Simulating the Flow of Liquid Droplets

Patrick Fournier[†] Arash Habibi[†*] Pierre Poulin[†]

[†] Département d'Informatique et de Recherche Opérationnelle
Université de Montréal

[*] ACROE Grenoble

Abstract

The ever-changing nature of liquids makes them very difficult to model and animate. This paper addresses the simulation of one aspect of liquids, i.e. droplets running down surfaces. We present a model oriented towards a visually-satisfying simulation and efficiency.

The efficiency results from the separation between the shape and the motion of a droplet. The motion accounts for all changes encountered along the path followed over a mesh of triangles. It is affected by various properties modeled as friction, adhesion, roughness, and collisions between droplets. Streaks are also added along the paths.

We characterize the shape of a droplet by a small set of properties, such as volume conservation, surface tension, etc. We model them as constraints to satisfy. The shape model is mainly based on mass-springs. It is simple and efficient, and it guarantees that whatever the values of the unconstrained parameters, all produced shapes satisfy the characteristic properties, and thus, can represent different types of droplets.

Rendered animations of various liquids illustrate the resulting simulation.

Keywords: natural phenomena, drops, particle system, mass-spring model, textured illumination.

1 Introduction

Nature provides an endless realm of phenomena to model, animate, and render. Its richness and complexity offer an incredible challenge for any computer artist. For this reason, several natural phenomena have been successfully modeled in computer graphics over the past years. They include among others terrains, plants, waves, clouds, and smoke. Their contributions to realism are of great importance and a constant source of inspiration and motivation.

Several researchers have addressed the simulation of water motion in computer graphics in the form of waves [22, 27, 10, 31] and connected fluids [18, 8, 3]. Solving accurately the fluid dynamics in its environment is a very complex task as a result of the ever-changing shape and topology of water.

The work presented in this paper addresses the simulation of droplets as they flow down a surface. While much of the work in simulating water has been devoted to water motion as a connected body, very little looked at water in the form of droplets. Nishikawa and Abe [24] study the deformations produced on a falling droplet as it enters in contact with a surface. They use the Navier-Stokes equation, and distribute massless markers in the liquid to determine the water surface position. Although very realistic in terms of physical simulation, extending this approach to handle hundreds or thousands of droplets running onto surfaces would be prohibitively expensive.

Miller and Pearce [23] and Tonnesen [30] study the attraction and repulsion forces between particles to simulate various degrees of fluid viscosity and matter state change such as melting. Their model is more concerned with the interaction between liquid particles than their behavior as they flow on surfaces. Moreover, all their particles are spherical, which in the case of liquid droplets lacks some realism.

O'Brien and Hodgins [26] present a general model that divides each phenomena by treating the liquid either as a volume, a surface, or a set of particles, each model interacting with the others. They produce simulations of splashing and waves as objects (forces) collide with the liquid. The drops of liquid do not interact with each other, and are simply merged with the liquid volume when they fall back into it. Finally, they do not handle the flow of droplets over surfaces.

The work of Dorsey *et al.* [7] is more closely related to our goals. They simulate several weathering effects as water deposits sediments while flowing on surfaces. The structures of their particle system and their surfaces appear similar to ours. However because their focus is more involved with the staining effects over a long period of time, they do not provide any equation for the motion of their droplets. Moreover they are not concerned with the shapes and visual appearances of the droplets themselves.

1.1 Our Simulation System

This paper focuses on the simulation of *large* liquid droplets as they flow down a surface. We are mainly interested in a *visually-appealing* simulation handling up to thousands of droplets within seconds of computing. Therefore, although we base our derivations from the physics, we assume simplifying conditions that allow for an efficient simulation. In fact an accurate physical simulation would involve a tremendous toll on processing as one needs to consider the motion of liquid within a droplet, the capillarity of its surface, the modeling of the terrain roughness, and the interaction forces between each point on the droplet surface and the terrain. This accuracy goes far beyond the scope of this paper.

We represent a droplet as a particle in a particle system and its motion is determined in this scheme. Section 2 describes how various droplet parameters and the terrain can influence its motion. It also shows how the droplet can modify its environment by interacting with other droplets and leaving streaks of liquid on the terrain. We assume that the droplet deformations do not affect its global motions. Therefore the shape can be derived from the motion without any feedback. This shape is modeled by dynamically satisfying a set of constraints developed in Section 3. Finally the rendering of droplets is presented in Section 4.

2 Motion

The motion of droplets is generated by a particle system [28] in which each droplet is represented by one particle. Such a representation offers several advantages in terms of generality and flexibility in order to simulate a wide spectrum of behaviors. The motion of a droplet is governed by the classical mechanics: $\vec{F} = m\vec{a}$. A droplet therefore contains a position $\vec{p}$, a velocity $\vec{v}$, an acceleration $\vec{a}$, and a mass m. The size of a droplet is derived from its mass and the liquid density.

In our system, a droplet flows over a surface defined as a mesh of triangles. This general representation (most surfaces can be approximated by a mesh of triangles) simplifies the motion equations and collision detection between droplets. At the beginning of the simulation, we build a *neighborhood* graph in which each triangle is linked to the triangles adjacent to itself. During the simulation, each triangle knows which droplets are currently over its surface, and each droplet knows which triangle it lies on. The triangles are usually small enough so that few droplets lay generally on the same triangle at any given time.

