Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Modeling Ice Dynamics As A Thin-Film Stefan Problem

Theodore Kim* David Adalsteinsson† and Ming C. Lin*

*Department of Computer Science, University of North Carolina at Chapel Hill, U.S.A.
†Department of Mathematics, University of North Carolina at Chapel Hill, U.S.A.

Abstract

Large, 3D ice formations such as icicles exhibit a high degree of geometric and optical complexity. Modeling these features by hand can be a daunting task, so we present a novel physically-based algorithm for simulating this phenomenon. Solidification is usually posed as a so-called 'Stefan problem', but the problem in its classic form is inappropriate for simulating the ice typically found in a winter scene. We instead use the 'thin-film' variant of the Stefan problem to derive velocity equations for a level set simulation. However, due to the scales involved in the problem, even an adaptive grid level set solver is still insufficient to track the tip of an icicle. Therefore, we derive an analytical solution for the icicle tip and use it to correct the level set simulation. The results appear to be in agreement with experimental data. We also present a physically-based technique for modeling ripples along the ice surface that alleviates the need to explicitly track small-scale geometry. To our knowledge, our approach is the most complete model available, and produces complex visual phenomena that no previous method has been able to capture.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]:

1. Introduction

The dense optical and geometric complexity of ice adds appeal and authenticity to any winter scene. Like wind and fire, ice is considered elemental, so it is useful as a dramatic tool in visual effects. Large scale ice formations have appeared in recent films such as *The Incredibles* and *The Lion, the Witch and the Wardrobe*. In the former, the icicles were modelled as smooth, stylized cones, and in the latter, they had to be hand-molded from plaster and clay. In the absence of a computational model for solidification, modeling realistic ice formations by hand can be a daunting task. This task becomes even more demanding if an animation of the ice forming is required, as it is unclear what the dynamics of the ice surface should be.

Recent research efforts have found much success in simulating fluids (see e.g. [SRF05] and [FOK05] for recent results), but these algorithms are designed to capture features such as vorticity and splashing. Freezing is instead characterized by sharp icicle tips, and the optical effects caused by surface rippling. The underlying pattern formation mechanisms are different, so new algorithms are needed. While some recent work has addressed the problem of ice formation [KL03,KHL04], it dealt with small-scale, essentially 2D formations such as frost on surfaces or snowflakes. The techniques described therein assume that the ice crystal is surrounded by a large bath of water, and while an argument can be made that this is the case in 2D, it is clearly not the case in 3D. This makes their extension to the complex 3D cases we present here difficult.

Simulating large scale ice formation is a challenging task due to the wide range of scales involved. An interesting ice formation is on the scale of roughly 1 meter, but the tip of an icicle is roughly 2 millimeters in radius, and a thin water film coats the ice that is on the order of tenths of a millimeter in thickness. Surface tension forces in this thin film drive the formation of sharp icicle tips, and also cause the formation of ripples along the icicle surface. Therefore, in order to convincingly simulate the formation of large scale features, features that span four orders of magnitude must be accurately resolved. In this paper, we present a method that captures these various multi-scale phenomena in a single unified simulation.

Phase transition problems are usually stated as a *Stefan Problem*. Usually this problem examines how ice forms in an infinite supply of water. We are instead interested in the case where the water supply is severely limited. Therefore,

we present a 'thin-film' version of the Stefan problem and design a novel method for solving this problem efficiently.

Features on the ice surface frequently merge, so we have elected to use level set methods for the overall simulation [OF03, Set99]. However, even adaptive mesh techniques such as the octree level set solver in [LGF04] cannot provide the resolution needed to resolve surface tension forces that are four orders of magnitude smaller than the overall domain, so we describe methods of tracking small-scale features separately. We derive an analytical solution for the dynamics of the icicle tip, as well as a curvature-dependant evolution equation for the ice far from the tip. In order to avoid explicitly tracking a large amount of ripple geometry, we modify an analytical model from physics that poses surface ripples as a Fourier mode along the ice surface. We use this method to defer explicit instantiation of the ripple geometry to render time.

Our contributions are as follows:

- A level set approach to the thin-film Stefan problem;
- An analytical solution for the tip of an icicle that appears to be in agreement with experimental data;
- A non-linear, curvature-driven evolution equation for the ice front far from the icicle tip;
- A physically-based method for simulating surface ripples that avoids the need to track small scale geometry in the simulation;
- A unified simulation framework for modeling complex ice dynamics.

We verify the results of our simulation numerically with experimental data and visually with photographs. We also demonstrate our algorithm on several scenarios, including natural 3D icicle formation, an ice sculpture, and a freezing fountain scene. Our approach is quite efficient, with each step taking an average of 2.5 to 12 seconds and the total simulation running times ranging between 5 and 30 minutes.

2. Previous Work

Various forms of phase transition have attracted recent attention in computer graphics. Carlson et al. [CMIT02] and Wei et al. [WLK03] respectively presented MAC and Lattice-Boltzmann based methods for simulating melting. Subsequently, Rasmussen et al. [REN*04] proposed an IMEX scheme for the viscous forcing terms, and Losasso et al. [LIGF05] described a method of melting Lagrangian solids into Eulerian liquids.

