order and frequency to achieve performance improvement when focusing on a certain phenomena. They used a layered data structure described in [BF01] with layers of terrain, water, and dissolved soil. A full 3D Eulerian approach and a material transport equation to simulate hydraulic erosion was described in [BTHB06]. Their model supports simulation of both cohesive and cohesionless material. Recently, [SBBK08, MDH07] used a shallow-water fluid model to achieve erosion at interactive rates. Sediment transport capacity is calculated for each water cell and is proportional to the current slope angle and speed of the flow. Another recent hydraulic erosion simulation on the GPU employing a 2.5D approach is presented in [ASA07]. A modified version of the Newtonian physics approach from [NWD05] is used to simulate fluid. Deposition and erosion depend on the maximum sediment capacity. Transportable sediment capacity is proportional to the velocity of the flow and to the amount of

water in each cell. Another full 3D Eulerian simulation of erosion and corrosion was described in [WCMT07]. In their paper, scalar fields are used to track temperature, chemical concentration and sediment. Solid objects are represented by a level set stored in a 3D grid, which is transformed into the fluid grid's coordinate system every frame to compute fluid-solid interactions.

Multi-phase flows in SPH Theory of multi-phase flows describes how different phases - i.e., gas, solid and liquid - can be coupled together. Multiple phases can be quite easily implemented into the SPH solvers thanks to the particle-based nature of the SPH and that by way of representing each phase with a different type of particles. A fundamental paper on the topic is by Monaghan *et al.* [MK95], who formulates a way to handle liquid and gas phases together, but the method is general enough to apply their ideas and incorporate solid phases as well. However, adding new and more particles into the the system brings a greater perfomance impact on the simulation and so we have taken on an approximative, but effective, solution using sediment sampled at fluid particles instead. This approach is similar to the work of Lenaerts *et al.* [LAD08] who described diffusion process within a porous material.

3. Fluid Simulation

The state of a fluid can be described by a velocity field $\mathbf{v}$, a density field ρ, a pressure field p, and the fluid viscosity μ. The Navier-Stokes equations consisting of the conservation of mass (1) and the conservation of momentum (2), describe the fluid evolution over time as the function of external forces $\mathbf{f}^e$.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{1}$$

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = -\nabla p + \mu \nabla^2 \mathbf{v} + \rho \mathbf{f}^e, \tag{2}$$

Eulerian approaches provide values of the fluid on a grid, whereas the Lagrangian approaches, such as SPH, calculate the dynamics by tracking a set of moving particles. In the Lagrangian approach the equations can be substantially simplified. First, the mass conservation equation (1) is implicitly satisfied. Second, since we track the fluid properties of the particles as they move with the flow, the term $\partial \mathbf{v}/\partial t + \mathbf{v} \cdot \nabla \mathbf{v}$ in (2) can be replaced by the total derivative $\mathrm{d}/\mathrm{d}t$

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla. \tag{3}$$

3.1. Smoothed Particle Hydrodynamics

The SPH method [DC96, Mon05] is an approximative numerical solution to fluid dynamics equations (1) and (2). Originally developed by Gingold and Monaghan [GM77] and independently by Lucy [Luc77], the method builds on representing fluid as a set of particles whose quantities are interpolated from discrete positions (the so called SPH particles' positions). Each particle has its position $\mathbf{r}$, velocity $\mathbf{v}$, and mass m, which are input into the fluid dynamic equations.

A scalar quantity A at a position $\mathbf{r}$ can be interpolated from the particles using the following formula:

$$A_s(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \tag{4}$$

where m_j is the particle mass and ρ_j the density. The term W denotes a smoothing kernel around the particle $\mathbf{r}$ with radius h. The sum takes into account the nearby particles not farther than h from $\mathbf{r}$. Simulation stability is maintained when $h \geq 2\Delta s$, where Δs denotes particle spacing. In practice $h \approx 2\Delta s$. To solve Equation (2) we use techniques and kernels suggested by Müller *et al.* [MCG03]. For the time-stepping we use the Leapfrog Verlet algorithm [Ama06], which needs only one extra variable to store velocities at half-time steps and allows us to use time steps of up to $\Delta t = 2$ [ms].

3.2. External Forces and Boundary Particles

To handle the interactions, i.e., friction, sediment erosion and deposition, between the terrain and the fluid, we use the *boundary particles* on the terrain as depicted in Figure 2. The terrain is represented as a uniform heightfield and the scanline algorithm for triangle seeds boundary particles with the spacing Δs. The number of particles depends on the size of the triangle and in our experiments the number varies between one to twenty particles per triangle. The effect of different sampling of the terrain elements is subtle and is hardly noticeable.