2.1 Motion on the Mesh

During the simulation, each droplet rolls on the mesh. Between two time steps (consecutive frames), a droplet might travel over several triangles. We compute the motion of the droplet over each of these triangles, to ensure that the droplet is properly affected by *all* the deformations of the surface it traversed. When a droplet goes from one triangle to another, we use the triangle neighborhood graph to quickly identify the triangle which the droplet moves to. The passage from one triangle to the next produces a collision if they form an angle less than π; this collision is treated as perfectly inelastic.

The motion on bi-parametric surfaces has also been studied [15, 7]. However the scheme based on projecting the motion onto the tangential plane and correcting to ensure the point remains on the surface is expensive and prone to errors. Using more frequent simulation time steps as a function of surface details frequency and droplet velocity can reduce artifacts but not eliminate them.

2.2 Motion on a Triangle

Two forces applied on our droplets are gravity $\vec{F_g}$ and friction $\vec{F_f}$. If we suppose that forces on a droplet remain constant over a triangle, we can derive an equation for the position of a droplet at any moment. The gravitational force $\vec{F_g}$ does not change as the droplet moves inside a static triangle because triangles are planar. The friction due to surface roughness is modeled as a linear viscous force with a constant negative factor k_f. The tangential force due to this friction is expressed in function of time t as

$$\vec{F_f}(t) = k_f \vec{v}(t).$$

In reality, k_f varies on a given surface. However because our triangles are usually small, we chose to approximate the friction coefficient over a single triangle with an *average friction coefficient*.

Therefore, considering the constant resulting force $\vec{F_r} = \vec{F_g} + \vec{F_f}$ assumption over a triangle, we can derive the equations for velocity and position of a droplet at any time t:

$$\vec{v}(t) = -\frac{\vec{F_r}}{k_f} + \left(\vec{v}(t_i) + \frac{\vec{F_r}}{k_f}\right) e^{k_f t/m}$$

$$\vec{p}(t) = \vec{p}(t_i) - \frac{\vec{F_r} t}{k_f} + m\left(\vec{v}(t_i) + \frac{\vec{F_r}}{k_f}\right)\left(\frac{e^{k_f t/m} - 1}{k_f}\right).$$

When a droplet enters a triangle at time t_i, we immediately compute when and where it will exit. This corresponds to intersecting its trajectory with each side of the triangle using the Newton-Raphson root finding algorithm.

Table 1 provides the computation time for randomly generated droplets to traverse a rectangle subdivided in a mesh of polygons (no rendering). One can observe that the mesh resolution does not influence much the computation time for the traversal of a droplet. Moreover as expected, the computation time increases quite linearly as a function of the number of droplets.

Number of Polygons	Number of Droplets 10	20	40	60	80	100
2	1.0	2.0	4.1	6.1	8.0	10.6
8	0.9	1.7	3.8	6.2	7.9	10.1
32	0.9	1.6	4.0	6.2	8.1	10.2
128	1.0	2.2	4.1	6.5	8.5	10.2
512	1.0	2.1	4.2	6.4	8.3	10.1

Table 1: Droplets traversing a mesh of polygons (in sec.)

Number of Polygons	Number of Droplets 1000	2000	4000	8000
2	0.63	2.56	12.75	59.37
8	0.21	0.80	3.12	15.71
32	0.05	0.30	1.28	5.84
128	0.05	0.20	0.81	4.18
512	0.01	0.10	0.62	3.10

Table 2: Droplets (no modeling) traversing a mesh of polygons (in sec.)

Computing the shape of each droplet is more than 100 times more expensive than its traversal. The modeling phase is factored out in Table 2. Since that as the number of polygons increases, less droplets lie on the same polygon, we can observe that the $O(n^2)$ factor for intersections between droplets becomes less important. Without these intersections, traversal is very inexpensive to compute (as hinted by the last row of the table), and we observed a small linear cost as the number of polygons traversed increases.

2.3 Surface Properties

Many surface properties can affect the behavior of our droplets. We consider adhesion and roughness. The droplet flows on the surface because it *adheres* to it. The adhesion is a force along the surface normal. It is function of the contact area between the droplet and the surface, and the adhesion coefficient $D_\triangle$ between the liquid and the surface. A droplet will fall from a surface if its adhesion force is smaller than the component of its acceleration force that is normal to the surface. As illustrated in Fig. 1 (left), an hemispherical droplet of radius r will fall off the surface if

$$\vec{a_N} > \frac{D_\triangle \pi r^2}{m}.$$

We assume the surface roughness will only reduce the tangential force. This factor is function of the size of the bumps relative to the size of the droplet (Fig. 1 (right)):

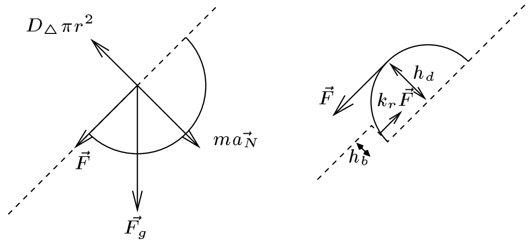

Figure 1: The effects of adhesion and roughness

$$k_r = \begin{cases} h_b/h_d & h_b < h_d \\ 1 & \text{otherwise} \end{cases}$$

where h_d is the height of a droplet, and h_b is the height of a bump. The effect of roughness on force $\vec{F}$ applied on a droplet is simulated by

$$\vec{F} \leftarrow \vec{F}\,(1 - k_r)\,.$$

2.4 Collisions between Droplets

By considering small triangles, only a few droplets should lay over a given triangle at any given time. This greatly reduces the detection of collisions. Moreover, we will only detect collisions at the time of the frames. This important reduction of complexity comes at the expense that we might miss actual collisions between frames. However it will always be possible to compute more frames than displayed, therefore reducing the number of missed collisions.