However, solidification has not been as well examined. To the best of our knowledge, the only work that bears some resemblence to ours is [KG93]. They described a random-walk model of icicle growth, where water droplets walk along an ice surface and freeze with a certain probability. However, their approach does not naturally handle the formation of more than one icicle. More seriously, the notion that icicles form as the aggregate of discrete droplets is not empirically supported. Kim et al. [KL03, KHL04] presented ice formation algorithms for relatively small-scale 2D growth on 3D objects. As described earlier, their approaches cannot generate the structures in this paper.

In physics and glaciology, there exists some work on the problem of thin-film ice growth. Several analytical models exist for icicle formation, such as [MMN*94, Mak88, SL94], but these models are concerned with accurately capturing the ratio of an icicle's length to its radius. They are not directly applicable to visual simulation, as they would merely generate simple cones and cannot capture surface rippling effects. Thin film ice formation is also a topic of interest in mechanical engineering, as ice forming on the wing of an aircraft is a hazardous scenario. The Messinger model [Mes53] is the standard method of determining when ice will form, but is an energy balance model that is also not directly applicable to visual simulation. Myers and Hammond [MH99] recasted the problem as a thin film Stefan problem, but only solved the 1D case.

Until very recently, the physics of ripple formation on crystal surfaces was a poorly understood phenomenon. However, Ogawa and Furukawa [OF02] recently proposed a model of ripple formation which was subsequently refined by Ueno in a pair of articles [Uen03, Uen04]. The model from [OF02] only applies to a cylinder, and Ueno's model was derived for an inclined plane. We will later modify Ueno's model for our simulation.

3. The Stefan Problem

3.1. Background

In math and physics, solidification is usually posed as a Stefan problem. First posed by Josef Stefan as a model of ocean ice forming in arctic regions, the Stefan problem has since found applications in fields ranging from geology to metallurgy. The richly non-linear behavior of the problem has also attracted considerable interest in mathematics [Hil87, Mei92]. An excellent historical overview of the Stefan problem is available in [Wet01].

There are only a handful of known closed form solutions to the Stefan problem, and these only apply to simple geometries. Stefan originally solved the planar case, and subsequently the case of a sphere [Fra49] and a parabola [Iva47] were derived. These cases are often referred to eponymously as the "Frank sphere" and "Ivantsov parabola" solutions. We will later base some of our thin film equations on these solutions.

Due to its ability to handle geometry with rapidly changing topology, level set methods have found recent success in solving the Stefan problem numerically. The method was first applied in [SS92]. More recently, second [GFCO03] and fourth order [GF05] approaches were proposed. All of this

work dealt with the classical Stefan problem. We are instead interested in applying level set methods to the thin film case, because it is more appropriate for modeling ice formation in natural scenes. While our work may at first appear related to that of [CMK*01], the 'thin-film' referred to in this case is 2D epitaxial growth on a semiconductor wafer, not a thin-film water supply.

3.2. The Classic Stefan Problem

The Stefan problem is composed of two simple equations. Assume we have a heat field T defined continuously over some computational domain, and an initial ice/water interface Γ. The heat field evolves according to the heat equation

$$\frac{\partial T}{\partial t} = D\nabla^2 T, \tag{1}$$

where D denotes a diffusion constant. The ice/water interface then evolves in the normal direction according to

$$\frac{\partial \Gamma}{\partial t} \cdot \mathbf{n} = D\frac{\partial T}{\partial \mathbf{n}}, \tag{2}$$

where $\mathbf{n}$ denotes the normal direction. Fluid velocity and the coefficient of expansion of ice are assumed to be negligible. Many different flavors of the Stefan problem exist that impose various boundary conditions on the heat field and interface. We select the *one-sided* Stefan problem as the most appropriate for our case. In this case the ice/water interface is assumed to be the freezing temperature of water, T_f, and the temperature of the fluid infinitely far from the interface is set to some undercooled temperature T_u, where $T_u < T_f$. Both of these assumptions are necessary if the crystal is to grow. If the temperature at the crystal surface were greater than T_f, phase transition would not occur, and if the temperature of the fluid were not lower than T_f, then by Eqn. 2, no growth would occur.

We are dealing with an overall timescale on the order of hours, so we can assume that the heat field is essentially in equilibrium. Eqn. 1 then simplifies to the Laplace equation

$$\nabla^2 T = 0. \tag{3}$$

This quasi-steady state approximation and the boundary conditions just described are common to all the existing models from glaciology [MMN*94, Mak88, SL94] and physics [OF02, Uen03] as well.

3.3. The Thin Film Stefan Problem

Unlike the classic Stefan problem, we want to model the situation where a thin film of water continuously coats the outside of the ice. We assume that the crystal surface is at freezing temperature T_f, but instead of specifying the undercooled temperature T_u at some infinitely far away boundary, we specify it at a small offset δ from the interface. More formally, we specify this as

$$T(\Gamma + \delta(\Gamma \cdot \mathbf{n})) = T_u. \tag{4}$$

Using this modified boundary condition, we can derive thin-film evolution equations that can be solved using level set methods.