In our simulations, the external force $\mathbf{f}^e$ is the sum of the following three components

$$\mathbf{f}^e = \mathbf{g} + \mathbf{f}^t + \mathbf{f}^b,$$

where $\mathbf{g}$ is the gravity, $\mathbf{f}^t$ is the surface tension force, and $\mathbf{f}^b$ is the force due to solid boundaries. For the simulation of water surface tension we use an effective method [BT07] based on the attraction forces between fluid particles. To resolve the behavior of fluid on the solid interface we apply forces due to the no-penetration $\mathbf{f}^{np}$ and no-slip $\mathbf{f}^{ns}$ boundary conditions

$$\mathbf{f}^b = \mathbf{f}^{np} + \mathbf{f}^{ns}. \tag{5}$$

The no-penetration condition states that the fluid cannot penetrate the boundary surface. To repel the fluid particles from the boundary we use a common penalty-force method [Ama06]:

$$\mathbf{f}^{np} = (K_S d - (\mathbf{v} \cdot \mathbf{n}) K_D)\, \mathbf{n}, \tag{6}$$

where K_S is the penalty force stiffness and K_D is the damping coefficient for the velocity $\mathbf{v}$ of an approaching fluid particle,

d is the penetrated distance measured normal to the boundary, and $\mathbf{n}$ is the unit-length surface normal. It can be seen from Equation (6) that the penalty force method behaves as a spring-based model, because the more a particle penetrates the boundary the more it is pushed away from the surface.

The no-slip condition sets the fluid's relative speed on the boundary to zero. In other words, fluid undergoes friction from the underlying boundary. We apply approach similar to Müller *et al.* [MST*04] and use the viscosity equation [MCG03] considering the boundary particles b

$$\mathbf{f}^{ns}(\mathbf{r}) = \sum_b L_b^2 \tau^{visc}(|\mathbf{r} - \mathbf{r}_b|), \tag{7}$$

where $L_b = \Delta s$ is the distance between boundary particles and τ^{visc} is traction and has unit *unit force per area* so it yields a force when integrated over boundary particles on the surface. The traction is expressed as

$$\tau^{visc}(r) = -\mu\, \mathbf{v} \nabla^2 W_v(r,h), \tag{8}$$

where μ is the boundary friction constant and W_v is the viscosity kernel [MCG03].

4. Hydraulic Erosion Model

This section describes the integration of an erosion model with SPH-based fluid simulation.

Our simulation uses two kinds of particles (see in Figure 2). The *SPH particles*, enhanced by the sediment they carry, and the *boundary particles* that enable sediment exchange between the terrain and the SPH particles. The erosion algorithm runs in the following steps:

1. Calculate fluid and boundary forces.
2. Calculate sediment transfer among SPH particles
3. Calculate erosion and deposition exchange between boundary particles and SPH particles.
4. Update terrain height according to the change of sediment in boundary particles.

The erosion model is inspired by the Eulerian approach introduced in [WCMT07]. The entrained sediment is sampled at SPH particles and the amount of sediment is denoted as a solid volume fraction C, a percentage of local volume occupied by sediment particles.

4.1. Erosion

Moving fluids incurs a shear stress τ on the solid boundary. Shear stress is a force applied on a solid object by parallel fluid forces.

To apply the shear stress to a solid, Wojtan *et al.* [WCMT07] give the solid object non-Newtonian fluid characteristics via a power-law model

$$\tau = K\theta^n,$$

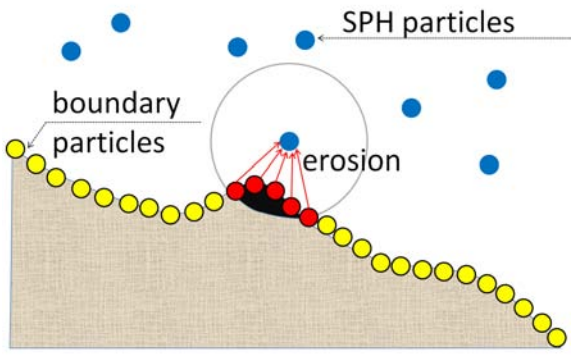

Figure 2: *In the erosion process the SPH particle takes certain amount of material from the soil using the boundary particles that are within its radius.*

where τ is the shear stress, $K = 1$ is the shear stress constant, θ is the shear rate (a measure of shear deformation) and n is the flow behavior index. We set $n = 0.5$ that is a typical value for pseudoplastic (shear-thinning) fluids. A shear-thinning of fluid exhibits decreasing viscosity with an increasing shear rate, which means that if a solid object is given enough shear stress, it will deform like a liquid. The shear rate can be approximated with a velocity of the fluid relative to the solid surface v_{rel} and distance l over which the shear is applied

$$\theta = \frac{v_{rel}}{l}, \tag{9}$$

l is the distance between a fluid and a boundary particle.