When two droplets collide, the resulting deformations happen very fast in reality; within our context, we simply assume this is instantaneous. The combined droplet has new mass, position, velocity, and acceleration. The new mass is simply the sum of the two colliding droplets; the new position and velocity are computed as a weighted average of the positions and velocities of the two colliding droplets. We also have to compute the exit point and time on the triangle for this new droplet.

2.5 Leaving Liquid Streaks

The path followed by a droplet on a triangle might affect the paths of future droplets crossing this path, for instance by filling some of the holes due to roughness. We simulate this effect by reducing the roughness along this path. Instead of intersecting the path of a droplet with all the streaks left by previous droplets, we use two roughness parameters describing the roughness along the X and the Y axes. When a droplet passes through a triangle, it leaves a streak. To simplify computations, we

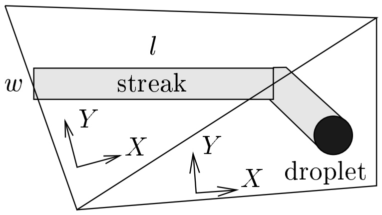

Figure 2: A polygonal streak left by the droplet

consider that streaks on the triangle forms a rectangular area of the width of the droplet. Its length l is the distance between the entry and exit points (Fig. 2). We compute the *projected area* of the streak on the two axes:

$$\begin{aligned} A_x &= w|\sin\theta| + l|\cos\theta| \\ A_y &= w|\cos\theta| + l|\sin\theta| \end{aligned}$$

which we use to modify the associated roughness coefficients k_{rx} and k_{ry} for the triangle:

$$k_{rx} = \begin{cases} k_{rx}\,(1 - A_x/A_t) & A_x \le A_t \\ 0 & \text{otherwise} \end{cases}$$
$$k_{ry} = \begin{cases} k_{ry}\,(1 - A_y/A_t) & A_y \le A_t \\ 0 & \text{otherwise} \end{cases}$$

where A_t is the area of the current triangle. If a droplet stays over the same triangle for more than one frame, we replace this rectangle streak by a list of rectangles corresponding to the previous and current positions for the droplet. Therefore the smaller the triangles are, the more accurate solution becomes. The appearance of these streaks are discussed in Section 4.

3 Shape

Generally, motion and shape are closely and mutually correlated and must be simulated globally in a single model. However, in many cases [14] such as in liquid droplets, the energy involved in the motion of the drop is much higher than the energy involved in its deformations. This means that motion can be simulated regardless of any shape consideration, but the shape must be simulated with motion inputs. This would not have been possible, for example in the case of interacting solid objects, since their motion depends on this interaction, which, in turn, depends on their shape.

Thus in this section, given the motion of a droplet, our aim is to simulate its *visual contour* and shape as it interacts with the underlying solid surface and with other force fields.

It seems difficult to explicitly describe the shape of a droplet, all the more as this shape may be subject to important variations as the drop moves on the surface. However we can easily state a certain number of properties that must always be satisfied by this shape. Next we can use these properties as constraints enabling us to design a model in which all these constraints will be automatically satisfied.

In this section we state and describe these constraints. Next we present a simple dynamic model representing the drop's contour, and add to this model the necessary dynamic links capable of satisfying these constraints.

3.1 Shape Modeling with Dynamic Constraints

We assume that the shape of a droplet can be understood as the result of the interaction between two competing tendencies due to different physical phenomena :

Constraint a: The tendency to have constant volume (a liquid is incompressible).

Constraint b: The tendency to have the smallest contact surface with air (capillarity forces).

These tendencies have a geometrical nature, therefore the shape generation process could be based on geometrical operations [2, 5]. This approach produces a shape each time all constraints provide a unique solution. In all other cases (underdetermination or overdetermination), no shape can be produced. For example the two above constraints are sufficient to produce a spherical shape. Strictly speaking any supplementary condition would produce an overdetermined problem.

The alternative is a dynamic shape generation process Constraints can be expressed as competing dynamic properties that bring the model to a balance position which either satisfies all constraints or is the best (most stable) compromise between them. Underdetermination simply results in a model that has an infinity of balance shapes all satisfying the constraints. Choosing dynamic shape generation for liquid droplets enables us to specify several other constraints in order to achieve more complex shapes. For droplets, we consider two additional constraints. Constraint **c** accounts for the interaction of the droplet with a flat hydrophilic surface, whereas constraint **d** accounts for the action of external forces such as gravity and any other uniform force field.

Constraint c: The tendency to have the largest contact surface with the underlying surface.