The simplest case is an evolving planar interface. In 1D, Eqn. 4 simplifies to $\Gamma + \delta = T_u$. The Laplace equation is easily integrated in this case, and by inserting the result into Eqn. 2, we obtain a constant velocity for the planar case:

$$\frac{d\Gamma}{dt} = D\frac{T_u - T_f}{\delta}. \tag{5}$$

This planar solution matches the one obtained in [Uen03].

The cylindrical solution can be obtained in a manner similar to the classical solution described in [Hil87]. For the cylindrical case, we must instead solve the polar Laplace equation,

$$D\frac{\partial^2 T}{\partial r^2} + \frac{D}{r}\frac{\partial T}{\partial r} = 0, \tag{6}$$

where r is the radial coordinate. We assume the crystal is growing in the positive r direction, and define the current interface position as r' within this coordinate system. The exact method we use to transform Γ to r' will be discussed in Section 5.2. By applying the two boundary conditions to the polar Laplace, we obtain

$$T(r') = DT_f + D\frac{T_u - T_f}{\log\left(\frac{r'+\delta}{r'}\right)}(\log r - \log r'). \tag{7}$$

We want a velocity equation at r', so we take the derivative of Eqn. 7, solve for $r = r'$, and substitute the result into Eqn. 2 to obtain:

$$\frac{\partial \Gamma}{\partial t} = D\frac{\partial T(r')}{\partial r} = D\frac{T_u - T_f}{r' \log\left(\frac{r'+\delta}{r'}\right)}. \tag{8}$$

We can obtain a similar solution for the negative curvature case. This corresponds to the case where the ice freezes radially inwards to fill in a cylindrical hole. Following steps similar to those above, we obtain

$$\frac{\partial \Gamma}{\partial t} = -D\frac{T_u - T_f}{r' \log\left(\frac{r'-\delta}{r'}\right)}. \tag{9}$$

Eqns. 8 and 9 can be consolidated into a single velocity equation,

$$\frac{\partial \Gamma}{\partial t} = D\frac{T_u - T_f}{|r'| \log\left(\frac{|r'|+\delta}{|r'|}\right)}. \tag{10}$$

This is the equation that the level set solver solves. We note this equation implicitly includes Eqn. 5 as well. The planar case corresponds to the case of a cylinder of infinite radius,

and in the limit

$$\lim_{r'\to\infty} D\frac{T_u - T_f}{|r'|\log\left(\frac{|r'|+\delta}{|r'|}\right)} = D\frac{T_u - T_f}{\delta}, \qquad (11)$$

the planar velocity equation is retrieved. Therefore, Eqn. 10 describes the interface velocity of cylinder with positive and negative curvature, as well as a plane. We have not encountered this equation elsewhere in the literature, so to our knowledge, it is novel.

On a high level, Eqn. 10 induces faster growth in regions of high curvature. While it would be simpler to only use the planar solution, we found that this produces unnaturally sharp corners in the results. The curvature-dependant term smooths away these corners.

3.4. The Thin Film Ivantsov Parabola

In this section we will derive a solution to the thin-film Stefan problem for parabolic geometry. The paraboloid solution drives icicle growth, so it is crucial that it be solved accurately. From a visual simulation standpoint, correctly capturing the velocity of an icicle tip is important because it determines the overall shape of the icicle. If the velocity is too slow, we will get unconvincingly stubby icicles, and if it is too fast, we will get equally unconvincing needles.

A brute force solution would directly simulate the water flow along the surface of the icicle, and correctly model the surface tension forces that give rise to pendant drops, which in turn form icicle tips. However, the water layer along the icicle wall is 0.1 mm in thickness, so tracking this feature inside a 1 meter^3 cube would require at least a 10000^3 grid. Even with an octree solver, this resolution is intractable.

Fortunately, there is a simple solution. The experimental measurements in [MMN*94] indicate that across a wide range environmental conditions, the radius of the tip of an icicle remains fixed at approximately 2.5 mm. By inserting this experimental value into an analytical solution for the growing icicle tip, we can correct the signed distance function at every timestep. We are able to solve for the dynamics of the icicle tip independently because the physics only depend on three *local* factors: the curvature, the ice and air temperatures, and the radius of curvature of the icicle tip. We have found that by using this method, we can obtain satisfactory results on a much coarser 256^3 grid. In practice, we actually used an octree with a maximum depth of 8, corresponding to at most a 256^3 grid.