The relation between the shear stress and the erosion rate ε was formulated by Partheniades [Par65]

$$\varepsilon = K_\varepsilon(\tau - \tau_c),$$

where K_ε is the erosion strength and τ_c is the critical shear stress (material erosion resistance). The real values for the erosion rate can be found in an erosion measurement report by Wiberg [Wib03]. Finally, the change of mass at the boundary particle b due to erosion by SPH particles j within distance h is expressed as

$$\frac{dM_b}{dt} = -\sum_j L_b^2 \varepsilon(j). \tag{10}$$

4.2. Sediment Transportation

When the sediment enters the flow, its movement is determined by physical and biogeochemical processes; e.g., velocity of the flow, gravity, chemical interactions such as diffusion, heat transport, etc. The general equation describing this process was formulated in [Jen08]

$$C(\mathbf{x},t) = P(C) + J(C,\mathbf{x},t), \tag{11}$$

where P represents the physical redistribution processes, such as advection and diffusion, and J stands for sources and sinks that reflect erosion and deposition in our simulations. To incorporate the Equation (11) into the SPH we use the diffusion equation for SPH by Monaghan [Mon05]

$$\frac{dC}{dt} = \frac{1}{\rho}\nabla(D\nabla C) + J, \tag{12}$$

where dC/dt is the total derivative, denoting the time rate of change following particles at velocity $\mathbf{u}$, and D is the molecular diffusivity. Monaghan used (12) for salt diffusion and so he ignored any other advection than the one due to the velocity of the flow, meaning $\mathbf{u} = \mathbf{v}$, which is implicitly satisfied when sampling sediment C at SPH particles. In our simulation $\mathbf{u}$ is the total velocity of a sediment particle and is expressed as the sum of the velocity of the water flow $\mathbf{v}$ and the sediment settling velocity $\mathbf{v}_s$

$$\mathbf{u} = \mathbf{v} + \mathbf{v}_s. \tag{13}$$

This way (12) expresses the change of sediment concentration at the (imaginary) sediment particles. However, rather than using explicit sediment particles, our simulation couples sediment concentrations with the SPH particles. To change from the framework following sediment particles to a framework following SPH particles, we first switch into a space-fixed Eulerian frame of reference by inserting (3) and (13) into (12). The sediment transportation in the Eulerian frame is now given by

$$\frac{\partial C}{\partial t} + (\mathbf{v} + \mathbf{v}_s)\cdot\nabla C = \frac{1}{\rho}\nabla(D\nabla C) + J. \tag{14}$$

The term $\partial C/\partial t + \mathbf{v}\cdot\nabla C$ can be replaced back by the total derivative dC/dt. Then the advection-diffusion equation for sediment transport in SPH has the following form

$$\frac{dC}{dt} = -\mathbf{v}_s\cdot\nabla C + \frac{1}{\rho}\nabla(D\nabla C) + J. \tag{15}$$

The following two sections describe the numerical implementation of both the advection term $-\mathbf{v}_s\cdot\nabla C$ and the diffusion term $\frac{1}{\rho}\nabla(D\nabla C)$ in the context of SPH, as illustrated in Figure 3.

4.3. The Donor-Acceptor Advection Scheme

To implement material advection between SPH particles in the direction of the settling velocity, we introduce a model based on a donor-acceptor scheme. We define the SPH fluid particle i to be either a donor or acceptor in each i-j particle relation. If an acceptor receives sediment from a donor particle, its relative position with respect to the donor particle is in the same direction as the advection vector $\mathbf{v}_s$. Taking into consideration the donor-acceptor scheme, the corrected interpolant of the advective term $-\mathbf{v}_s\cdot\nabla C$ in (15) is formulated in the SPH:

$$-\mathbf{v}_s\cdot\nabla C_i = -\sum_j \begin{cases} m_j\frac{C_j}{\rho_j}(\mathbf{v}_s\cdot\hat{\mathbf{r}}_{ij})F(|\mathbf{r}_{ij}|,h), & \mathbf{v}_s\cdot\mathbf{r}_{ij} \geq 0 \\ m_i\frac{C_i}{\rho_i}(\mathbf{v}_s\cdot\hat{\mathbf{r}}_{ij})F(|\mathbf{r}_{ij}|,h), & \mathbf{v}_s\cdot\mathbf{r}_{ij} < 0 \end{cases} \tag{16}$$