Constraint d: The tendency to be deformed by external forces (in this work we limit our concern to uniform force fields).

We assume these four constraints are sufficient to produce realistic droplet shapes moving on a surface. Therefore in the remaining of this paper, we will not seek to produce a physical model of a droplet but merely to dynamically satisfy these four constraints without any further physical considerations.

We propose a model and we prove that this model indeed satisfies these constraints. For the sake of conciseness, these proofs are not developed in this paper. For those readers specifically interested in these developments, a detailed discussion is available through the site associated with this paper on `www.iro.umontreal.ca/labs/infographie/papers`

3.2 Modeling Hypotheses

3.2.1 A Spring-mass Model

Our model is based on point-mass (or particles) which can cover a wide range of fluid phenomena. Point-mass models are popular in computer graphics [4, 29, 30, 23, 9, 20, 13] and are also used in physics [11, 6]. They can account for small deformations as in the case of rigid objects [25] as well as for high deformations and even topology changes. We simply use them as a general-purpose dynamic modeler. Thus point-mass modeling is a widespread approach. So novelty in this work is not to be found in the modeler itself, but in the expression of droplet's dynamic deformations in terms of this already-existing widespread paradigm.

We do not explicitly deal with the shape of merging or separating droplets (i.e. no topology changes), therefore each point-mass in our model is permanently connected to a limited and fixed set of other point-masses by links accounting for the forces exchanged between point-masses. If we limit our concern locally to a first order (linear) approximation of the involved forces, it follows that a great part of our model can simply be expressed as a *mass-spring* model. This is not too restrictive since we know [19, 16] that a great number of non-linear interactions can be modeled by the assembly of linear interactions.

3.2.2 Quasi-static Hypothesis

Balance can be reached after a period of transient motion. We assume droplets reach a balance shape quickly. Therefore our sequences are only composed of balance shapes obtained after some steps of non-displayed simulation. Dealing only with balance shapes has several important consequences.

¿From a computational point of view, we know that as long as the external forces exerted on the droplet remain the same, its shape remains unchanged and therefore can be reused without further simulation.

Second order mechanical systems are entirely governed by the well-known relations:

$$\vec{F} \;=\; m\vec{a}$$

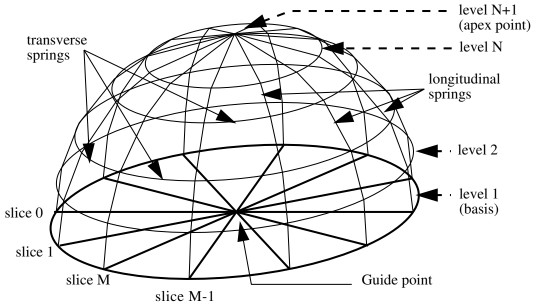

Figure 3: The structure of the droplet model

$$\vec{F} \;=\; \left(k(l_0 - l) - z\frac{dl}{dt}\right)\vec{u}.$$

They respectively express the acceleration $\vec{a}$ of a point of mass m on which force $\vec{F}$ is applied, and the force $\vec{F}$ produced by a damped spring of stiffness k, viscosity z, rest length l_0 and length l ($\vec{u}$ is the unit vector of the spring).

Balance implies zero velocities and zero accelerations. According to the above relations this means that the balance shape (i.e. all our sequences) does not depend on any mass or any viscosity, but only on the links stiffness and rest length.

It should be noted that balance shape does not depend on mass considered as a coefficient of inertia, but it does depend on the weight of masses. We account for gravity separately by a uniform force field, and therefore shape is indeed independent of masses.

The control of viscosity and mass simply enables us to reach balance as rapidly as possible. Therefore the balance shape is only derived from the links stiffness and rest length.

3.2.3 The Structure of the Model

We have not specifically studied the transition shape between a free droplet and a droplet moving on a surface. It can be trivially proved that in the absence of underlying surface, constraints **a**, **b**, and **d** result in a sphere for the balance droplet shape. However this is not the case when all four constraints are considered. We assume the curvature of the underlying surface is fairly low in the vicinity of the droplet so this surface can be locally modeled by a normal vector and a point (*guide point*).

We have experimentally observed that our model satisfies the four constraints for different discretizations of the droplet surface. However, for the discretization shown on Fig. 3, this can be analytically proved. In the remaining of this paper we will always refer to this discretization.

Our model is characterized by M radial slices, N levels, and the apex point A_{N+1}. Thus the model is composed of $NM + 1$ point-masses. Point $A_{i,j}$ is the point-mass of level $i \in [1, N]$ and slice $j \in [1, M]$. Points $A_{1,j}$ are in contact with the underlying surface and are called the *basis points*.

In order to obtain a smooth final shape for the droplet, this network of points forms the control points of a B-spline patch. This dissociates the geometrical smoothness, which may require many points, from the number of mechanical degrees of freedom, which generally requires much fewer points. We obtain fine results even with as low values as $M = 3$ and $N = 2$. Most of the images in this paper were in fact computed with these values.

3.3 The Dynamic Resolution of the Constraints

The basis points are attached to the underlying surface by very stiff springs with a zero rest length. These springs always stay perpendicular to the surface. Therefore the basis points can freely move on this surface but can hardly get away from it.