In crystal growth, the Ivantsov parabola solution is often used to model the growing tip of a dendrite. We conjecture that icicles are the thin film analogs of classic Stefan problem dendrites. There are several different methods of obtaining the Ivantsov parabola solution, but we model our thin-film derivation after the derivation given in [Sai96]. Crucial steps, such as Eqn. 14, and the dual definition of the Peclet number, mirror steps from that derivation. Assume that a parabolic

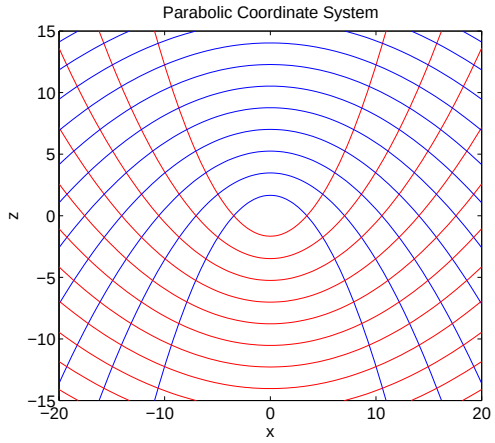

Figure 1: ***2D slice of parabolic coordinate system:*** *Red lines are parabolas of constant* ξ*, and blue lines are constant* η*. Note how the distance betwen adjacent lines increases far from the tip.*

crystal is growing in the z direction with a velocity V. In this case the z direction corresponds to the direction that points toward the ground. We define a moving frame $z' = z - Vt$ so that at time t, $z' = 0$ always denotes the current position of the parabola tip. We then define a parabolic coordinate system

$$\begin{aligned} \xi &= r - z' \\ \eta &= r + z' \\ \theta &= \arctan(x/y), \end{aligned}$$

where $r = \sqrt{x^2 + y^2 + z'^2}$. Intuitively, ξ and η each define a paraboloid in space, and their intersection forms a circle. The θ coordinate then defines a point on this circle. (See Fig. 1)

We represent the overall interface as a paraboloid η', which is defined as the η that contains the current tip z'. The Laplace equation in parabolic coordinates becomes

$$\frac{\partial}{\partial\eta}\left(\eta\frac{\partial T}{\partial\eta}\right) + \frac{1}{l_D}\left(\eta\frac{\partial T}{\partial\eta}\right) = 0, \qquad (12)$$

and Eqn. 2 simplifies to

$$\frac{\partial T}{\partial\eta} = \frac{1}{l_D}, \qquad (13)$$

where l_D denotes the diffusion length. It appears that this equation no longer describes a velocity, but it is implicit in the l_D term. The thin film boundary conditions can then be stated as $\eta' = T_f$ and $\eta' + \delta = T_u$. Due to the parabolic coordinate system, the second boundary condition is only meaningful near the tip. As shown in Fig. 1, the normal distance between two adjacent values of η increases as the distance from the tip increases. Therefore, far from the tip, the distance between the two will be much greater than δ. Fortu-

symbol	definition	value
T_f	freezing temperature	0^oC
T_u	undercooled temperature	-4.9^oC
L	latent heat of fusion	3.3×10^8 J/m^3
D	thermal diffusion constant	1.3×10^{-7} m^2/s
C_p	specific heat	4.2×10^6 J/(Km^3)
h_0	thickness of water layer	10^{-4} m
k	wavenumber	600
n	solid over liquid thermal conductivities	3.92
ε	initial ripple amplitude	1×10^{-4} m
σ_r	ripple amplification rate	5.2×10^{-4} s^{-1}
v_p	ripple translational velocity	-6.1×10^{-7} m/s

Table 1: *Symbols and values. Values are from [Uen03].*

nately, since we are only interested in values near the tip, far away inaccuracies are irrelevant. Using these boundary conditions, the parabolic Laplace equation integrates to

$$T(\eta) = \frac{C_p(T_u - T_f)}{L}\left(1 - \frac{\int_{\eta'}^{\eta} \frac{e^{-\frac{\eta}{l_D}}}{\eta} d\eta}{\int_{\eta'}^{\eta'+\delta} \frac{e^{-\frac{\eta}{l_D}}}{\eta} d\eta}\right). \quad (14)$$

where L is the latent heat, and C_p is the specific heat. Inserting this result into Eqn. 13, we obtain

$$\begin{aligned}\frac{C_p(T_u - T_f)}{L} &= \frac{\eta'}{l_D} e^{\frac{\eta'}{l_D}} \int_{\eta'}^{\eta'+\delta} \frac{e^{-\frac{\eta}{l_D}}}{\eta} d\eta \\ &= \frac{\eta'}{l_D} e^{\frac{\eta'}{l_D}} \left(\int_{\eta'}^{\infty} \frac{e^{-\frac{\eta}{l_D}}}{\eta} - \int_{\eta'+\delta}^{\infty} \frac{e^{-\frac{\eta}{l_D}}}{\eta}\right) d\eta.\end{aligned} \quad (15)$$

By applying exponential integral notation $E_1(P) = \int_P^{\infty} \frac{e^{-x}}{x} dx$ and a change of variables using the Peclet number $P = \frac{\eta'}{l_D}$, this equation can be rewritten as

$$\frac{C_p(T_u - T_f)}{L} = P\left(e^P\right)\left(E_1(P) - E_1\left(P + \frac{\delta}{l_D}\right)\right). \quad (16)$$

Velocity can be solved for by substituting the identity $P = \frac{\eta' V}{2D}$, where V is the tip velocity and D is the thermal diffusitivity. Like the classic Ivantsov parabola solution, Eqn. 16 is difficult to integrate explicitly, so we solve it numerically. Similar to the method used in the classic case, we first approximate $E_1(P)$ with its Puiseux series

$$E_1(P) = \gamma + \ln P + \sum_{n=1}^{\infty} \frac{(-1)^n P^n}{n!n}, \quad (17)$$

where γ is the Euler-Mascheroni constant, and then solve for V using Newton iteration.