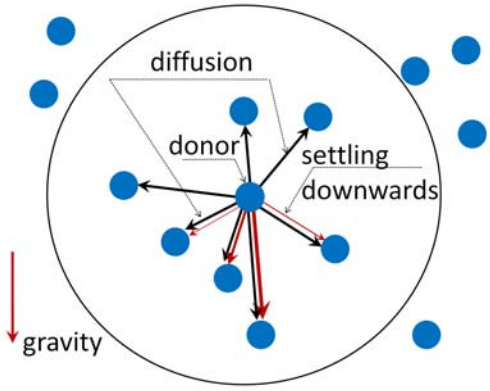

Figure 3: *The donor-acceptor scheme uses two methods for material distribution. Settling by the gravitational force (red arrows) that is stronger in the direction of gravity and diffusion (black arrows).*

where

$$\begin{aligned} \nabla W(\mathbf{r}_{ij},h) &= \hat{\mathbf{r}}_{ij}F(|\mathbf{r}_{ij}|,h), \\ \mathbf{r}_{ij} &= \mathbf{r}_i - \mathbf{r}_j, \\ \hat{\mathbf{r}}_{ij} &= \mathbf{r}_{ij}/|\mathbf{r}_{ij}| \end{aligned}$$

and F is the derivative of the cubic spline kernel W taken with respect to $|\mathbf{r}_{ij}|$ [Mon05]. Because the gradient of the cubic spline kernel is negative, Equation (16) agrees with the material advection in the direction of $\mathbf{v}_s$. Particle i acts as the acceptor for $\mathbf{v}_s\cdot\mathbf{r}_{ij} \geq 0$ and as the donor otherwise.

The magnitude of the settling velocity $\mathbf{v}_s$ depends on several physical aspects, mainly on the density and size of the sediment particles, and gravity. Similar to [WCMT07], we model all those aspects with the hindered settling velocity for small spheres in fluid

$$\mathbf{v}_s = \frac{2}{9}r_s^2\frac{\rho_s - \rho_f}{\mu}\mathbf{g}f(C), \tag{17}$$

where ρ_s, ρ_f are the sediment and fluid densities, respectively, $\mathbf{g}$ is the acceleration due to gravity, r_s is the radius of a sediment particle ($r_s = 1$ [mm] in our simulations), μ is the viscosity of the fluid, and C is the solid volume fraction at the acceptor particle. The function $f(C)$ is the hindering settling function approximating decreasing advection rate with higher sediment concentration. It is approximated using the Richardson-Zaki relation [RZ54] as

$$f(C) = 1 - (C/C_{max})^e,$$

where C_{max} is the maximum solid volume fraction in a fluid particle and $4 < e < 5.5$ is an exponent. For the case of $C > C_{max}$ we set $f(C) = 0$.

Due to the material transfer between multiple neighbors at the same time and the discrete time steps, it is necessary

to make sure that C stays in the valid range, $0 \leq C \leq C_{max}$, when evaluating Equation (16). Thanks to the hindered settling velocity, the saturated particles $C \geq C_{max}$ do not receive additional material. This significantly simplifies the problem, which can then be solved in just two steps. First, we compute the sediment exchange in both settling inside the water body and in deposition to boundary particles and save the total removal at each SPH particle. Second, we apply scaled sediment transfers to particles so that the material removed is lower or equal to the amount of sediment present in the SPH particle.

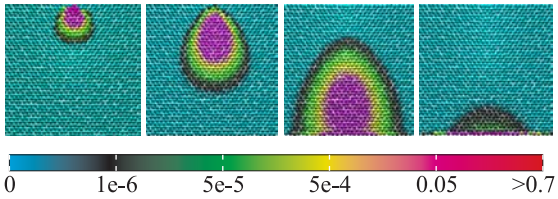

Figure 4: *Settling of the material in a static fluid. Single particle with a material on the top delivers material to the particles below. Sediment concentration C is expressed by the particle color according to the legend.*

As an alternative to our donor-acceptor scheme, we could use one of the existing advection schemes applied in Eulerian approaches, such as the unconditionally stable semi-Lagrangian algorithm [Sta99]. However, we did not encounter any stability problems due to our advection scheme and, in contrast, it does conserve the mass to the precision of floating point. It should be noted that the incompressibility equation is implemented with a soft constraint and, thus, there can be volume fluctuations in the simulation. Furthermore, Stam's algorithm involves backward steps and additional interpolation, which makes it more computationally expensive in the SPH context than our method. Figure 4 shows an example of advection of sediment on a fixed grid of particles. A single particle with sediment on the top transfers its content to the particles below.

4.4. Diffusion

Diffusion (see in Figure 5) is the tendency of material to flow from an area with high concentration to an area with low concentration. According to [Mon05], the diffusion of matter in fluid can be described as

$$\frac{\mathrm{d}C}{\mathrm{d}t} = \frac{1}{\rho}\nabla(D\nabla C), \tag{18}$$

where D is the coefficient of diffusion. We use $D = 0.1$, which, according to [Jen08], is the maximum value for strongly mixed water flows. Monaghan [Mon05] suggests expressing the SPH form of (18) with an integral approximation to the second derivative rather than twice differentiating an SPH interpolation. The reason for this is that differentiating has some serious disadvantages in SPH. In particular, it is sensitive to particle disorder and the second derivative of the kernel function can change sign, meaning, that the concentration of sediment could flow in an opposite way than it should. The diffusion equation is then

$$\frac{\mathrm{d}C_i}{\mathrm{d}t} = \sum_j \frac{m_j}{\rho_i \rho_j} D\,(C_i - C_j)\,F(|\mathbf{r}_i - \mathbf{r}_j|, h), \tag{19}$$

where F is the gradient function of a cubic spline kernel. This equation shows a positive rate of change for values of $C_i < C_j$.