We describe next our model and discuss the conditions for which the constraints are met.

3.3.1 Satisfying Constraints b, c, and d

In order to realize constraint **c**, all basis points are linked to the guide point by springs which tend to increase the area of the basis far more than the nominal radius of the droplet. These springs are characterized by a very large rest length l_0 and by a stiffness controling the deformability for the droplet shape. These springs constraint only the basis points.

In order to realize constraint **b**, we link the point-masses of our model in a network of springs. More precisely *longitudinal* springs of stiffness k_l and rest length l_{0l} are arranged so that

$$(\forall j \in [1, M])\ (\forall i \in [1, N-1]) \quad A_{i,j} \text{ is linked to } A_{i+1,j}$$
$$(\forall j \in [1, M]) \quad A_{N,j} \text{ is linked to } A_{N+1}$$

and *transverse* springs of stiffness k_t and rest length l_{0t} so that

$$(\forall i \in [1, N])\ (\forall j \in [1, M-1]) \quad A_{i,j} \text{ is linked to } A_{i,j+1}$$
$$(\forall i \in [1, N]) \quad A_{i,M} \text{ is linked to } A_{i,1}.$$

In order for the droplet surface to be as small as possible (constraint **b**), we assign zero rest lengths to all longitudinal springs.

The realization of constraint **d** is trivial.

The Homogeneity of the Surface

This set of springs and masses is the discrete counterpart of a continuous physical surface with homogeneous density and homogeneous tension. However, in the discrete surface, the masses are not homogeneously distributed. The number of masses in the vicinity of the apex is higher than at the basis, and the transverse lengths between masses are smaller. If all of the masses were equal and if all transverse stiffnesses were equal, then the apex would be a region of high density and high stiffness. In order to have a globally homogeneous discrete surface, we may assign different values to masses situated on different levels. For the same reason, we will have to assign to the transverse springs of level i respectively a rest length ${l_0}_t^i$ and a stiffness k_t^i of:

$$\begin{aligned} {l_0}_t^i &= {l_0}_t^1 \frac{(N+1-i)}{N} \\ k_t^i &= k_t^1 \frac{(N+1-i)}{N} \end{aligned}$$

where ${l_0}_t^1$ and k_t^1 are respectively the rest length and the stiffness of the basis transverse springs. We assume the basis transverse springs are very stiff.

The Resulting Shape

We can prove that if:

1. the longitudinal springs have zero rest length
2. the transverse basis springs are quasi-rigid
3. the rest lengths of the transverse springs are defined by ${l_0}_t^i = {l_0}_t^1 \frac{(N+1-i)}{N}$
4. all other forces applied to the model have the same value, at least on the masses of each level,

then the balance shape of the points on each level i can be obtained from the balance shape of the basis points by a simple translation and a scaling by a factor of $(N + 1 - i)/N$. This implies that at rest:

- all the levels are planar (since the basis level is necessarily planar)
- all the levels are parallel to the basis level (i.e. to the underlying surface)
- the shape of all the levels is similar to the shape of the basis level
- each edge of each level is parallel to the homologous edges on other levels.

Such a resulting shape is displayed in Fig. 4. This proof is derived from the expression of the balance condition for different masses.

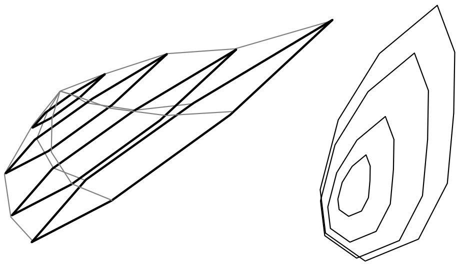

Figure 4: A qualitative representation of the droplet model's shape as a result of satisfying constraints **b**, **c**, and **d**. All levels have the same shape as the basis level.

3.3.2 Constraint a

In this work, we do not deal with the volume of the final spline-interpolated shape, but instead with the volume of the mechanical model (Fig. 4). Furthermore the elaboration of a general-purpose true-volume preserving dynamic link is beyond the scope of the present work. However we can prove that if the four stated conditions are satisfied, then the volume of the mechanical model can be precisely computed as:

$$V_{tot} = \frac{1}{3N^2} S \sum_{i=1}^{N+1} \alpha_i h_i \qquad (1)$$

where S is the area of the basis level, h_i is the distance between the ith level and the basis level, and the α_i are constant parameters depending only on the value of i and N.

We have observed that the area S remains practically constant during the deformations. In this case, the volume of the droplet remains constant if and only if the weighted sum in Eq. 1 remains constant. To preserve the weighted sum of heights at a given value, we introduce a specific dynamic link more complex than a simple spring. Let us call it the *solco* (as in *sum-of-length-constant*) connection.

The *Solco* Connection

The *solco* connection consists of a set of links connecting point-masses. Each *solco* is characterized by L links. Each link i connects two point-masses (A_i, B_i). It is also characterized by a stiffness k_s, a viscosity z_s, and a rest sum-of-lengths sol_0.