4. A Ripple Formation Model

The equations we have presented so far will produce sharp icicle tips, and smooth features far from the tip. However, other non-smooth features occur in ice formations. This is most visible as ripples along the surface of an icicle. In this section, we present a method of simulating these features.

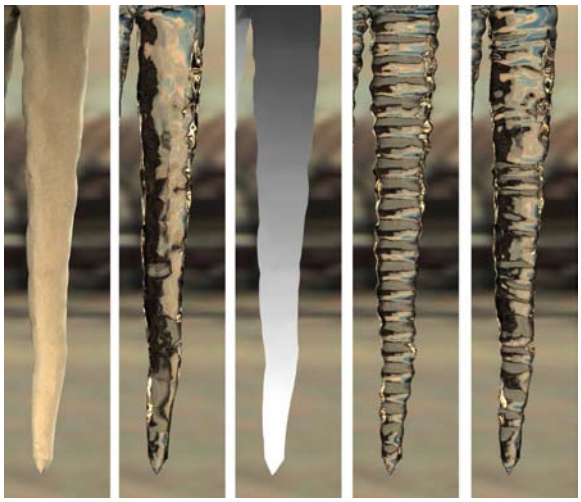

Figure 2: ***Rendering stages. Left to right:*** *1. Matte shaded level set results. 2. Ray traced level set results. Note the unnatural smoothness of the surface. 3. Recorded arrival times. Whiter regions denote later arrival times. Note how the time increases near the tip. 4. Ray traced results obtained by inserting the arrival times into the Ueno model. Note the unnatural uniformity of the ripples. 5. Final render using velocity and amplitude perturbation.*

Until recently, the formation of these features was poorly understood. Pattern formation of this type is usually explained in terms of Mullins-Sekerka theory [MS64], but Mullins-Sekerka theory predicts the formation of patterns at many wavelengths, whereas experiments show that ice ripples only form at a wavelength of roughly 1 cm. Recently, [Uen03] showed that one of the elements of Mullins-Sekerka theory, the Gibbs-Thomson effect, does not apply in the case of ice ripple formation, and proposed an alternate model that is in excellent agreement with experimental data.

The Ueno model can be stated in one dimension as

$$u(x,t) = \varepsilon e^{\sigma_r t} \sin(k(x - v_p t)), \quad (18)$$

where x and t are spatial and temporal coordinates, ε is the amplitude of the initial ripple, k is a wavenumber, σ_r is an amplification rate, and v_p is a translational velocity. Intuitively, we can think of Eqn. 18 as a sine wave being transported along the ice surface, which amplifies in time by a factor σ_r and climbs up the length of the icicle with a velocity v_p. The representation is attractive because it allows us to track just a single 'creation time' scalar during the simulation, and leave the instantiation of ripple geometry to the renderer.

If we use Eqn. 18 directly in our simulation, we will obtain unnaturally symmetric ripples, as can been seen in the

second to last image of Fig. 2. This is because in the Ueno model, the translational velocity v_p is assumed to be constant. A brute force method of introducing more visual variety would be to take the derivative of Eqn. 18 and integrate it at every grid point, at every timestep. This would impose a small timestep restriction on the simulation, and since we are dealing with timescales on the order of hours, we would like to avoid such a restriction.

Instead, we observe that the v_p variable can be interpreted as the average translational velocity of the interface over the lifetime of the ripple. Various environmental conditions cause this average velocity to fluctuate over time, so we can imitate this physical noise using numerical noise. We elect to use an easily controlled Perlin noise function with a 1 cm wavelength. At render time, each vertex does a lookup into a 3D Perlin noise function and uses it to jitter v_p. The result of this simple perturbation can be seen in the final image of Figure 2. In addition to avoiding a timestep restriction, this approach decouples the small scale detail almost entirely from the level set simulation. When designing ice patterns, the small scale ripple details can then be tweaked without having to rerun the simulation.

5. A Level Set Solver

5.1. Background

We will now describe how to solve the equations from the previous sections using level set methods [OF03, Set99]. Level set methods can simulate interfaces with rapidly changing topology by embedding the interface as an isosurface in a higher-dimensional function, which is usually a signed distance function ϕ. The function is then evolved according to the equation

$$\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0, \tag{19}$$

where $\mathbf{v}$ is some velocity field. Because the ice interface often merges, we have decided to use level set methods in this work. We specifically use the narrow band level set method [AS95], where the narrow band is tracked using an unbalanced octree, much as in [LGF04]. We use fifth order HJ-WENO for the spatial derivatives and second order TVD RK for timestepping (see [OF03] for details). A hybrid particle level set method [EFFM02] has recently been successful in simulating the Navier-Stokes equations because it uses Lagrangian particles to re-introduce smeared out small scale detail. In our case, we have captured the small scale detail using alternate methods, so a basic level set solver suffices.