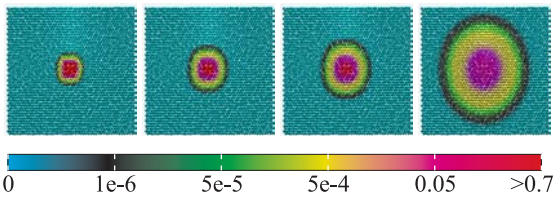

Figure 5: *Diffusion of the material in a static fluid from a particle in the middle. It can be seen that the sediment diffuses from the center in all directions opposite to the gradient of C.*

4.5. Deposition

When a fluid particle gets close enough to the ground, it deposits the captured sediment to the boundary particles. The amount of sediment mass deposited onto the boundary particle b due to exchange of C from fluid particles j is expressed as

$$\frac{dM_b}{dt} = \sum_j \rho s \frac{m_j}{\rho_j} \frac{dC(j)}{dt}, \tag{20}$$

where ρs is the density of solid material and V_f is the volume of the fluid particle. The term $dC(j)/dt$ is expressed by Equation (16) with the difference that fluid particles are strictly set as donors and boundary particles as acceptors. For the case of $(\mathbf{v}_s \cdot \hat{\mathbf{r}_{bj}}) < 0$, there is no material deposition from the fluid particle.

4.6. Terrain Modification

Boundary particles are used as a means of communication between the fluid SPH particles and the underlying terrain, as can be seen for the case of deposition in Figure 6. Erosion and deposition are calculated in two steps. First by communication between the SPH and boundary particles and, in the second step, the communication between the boundary particles and the terrain itself.

The change of mass at a triangle p is expressed as a sum of changes at the triangle's boundary particles b

$$\frac{dM_p}{dt} = \sum_b \frac{dM_b}{dt}. \tag{21}$$

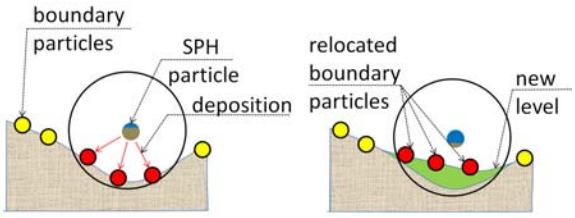

Figure 6: *Communication between a SPH particle and the boundary particles (left) and the boundary particles and the terrain (right). The sediment deposited from the SPH particle to the boundary particles in the first step (left) is used in the second step to update the model of the underlying terrain (right).*

In our implementation, the terrain is represented by a regular heightfield and adding and removing of material from a triangle corresponds to extruding and beveling the triangle vertices in the vertical-axis. Moreover, the triangles have the same projected area and therefore the volume change of a triangle is always the same as long as the total height change for all three vertices is the same. The equation for computing the total height change H of a triangle to change its mass by m is

$$H = \frac{3}{6}\frac{m}{\rho_s}\frac{1}{A_b}. \tag{22}$$

The term (1/6) reflects the fact that changing one vertex results in a change of six attached triangles to an inner vertex on the (regularly sampled) heightfield. The term A_b is the area of the vertically projected triangle. The total height change H is distributed considering the material slippage of eroded/deposited sediment grains. In the case of material deposition, that is when $H > 0$, we distribute H to the lowest triangle vertices until their heights are even, and then we add the height uniformly. Similarly, in case of erosion, the height is removed first from top-most vertices until their heights are even and then uniformly from all three.

5. Implementation and Results

Our implementation uses SPH calculation on the CPU. To find particle neighbors we use a kd-tree, which is efficient when dealing with scattered data sets, with sliding midpoint and implicit pointers techniques. For simulations of densely occupied domains, with less than 50,000 particles, we found the kd-tree to be negligibly slower than traditional hashing into a 3D grid. Our SPH simulation outputs updated positions of the particles that define the fluid.

To render the fluid we first re-sample the fluid density into the regular grid (density grid) using the same density function as for the SPH computations, find the cubes crossing the isosurface, and then apply iso-surface reconstruction using marching tetrahedra on the GPU [Pas04] . We have not used any isosurface function correction techniques.

We attempt to simulate large scenes with scattered fluids so finding the free level of water on a regular grid becomes quickly a performance bottleneck. To improve the performance we generate an additional boolean grid, which is four times smaller in each dimension than the density grid and store true value if there is a density contribution from any particle. Then we check for the isocubes in the density grid only if there is a true value in the boolean grid. In this way we have achieved a speedup of factor 20 in the finding of isocubes and three in complete isosurface generation.