At each time step, the *solco* computes the value sol from the weighted sum of the lengths of all pairs $[A_i, B_i]$ of segments as

$$sol = \sum_{i=1}^{L} \alpha_i ||\overrightarrow{A_iB_i}||.$$

The force $\vec{F}_{A_i,B_i} = -\vec{F}_{B_i,A_i}$ applied by A_i on B_i is expressed by:

$$\vec{F}_{A_i,B_i} = \left(k_s(sol_0 - sol) - z_s \frac{d(sol)}{dt} \right) \frac{\overrightarrow{A_iB_i}}{||\overrightarrow{A_iB_i}||}$$

One of the elements (either A_i or B_i) can be a surface, provided that it is a fixed object. For instance if A_i is a point and B_i a surface, the distance between A_i and B_i will be the distance between A_i and the closest point on B_i. The applied force will always be perpendicular to the surface.

The Realization of Constraint a

We connect the point-masses of all levels i to the underlying surface by one *solco* connection. The weights of the *solco* connection will all be equal for the masses of each level. The weight associated with level i is the same as the weight of h_i in Eq. 1.

Note that the *solco* connection applies identical forces to all the point-masses of each level. So the four conditions of the previous section are still satisfied.

This guarantees that as soon as the weighted sum of heights is smaller than the specified rest value (sol_0), all *solco* links exert repulsive forces to increase the volume of the droplet, and when the weighted sum of heights is greater than sol_0 the links exert attractive forces to reduce the volume.

Thus in practice the weighted sum of heights at rest equals the specified value.

3.4 Deformations

The shape produced by the above model represents the average balance shape of a droplet placed in a force field and interacting with an underlying surface. However, due to other physical conditions, such as the spatial variation of the surface properties, the actual shape of the drops may be slightly different. In comparison with the global shape of the drop, this is a small-scale deformation. This is why we do not model it on the large-scale mechanical model, but on the final spline-interpolated shape model. We account for the above deformations by randomly displacing the spline control points. In the real world, this deformation depends on the importance of capillarity forces in comparison with the internal volume forces. For light drops, capillarity forces are dominant and this deformation can be large. For heavier drops, volume forces are dominant and the shape tends towards the average shape.

On the other hand, the very light drops have low velocity and tend to undergo smaller deformations in the course of time. A visually satisfactory compromise between these two tendencies consists of modeling the amplitude of this random variation as an increasing function

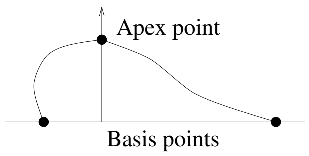

Figure 5: Two types of necessary shapes

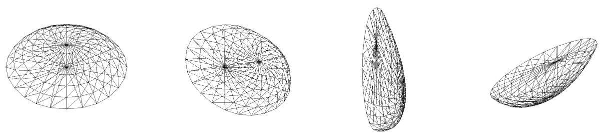

Figure 6: Droplet shapes for different orientations of the underlying surface. The segments are extracted from the spline patches.

of velocity (for low velocities) and as a decreasing function of velocity (for higher velocities). This amplitude is zero for droplets at rest and also for more rapidly moving (i.e. heavier) droplets.

3.5 Results

One of the great interests of this shape model is that through the control of the relative importance of constraints **a**, **b** and **c**, and more particularly the value of sol_0, it is possible to inject more or less liquid in the droplets and obtain heavier or lighter droplets. But the main point of constraint modeling is that as long as the four conditions mentioned above are satisfied, any change in a drop shape model can only lead to another drop shape model.

Now let us describe the particular realization that produced our images.

The minimal value for M allowing independent deformations in all three dimensions is $M = 3$. The minimal value for N is 1. However, if we desire to have both types of shapes represented on Fig. 5, $N = 1$ is not sufficient as it requires a higher order surface.

The images of Fig. 6 were obtained with the minimal values: $N = 2$ and $M = 3$. Each balance position calculation requires 292 simulation steps computed in 0.108 seconds on an SGI Indy R5000.

4 Rendering

The surface of most liquids is highly specular. This results in sharp highlights and reflections of the scene on its surface. The highlights simply correspond to a large value of the roughness coefficient n in Blinn's specular reflection model $(N \cdot H)^n$ [1]. Assuming droplets are very small with respect to the scene, a simple and efficient environment map [12] has proven to lead to satisfying results when mirror reflection must be considered.

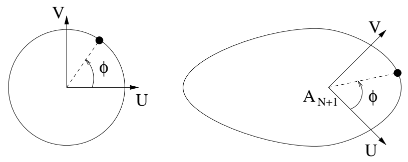

Figure 7: Applying an illumination texture at the base of a droplet

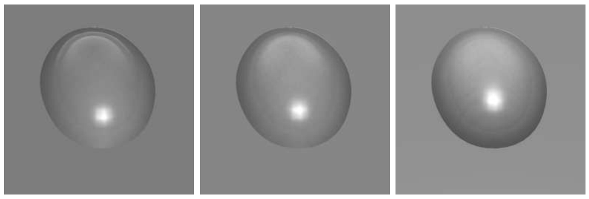

Figure 8: Illumination over and inside a droplet

The concentration of light due to the refraction in transparent liquids is more difficult to handle properly. Tracing rays from the light [17] offers an interesting avenue in the specific case of direct illumination of a single droplet. However the computations involved would prevent us from achieving reasonable rendering times. Because estimating the distribution of refracted illumination for any shape is extremely difficult, it becomes possible to consider even a very rough approximation of the distribution, as long as it exhibits a consistent behavior.