5.2. The Velocity Field

In order to evolve an existing ice interface, we must specify a velocity $\mathbf{v}$. At each grid point, we choose to approximate the interface as locally cylindrical, and use Eqn. 10 to compute the velocity in the normal direction. In order to do this, we first compute the maximum principal curvature at each grid point by computing the Gaussian curvature K and the mean curvature H. The maximum curvature is then the larger root κ of the quadratic $\kappa^2 - 2H\kappa + K = 0$ (see [Set99, OF03] for details). This κ describes the largest osculating cylinder at that point, so we plug this value into r' from Eqn. 10. The resultant radial velocity is then dotted against the normal direction, giving us the final velocity at that grid point.

Eqn. 10 is only defined along the interface, whereas we require velocities over the entire narrow band. Since curvature is defined over the entire domain, we use it to compute values for Eqn. 10 everywhere. This approach does not seem to distort the distance field much, so it works well in practice.

5.3. Inserting the Icicle Tips

In section 3.4, we derived the velocity of a parabolic icicle tip, with the goal of tracking these small scale paraboloids separate from the level sets and using them to correct the signed distance function. We now show how to perform this correction. The equation for a translating paraboloid pointing in the negative z direction is

$$z(x,y) = \frac{x^2}{2R} + \frac{y^2}{2R} - Vt, \tag{20}$$

where R is the radius of curvature, V is velocity, and t is time. In this case we use the radius of curvature of the icicle tip, which has been experimentally observed to be roughly 2.5 mm under a wide variety of conditions. We assume that the paraboloid is circularly symmetric, so we can instead solve the 2D case

$$z(x) = \frac{x^2}{2R} - Vt. \tag{21}$$

The squared distance from any point in space (p_x, p_z) to any point on this parabola is then defined as

$$S = (p_x - x)^2 + \left(p_z - \left(\frac{x^2}{2R} - Vt \right) \right)^2, \tag{22}$$

where S is the squared distance. If we want to then find the minimum distance to the parabola, we must find the zeros of the derivative of S,

$$\frac{dS}{dx} = -2(p_x - x) - \frac{2}{R} x (p_z - Vt - \frac{x^2}{2R}). \tag{23}$$

We find the roots of this equation numerically. The second derivative of S is very flat around the roots of interest, making Newton-Raphson a poor choice for a solver. Fortunately, we can obtain fairly tight bounds on the location of the root, making bisection method viable. Since a parabola is symmetric about the y axis, we can solve for the value of the root over the positive x domain (excluding $x = 0$) without loss of generality. Over this half space, Eqn. 23 can only have one root, corresponding to the point of minimum distance. If the point (p_x, p_y) is outside the parabola, then the root is guaranteed to be in the interval $\langle 0, p_x]$. If the point is inside the parabola, the root is guaranteed to be in the interval $\langle 0, \sqrt{2R(p_y + Vt)}]$, Using these bounds, we have found

that bisection method only requires a handful of iterations to converge.

With this method, we can correct the signed distance field of the level set solver. At every timestep, given the current position of a paraboloid, we compute the exact distance field values for a 4^3 neighborhood around the tip and overwrite the values in the level set distance field. Cases where the tip runs into an obstacle such as the ground can be handled by simply deleting the parabola and letting curvature-driven growth take over.

5.4. Tracking the Ripples

The last component of the level set solver is a ripple tracking method. In section 4, we described the ripples as a translating sine wave. We couple the ripple formation equation, Eqn. 18, to the level set solver using the time variable t.

The variable t in Eqn. 18 represents the length of time that a ripple has existed, not the overall time that the simulation has been running. In order to obtain this t, we need to track the creation time of each ripple. Since the icicle tips are the fastest moving features in the simulation, whenever the icicle tip solution is used to correct the signed distance function, we set the creation time in those grid cells to the current time. The initial ice front at the beginning of the simulation is given a creation time of $t = 0$.

We encounter the same problem when tracking the creation times that we did with the velocities; they are only defined along the interface. But, we need a method of ensuring that as the interface moves, the creation time moves with it. To accomplish this, we apply the method of fast extension velocities described in [AS99]. Instead of extending a velocity off of the front, we extend the creation time. An example of these tracked arrival times is shown in the middle image of Figure 2.

6. Rendering

We interface the level set solver with a renderer by performing marching cubes on the distance field, and sending the triangles to 3Delight, a RenderMan implementation. The creation time information is interpolated per vertex and sent to the renderer as well. A displacement shader then computes Eqn. 18, applies noise to the interface velocity, and generates the ripple geometry on a per pixel basis.

Ice presents a challenging rendering scenerio because refraction, reflection, and multiple scattering make up the bulk of the visual detail. The reflection and refraction components can be dealt with by using the two layer wetness model described in [JLD99].