The generation of a neighbor list of boundary particles takes approximately 50% of the simulation time. The neighbor list is created by querying neighboring triangles and checking each boundary particle to be within distance h. The erosion/deposition requires approximately the same time as the SPH calculation.

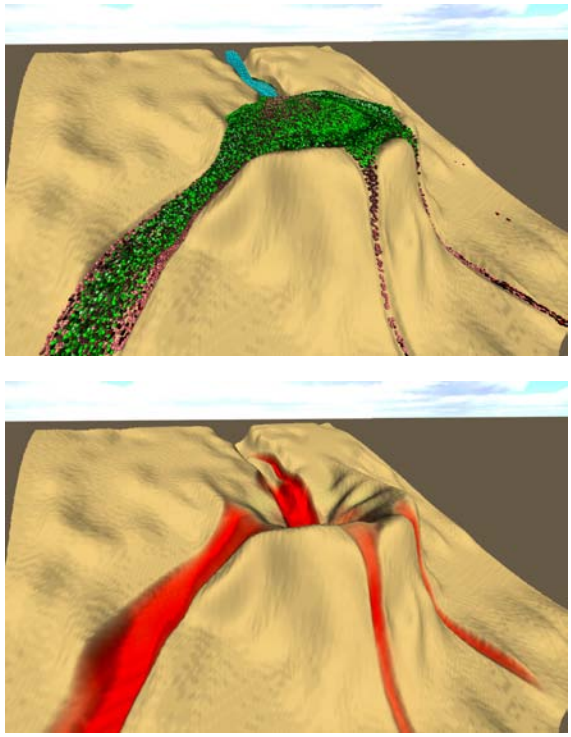

Figure 7: *Example of a lake being filled by water that erodes away from the boundary.*

Rendering of the free level of water is enhanced by using several image-space techniques, namely we render reflections, refractions, and caustics. Reflections are generated using cube mapping and refractions are achieved by perturbing texture coordinates for background texture look ups as described in [CLT07, Lun06]. To simulate caustics, we use

the caustics mapping of Shah *et al.* [SKP07]. The rendering time, compared to the SPH calculation, is small. For example the computation of caustics, using texture of size 1280x800, takes around 0.1 second.

All simulations and experiments were performed on an Intel Quad Core Q6600 at 2.4 GHz equipped with NVIDIA 8800 GT. The computational overhead of erosion simulation with respect to the SPH calculations is around 100% (see Table 1). For all of our simulations we set the erosion strength $K_\varepsilon = 0.1$ and the critical sheer stress $\tau_c = 3$.

Scene	Particles	SPH	Erosion	Boundary
Canyon	136,000	0.33 s	0.39 s	0.66 s
Lake	90,000	0.31 s	0.17 s	0.42 s
Waterfall	650,000	1.25 s	1.05 s	2.20 s

Table 1: *Computational times of the scenes from the paper. Each scene is described by the number of particles, time of calculation of the SPH, time of erosion calculation, and time of boundary collision calculation.*

The example in Figure 1 shows the effect of long term erosion on a terrain. The terrain is exposed to a river that finds its way through and forms a river channel. The simulation consists of 130,000 particles and a heightfield of dimensions 700×350. The computation time was 1.38 sec per time step. The SPH itself took 0.33 sec, the erosion and deposition took 0.39 sec, and the generation of the list of neighboring boundary particles took another 0.66 sec.

The scene in Fig 7 shows an overfilling lake and subsequent creation of new river paths. The simulation of 90,000 particles and the computation took 0.9 sec per time step.

The largest scene in Figure 8 shows the creation of waterfalls due to two sources of water. We used 650,000 particles, heightfield of dimensions 728×512 and the computational time was 4.5 sec per time step.

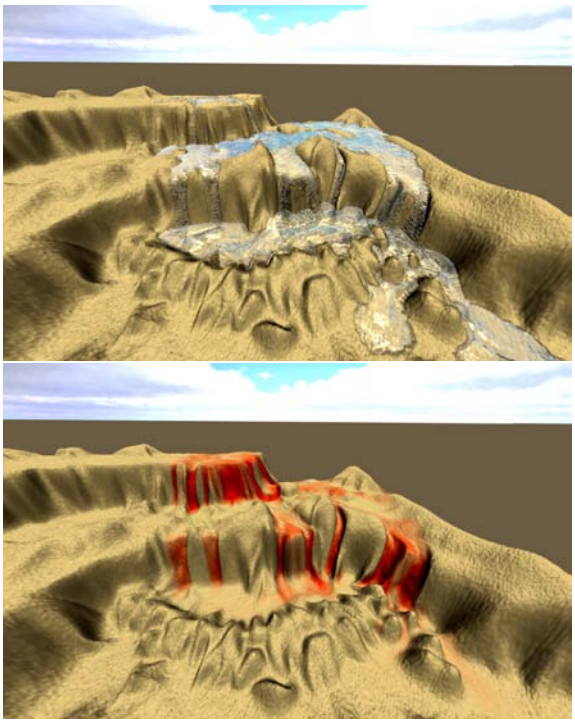

Figure 8: *Creation of multi-level waterfalls. Fluid made up of 650,000 particles (top) erodes the underlying terrain. Erosion is displayed in false colors—the level of red represents the decrease in terrain height.*