We computed a texture with a Monte Carlo ray tracer to capture the illumination at the base of a half-sphere for a set of incidence light directions θ from the base surface normal. We orient the U axis of the texture with the projected incoming light radial direction ϕ. The textured circle associated with a light direction is transformed to fit the contour of a droplet base, as illustrated in Fig. 7. The projected apex point A_{N+1} defines the new origin. Intermediate incidence angles are computed by interpolating between two illumination textures. Although inexact, the consistency and presence of illumination inside the droplet provide a more complete visual representation for transparent droplets at a reasonable cost. A droplet illuminated from three different light directions appears in Fig. 8.

The path followed by a droplet might affect the visual aspect of the surface. Kass and Miller [18] simply darken the region under the droplet. Specularity and controled surface normals along the edges of streaks also provide a better rendering of a streak.

5 Animated Sequences

Various animated sequences have been computed and are available through the site associated with this paper on `www.iro.umontreal.ca/labs/infographie/papers`

Droplets of Different Masses
: Several droplets of varying masses (ratio from 1 to 4) flow down a slanted polygon. Some of these droplets merge, and change velocity. Streaks are not drawn but the changes of roughness along their paths are taken into account.

Droplets of Different Viscosities
: Droplets flow down three slanted rectangles. Viscosity is set to 1, 10, and 100 for the left, center, and right rectangles, respectively. Notice the general differences of velocity.

Different Adhesions
: Regularly spaced droplets of the same mass flow down a sphere. Notice on the animated segments how increasing the adhesion forces between 0 to 10 determines at which angle the droplets detach from the surface.

Different Roughnesses
: Surface roughness is increased from the left to the right rectangles such that some of the small droplets in the right barely move. Notice in the red frame what happens. The top droplet suddenly accelerates when it reaches the less rough surface due to the streak (invisible) left behind by a previous droplet. The large droplet quickly reaches the smaller one, merges with it, but its velocity then appears to decrease due to the fact that the streak stopped there.

A Tear and More
: A red tear rolls down a mask made of 2000 triangles and leaves the surface when reaching below the chin. The streak left behind displays the path followed.

: More droplets roll down the mask. When they reach discontinuities of adjacency, such as eyes and lips, the droplet falls in free space. We did not detect collisions between a falling drop and the mask.

Different Shapes for a Droplet
: The force applied on a droplet goes from perpendicular, to left, and to right. The resulting shapes are displayed from the side and the top views.

6 Conclusion

The model presented in this paper offers several advantages. By separating the motion from the shape, we can simulate each phenomenon at a desired level of precision.

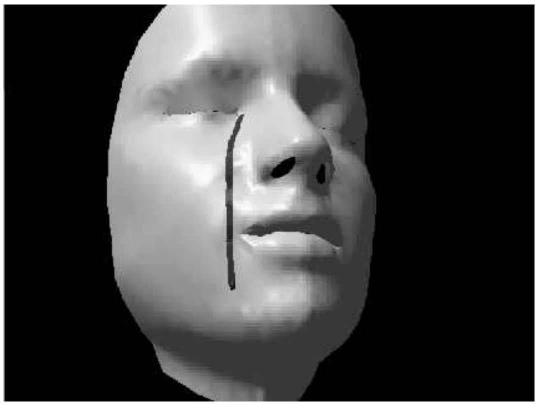

Figure 9: A droplet and its streak

The motion of a droplet is affected by surface roughness, adhesion forces, and friction. We introduce models to approximate each of these properties. While collisions between moving droplets could be handled exactly instead of at each computed frame, we considered the added computations and complexity were not essential as the resulting animations were already satisfying. However we could easily detect collisions at more intermediate frames than displayed, or introduce a threshold velocity below which the complete collisions should be computed. The streaks left behind droplets add much information about the motion and are visually important.

The droplets are currently displayed using simple shading and textures in OpenGL or RenderMan. This is quite acceptable for many droplet shapes. However the refraction inside the droplet is not currently handled in these rendering algorithms.

Our shape model produces various types of droplet shapes, all satisfying the characteristic properties. A further challenge would be to satisfy them during topology changes. This could be realized by replacing the springs by Lennard-Jones type interactions. Moreover, future work would include the elaboration of a deterministic model for small-scale deformations. This model would be based on the properties of the underlying surface. While this model is acceptable for the motion and shape of independent droplets, much still remains to be done on the deformations produced during the merging of droplets. Hydrodynamics within the droplet must then be considered and the computations involved will certainly increase.

Acknowledgements

We acknowledge financial support from Taarna Studios, NSERC, the Rhônes-Alpes region, and FCAR. We thank also Taarna Studios for its continuous support, and Cyriaque Kouadio for his illumination textures.

References

[1] J.F. Blinn. Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77 Conference Proceedings*, pages 192–198, July 1977.

[2] P. Borrel and D. Bechmann. Deformation of n-dimensional objects. *International Journal of Computational Geometry and Applications*, 1(4):427–453, 1991.

[3] J.X. Chen, N. da Vitoria Lobo, C.E. Hughes, and J.M. Moshell. Real-Time fluid simulation in a dynamic virtual environment. *IEEE Computer Graphics & Applications*, 17(3):52–61, May–June 1997.