Accounting for the multiple scattering effects is more difficult. An obvious choice is to use the dipole approximation from [JMLH01], but this model is not well suited to ice. The model assumes that the scattering medium is fairly homogeneous, but in the case of ice, the medium varies continuously from transparent at the surface to highly scattering at the core. It appears that even the recent work of [DJ05] cannot be applied, since it handles multiple discrete scattering layers, but not a continuum. Additionally, the dipole model approximates the surface as a semi-infinite plane, and this assumption breaks down at the sharp icicle tips. Solving these issues rigorously is beyond the scope of this paper. Instead, we will describe a method that provides reasonable visual results. Near the root of the icicle, the medium is sufficiently thick and the curvature sufficiently flat that the dipole approximation gives visually plausible values. Near the tips, the dipole approximation returns unnaturally dark values. Fortunately, we know that thin features usually denote newly created ice, which is nearly transparent. So, in addition to using the dipole approximation to render the ice, we also use it as a blending factor between the multiple scattering color and the purely refracted ray color.

7. Results and Validation

We have used our algorithm to simulate ice formation in several scenes, and also validated portions of our model against experimental data. The code was compiled using Intel Compiler 8, and the timings were obtained on a 3 GHz Pentium 4. All simulations take place on an bounded, unbalanced octree with maximum depth of 8, corresponding to a virtual 256^3 grid.

In Figure 5 we simulated ice forming on a fountain. The fountain was left running during a cold day, and the overflowing water froze into ice. The simulation averaged 12 seconds a timestep and completed in 30 minutes. In Figure 7 we simulated ice forming under a roof top. The simulation averaged 2.5 seconds a timestep and completed in 5 minutes.

In order to demonstrate the flexibility of our model, we simulated the 'icicle star' sculpture by internationally renowned artist Andy Goldsworthy [Gol90] in Fig. 4. Mr. Goldsworthy constructs sculptures from natural materials, in this case icicles. The formation of the icicle star could not occur naturally, because gravity forces a downwards water flow. However, our model implicitly allows water to flow in any direction by simply re-orienting the parabolic tips. For an animation of our 'zero-gravity' icicle-star growing, please see the supplementary video. The simulation averaged 5.1 seconds per timestep and was completed in 18 minutes.

We have not found Eqn. 16 elsewhere in the literature, so to test its validity we have compared it against experimental data. There is a limited amount of data available on the 3D ice growth, but [MMN*94] provides experimental data on icicle tip velocities under a range of undercoolings. In order to make a comparison to their data, we must select appropriate values for η' and δ. The η' value can be interpreted as the radius of curvature of the icicle tip. As

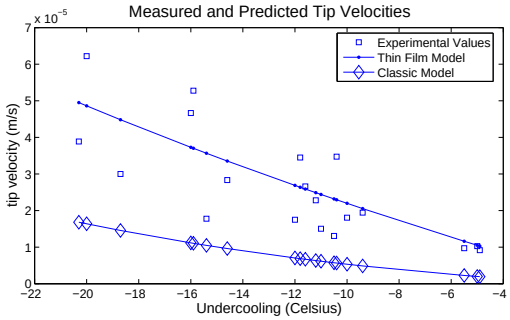

Figure 3: ***Experimental validation:*** *Our thin film model passes through the center of the data set, while the classic Ivantsov solution predicts much slower growth rates than those measured.*

stated earlier, experiments show that this value is 2.5 mm. The value of δ on the icicle surface is usually on the order of 1×10^{-4} m, but this value represents the water thickness along the icicle wall, not at the tip. At the tip, a pendant drop forms that is roughly the same radius as the underlying crystal. Therefore, we estimate the value of δ at the tip to be 2.5 mm as well. Fig. 7 shows how our predicted values compare to experimental data. We compute tip velocities over a variety of undercoolings, and compare our results to those of the classic Ivantsov relation. As stipulated by Eqn. 2, as the undercooling increases, the growth rate must increase as well. The classic solution predicts much slower growth and consistently undershoots the data. This is to be expected, because in the classic case, the temperature gradient at the tip has been 'stretched' by the infinitely far away boundary condition. While this experimental data set appears to be quite noisy, our solution appears to be in fair agreement, and we have found that it generates visually convincing results as well.

8. Summary and Future Work

We have presented an efficient physically based method for simulating 3D ice formations that are typically found in winter scenes. The model is, to our knowledge, the most complete approach currently available.

The most natural direction for future work is the integration of our model with melting and combustion simulations, creating a unified approach that can visually capture all three common states of matter and all the phase transitions in-between. This would also allow, for example, the droplets from the icicle tips to be simulated. The effects of an explicit flow simulation can be incorporated into the current ice simulation by allowing the δ and T_u variables to vary locally according to the flow. However, correctly resolving all these factors introduces some very challenging scale disparities in both space and time.

As previously mentioned, from a rendering standpoint, we know of no algorithm that can efficiently render the heavily inhomogeneous scattering medium that ice represents, so this also presents an interesting future work direction.