6. Conclusions and Future Work

We have presented a solution to coupling hydraulic erosion with SPH-based fluid simulation. Our design choices were motivated by the goal to keep the overhead of erosion simulation small. For that very reason, rather than introducing separate sediment particles, we represent the dissolved sediment as a volume fraction in SPH particles. Since the sediment does not entirely follow the water flow, we introduce an additional advection by the gravity field. We propose a novel donor-acceptor scheme for sediment advection in the context of SPH. We use boundary particles between the terrain and the water particles, for sediment exchange. Our results show that SPH affords for interactive erosion simulation in a moderately complex environment with fully 3D water.

There are many possible avenues for the future work. We have presented a general solution, but we have demonstrated it only on a height field. One important future work is an extension to a full 3D representation that would allow for carving caves, creating overhangs etc. Another possible future work would focus on a better way of geometry alterations, which are due to erosion and deposition, e.g., by distributing material changes to vertices via barycentric extrapolation [BFA02]. In addition, the performance of generation the particle neighbor list could be improved by using time coherence. Finally, we also want to analyze the stability of our SPH-based advection scheme.

Acknowledgements

We would like to thank to reviewers for their valuable and constructive comments.

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics). J. Křivánek acknowledges support from the Marie Curie grant number PIOF-GA-2008-221716.

References

[Ama06] AMADA T.: Real-time particle-based fluid simulation with rigid body interaction. In *Game Prog. Gems 6* (2006), Charles River Media, pp. 189–205.

[APKG07] ADAMS B., PAULY M., KEISER R., GUIBAS L. J.: Adaptively sampled particle fluids. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), ACM, p. 48.

[ASA07] ANH N. H., SOURIN A., ASWANI P.: Physically based hydraulic erosion simulation on graphics processing unit. In *GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia* (New York, NY, USA, 2007), ACM, pp. 257–264.

[BF01] BENEŠ B., FORSBACH R.: Layered data representation for visual simulation of terrain erosion. In *SCCG '01: Proceedings of the 17th Spring conference on Computer graphics* (Washington, DC, USA, 2001), IEEE Computer Society, p. 80.

[BF02] BENEŠ B., FORSBACH R.: Visual simulation of hydraulic erosion. *Journ. of WSCG 10*, 1 (2002), 79–86.

[BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (2002), ACM, pp. 594–603.

[Bri08] BRIDSON R.: *Fluid Simulation for Computer Graphics,*. A K Peters, 2008.

[BT07] BECKER M., TESCHNER M.: Weakly compressible SPH for free surface flows. In *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation* (University of Freiburg, 2007), pp. 1–8.

[BTHB06] BENEŠ B., TĚŠÍNSKÝ V., HORNYŠ J., BHATIA S. K.: Hydraulic erosion. *Computer Animation and Virtual Worlds 17*, 2 (2006), 99–108.

[BYM05] BELL N., YU Y., MUCHA P. J.: Particle-based simulation of granular materials. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM Press, pp. 77–86.

[CLT07] CRANE K., LLAMAS I., TARIQ S.: *GPU Gems 3 - Real-Time Simulation and Rendering of 3D Fluids*. Addison-Wesley, 2007, ch. 30 (Refraction 3.2).

[CMF98] CHIBA N., MURAOKA K., FUJITA K.: An erosion model based on velocity fields for the visual simulation of mountain scenery. *The Journal of Visualization and Comp. Animation, 9*, 4 (Oct.–Dec. 1998), 185–194.

[DC96] DESBRUN M., CANI M.-P.: Smoothed particles: a new paradigm for animating highly deformable bodies. In *Proceedings of the Eurographics workshop on Computer animation and simulation '96* (New York, NY, USA, 1996), Springer-Verlag New York, Inc., pp. 61–76.

[FM96] FOSTER N., METAXAS D.: Practical animation of liquids. In *Graphical Models and Image Processing* (1996), pp. 23–30.

[FSJ01] FEDKIW R., STAM J., JENSEN H. W.: Visual simulation of smoke. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), ACM, pp. 15–22.

[GM77] GINGOLD R. A., MONAGHAN J. J.: Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Royal Astronomical Society 181* (Nov. 1977), 375–389.