[4] N. Chiba, S. Sanakanishi, K. Yokoyama, I. Ootawara, K. Muraoka, and N. Saito. Visual simulation of water currents using a particle-based behavioural model. *Journal of Visualization and Computer Animation*, 6(3):155–172, July 1995.

[5] M. Desbrun and M.-P. Gascuel. Animating soft substances with implicit surfaces. In *SIGGRAPH '95 Conference Proceedings*, pages 287–290, Aug. 1995.

[6] D. D'Humiéres, P. Lallemand, and U. Frisch. Lattice gas models for 3d hydrodynamics. *Europhys. Letters*, 2:291–297, 1986.

[7] J. Dorsey, H.K. Pedersen, and P. Hanrahan. Flow and changes in appearance. In *SIGGRAPH '96 Conference Proceedings*, pages 411–420, Aug. 1996.

[8] N. Foster and D. Metaxas. Realistic animation of liquids. In *Proceedings of Graphics Interface '96*, pages 204–212, Aug. 1996.

[9] N. Foster and D. Metaxas. Modeling the motion of a hot, turbulent gas. In *SIGGRAPH '97 Conference Proceedings*, pages 181–188, Aug. 1997.

[10] A. Fournier and W.T. Reeves. A simple model of ocean waves. In *SIGGRAPH '86 Conference Proceedings*, pages 75–84, Aug. 1986.

[11] U. Frisch, B. Hasslacher, and Y. Pomeau. Lattice-gas automata for the navier-stokes equation. *Phys. Rev. Lett.*, 56:1505–1508, 1986.

[12] N. Greene. Applications of world projections. In *Proceedings of Graphics Interface '86*, pages 108–114, May 1986.

[13] A. Habibi, A. Luciani, and A. Vapillon. A physically-based model for the simulation of reactive turbulent objects. In *Winter School of Computer Graphics 1996*, Feb. 1996.

[14] A. Habibi. *Modèles physiques supports de la relation mouvement-forme-image*. Ph.D. thesis, ACROE/CLIPS - Institut National Polytechnique de Grenoble, Jan. 1997.

[15] G. Hégron. Rolling on a smooth bi-parametric surface. In *Eurographics Workshop on Animation and Simulation*, pages 205–213, 1991.

[16] S. Jimenez and A. Luciani. Animation of interacting objects with collisions and prolonged contacts. In *Proceedings of the IFIP WG 5.10 Working Conference*, Apr. 1993.

[17] H.W. Jensen. Rendering caustics on non-lambertian surfaces. *Computer Graphics Forum*, 16(1):57–64, 1997.

[18] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *SIGGRAPH '90 Conference Proceedings*, pages 49–57, Aug. 1990.

[19] A. Luciani, S. Jimenez, C. Cadoz, J.L. Florens, and O. Raoult. An unified view of multitude behaviour, flexibility, plasticity and fractures balls, bubbles and agglomerates. In *Proceedings of the IFIP WG 5.10 Working Conference*, Apr. 1991.

[20] A. Luciani, A. Habibi, A. Vapillon, and Y. Duroc. A physical model of turbulent fluids. In *Computer Animation and Simulation '95*, pages 16–29, Sept. 1995.

[21] A. Luciani, S. Jimenez, J. L. Florens, C. Cadoz, and O. Raoult. Computational physics: A modeler-simulator for animated physical objects. In *Eurographics '91*, pages 425–436, Sept. 1991.

[22] N.L. Max. Vectorized procedural models for natural terrain: Waves and islands in the sunset. In *SIGGRAPH '81 Conference Proceedings*, pages 317–324, Aug. 1981.

[23] G. Miller and A. Pearce. Globular dynamics: A connected particle system for animating viscous fluids. *Computers and Graphics*, 13(3):305–309, 1989.

[24] N. Nishikawa and T. Abe. Artificial nature in splash of droplets. In *Compugraphics '91*, vol. I, pages 457–466, 1991.

[25] J. Nouiri, C. Cadoz, and L. Luciani. The physical modelling of complex physical structures: The mechanical clockwork, motion, image and sound. In *Computer Animation 96*, June 1996.

[26] J.F. O'Brien and J.K. Hodgins. Dynamic simulation of splashing fluids. In *Computer Animation '95*, pages 198–205, Apr. 1995.

[27] D.R. Peachey. Modeling waves and surf. In *SIGGRAPH '86 Conference Proceedings*, pages 65–74, Aug. 1986.

[28] W.T. Reeves. Particle systems – A technique for modeling a class of fuzzy objects. *ACM Trans. on Graphics*, 2:91–108, Apr. 1983.

[29] D. Terzopoulos, J. Platt, and K. Fleischer. Heating and melting deformable models (from goop to glop). In *Proceedings of Graphics Interface '89*, pages 219–226, June 1989.

[30] D. Tonnesen. Modeling liquids and solids using thermal particles. In *Proceedings of Graphics Interface '91*, pages 255–262, June 1991.

[31] P.Y. Ts'o and B.A. Barsky. Modeling and rendering waves: Wave-tracing using beta-splines and reflective and refractive texture mapping. *ACM Trans. on Graphics*, 6(3):191–214, 1987.