[Jen08] JENKINS W.: Modeling, data analysis and numerical techniques for geochemistry, 2008. http:w3eos.whoi.edu/12.747/chaps/chap11/chapter_ade.pdf.

[KM90] KASS M., MILLER G.: Rapid, stable fluid dynamics for computer graphics. *Computer Graphics (Proceeding of SIGGRAPH 90) 24(4)* (1990), 49–57.

[KW06] KIPFER P., WESTERMANN R.: Realistic and interactive simulation of rivers. In *GI '06: Proc. of Graphics Interface 2006* (Toronto, Ont., Canada, Canada, 2006), Canadian Information Processing Society, pp. 41–48.

[LAD08] LENAERTS T., ADAMS B., DUTRÉ P.: Porous flow in particle-based fluid simulations. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–8.

[LM93] LI X., MOSHELL M.: Modeling soil: Realtime dynamic models for soil slippage and manipulation. In *Proceedings of SIGGRAPH'93* (1993), vol. 27(4) of *Annual Conference Series*, pp. 361–368.

[Luc77] LUCY L. B.: A numerical approach to the testing of the fission hypothesis. *Astronomical Journal 82* (Dec. 1977), 1013–1024.

[Lun06] LUNA F.: Pond water, 2006. http:www.moon-labs.com/resources/pondwater.pdf.

[MCG03] MÜLLER M., CHARYPAR D., GROSS M.: Particle-based fluid simulation for interactive applications. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 154–159.

[MDH07] MEI X., DECAUDIN P., HU B.-G.: Fast hydraulic erosion simulation and visualization on gpu. In *Pacific Graphics* (Los Alamitos, CA, USA, 2007), vol. 0, IEEE Computer Society, pp. 47–56.

[MK95] MONAGHAN J. J., KOCHARYAN A.: SPH simulation of multi-phase flow. *Computer Physics Communications 87* (May 1995), 225–235.

[MKM89] MUSGRAVE F. K., KOLB C. E., MACE R. S.: The synthesis and rendering of eroded fractal terrains. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1989), ACM Press, pp. 41–50.

[Mon05] MONAGHAN J. J.: Smoothed particle hydrodynamics. *Reports on Progress in Physics 68*, 8 (2005), 1703–1759.

[MST*04] MÜLLER M., SCHIRM S., TESCHNER M., HEIDELBERGER B., GROSS M.: Interaction of fluids with deformable solids: Research articles. *Comput. Animat. Virtual Worlds 15*, 3-4 (2004), 159–171.

[NWD05] NEIDHOLD B., WACKER M., DUESSEN O.: Interactive physically based fluid and erosion simulation. In *Proceedings of Eurographics Workshop on Natural Phenomena* (2005), pp. 25–32.

[OH95] O'BRIEN J. F., HODGINS J. K.: Dynamic simulation of splashing fluids. In *CA '95: Proceedings of the Computer Animation* (Washington, DC, USA, 1995), IEEE Computer Society, p. 198.

[ON00] ONOUE K., NISHITA T.: A method for modeling and rendering dunes with wind-ripples. In *PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2000), IEEE Computer Society, p. 427.

[Par65] PARTHENIADES E.: Erosion and deposition of cohesive soils. *Journal of Hydraulics Division of the American Society of Agricultural Engineer 92* (1965), 105–139.

[Pas04] PASCUCCI V.: Isosurface computation made simple: hardware acceleration, adaptive refinement and tetrahedral stripping. *In Joint Eurographics - IEEE TVCG Symposium on Visualization VisSym* (2004), 293–300.

[RZ54] RICHARDSON J. F., ZAKI W. N.: Sedimentation and fluidization. *Transactions on the Institution of Chemical Engineers 32* (1954), 35–53.

[SBBK08] ŠŤAVA O., BENEŠ B., BRISBIN M., KŘIVÁNEK J.: Interactive terrain modeling using hydraulic erosion. *ACM Siggraph/Eurographics Symposium on Computer Animation* (2008), 30–39

[SKP07] SHAH M. A., KONTTINEN J., PATTANAIK S.: Caustics mapping: An image-space technique for real-time caustics. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 272–280.

[SSP07] SOLENTHALER B., SCHLÄFLI J., PAJAROLA R.: A unified particle model for fluid-solid interactions. *Comp. Anim. and Virtual Worlds 18*, 1 (2007), 69–82.

[Sta99] STAM J.: Stable fluids. In *Proc. SIGGRAPH* (1999), 121–128.

[WCMT07] WOJTAN C., CARLSON M., MUCHA P. J., TURK G.: Animating corrosion and erosion. In *Eurographics Workshop on Natural Phenomena* (2007), 21-29

[Wib03] WIBERG P.: Sediment erosion and redistribution in fine-grained shelf environments. *Coastal Geosciences (CG): Annual Reports: FY07* (2003).