On the physics side, while Mullins-Sekerka theory predicts the formation of dendrites in an infinite bath, there is no equivalent theory for the thin film case. Such a theory would predict the locations of icicle initiation, allowing the simulation to automatically place the parabolic tips. Lastly, due to the similarly between the heat and mass transfer equations, recent results [SBB*05] also suggest that methods similar to the ones we present here could simulate stalactite formation.

Acknowledgements

The authors would like to thank the 3Delight team for their rendering support, in particular Olivier Paquet and Aghiles Kheffache for expediently answering our many questions and generously providing an additional license. Additionally, we would also like to thank the anonymous reviewers for their suggestions. This work was supported in part by Army Research Office, Defense Advanced Research Projects Agency, Intel Corporation, National Science Foundation, Office of Naval Research, and RDECOM.

Figure 4: ***Icicle Star:*** *Inspired by an Andy Goldsworthy sculpture [Gol90], we simulated the growth of an icicle star. See for example* www.gac.culture.gov.uk/gac_images/Fullsize/16622A.jpg *for the original photograph. Goldsworthy is an artist who constructs sculpture from natural materials, in this case icicles. While this formation cannot occur in nature, our user controls allow such a 'zero gravity' star to be grown. The simulation completed in 18 minutes.*

References

[AS95] ADALSTEINSSON D., SETHIAN J.: A fast level set method for propagating interfaces. *Journal of Computational Physics 118* (1995), pp. 269–277.

[AS99] ADALSTEINSSON D., SETHIAN J.: The fast construction of extension velocities in level set methods. *Journal of Computational Physics* (1999), 2–22.

[CMIT02] CARLSON M., MUCHA P., III B. V. H., TURK G.: Melting and flowing. *Proc. of ACM SIGGRAPH Symposium on Computer Animation* (2002).

[CMK*01] CHEN S., MERRIMAN B., KANG M., CAFLISCH R., RATSCH C., CHENG L., GYURE M., FEDKIW R., OSHER S.: Level set method for thin film epitaxial growth. *J. Comp. Phys. 167* (2001).

[DJ05] DONNER C., JENSEN H. W.: Light diffusion in multi-layered translucent materials. *ACM Trans. Graph. 24*, 3 (2005).

[EFFM02] ENRIGHT D., FEDKIW R., FERZIGER J., MITCHELL I.: A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics 183* (2002), 83–116.

[FOK05] FELDMAN B. E., O'BRIEN J. F., KLINGNER B. M.: Animating gases with hybrid meshes. In *Proceedings of ACM SIGGRAPH 2005* (2005).

[Fra49] FRANK F.: The influence of dislocation on crystal growth. *Disc. Faraday Soc. 5* (1949), 48–54.

[GF05] GIBOU F., FEDKIW R.: A fourth order accurate discretization for the laplace and heat equations on arbitrary domains, with applications to the stefan problem. *J. Comput. Phys. 202* (2005), 577–601.

[GFCO03] GIBOU F., FEDKIW R., CAFLISCH R., OSHER S.: A level set approach for the simulation of dendritic growth. *J. Sci. Comput. 19* (2003), 183–199.

[Gol90] GOLDSWORTHY A.: *Andy Goldsworthy: A Collaboration with Nature*. Harry N Abrams, 1990.

[Hil87] HILL J. M.: *One-dimensional Stefan Problems: an Introduction*. John Wiley & Sons, 1987.

[Iva47] IVANTSOV G.: Temperature field around the spherical, cylindrical and needle-crystals which grow in supercooled melt. *Dokl Akad Nauk USSR 58* (1947), 67.

[JLD99] JENSEN H., LEGAKIS J., DORSEY J.: Rendering of wet materials. *Rendering Techniques '99* (1999), 273–282.

[JMLH01] JENSEN H., MARSCHNER S., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. *Proceedings of SIGGRAPH 2001* (2001), 511–518.

[KG93] KHARITONSKY D., GONCZAROWSKI J.: A physically based model for icicle growth. *The Visual Computer* (1993), 88–100.

[KHL04] KIM T., HENSON M., LIN M.: A hybrid algorithm for modeling ice formation. *Proc. of ACM SIGGRAPH / Eurographics Symposium on Computer Animcation* (2004).

[KL03] KIM T., LIN M.: Visual simulation of ice crystal growth. *Proc. of ACM SIGGRAPH / Eurographics Symposium on Computer Animcation* (2003).

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *Proc. of SIGGRAPH* (2004), 457–462.

[LIGF05] LOSASSO F., IRVING G., GUENDELMAN E., FEDKIW R.: Melting and burning solids into liquids and gases. *IEEE TVCG* (2005).

[Mak88] MAKKONEN L.: A model of icicle growth. *Journal of Glaciology* (1988), 64–70.

[Mei92] MEIRMANOV A.: *The Stefan problem*. W. De Gruyter expositions in mathematics, 1992.

[Mes53] MESSINGER B.: Equilibrium temperature of an unheated icing surface as a function of air speed. *J. Aero. Sci* (1953), 29–42.

[MH99] MYERS T., HAMMOND D.: Ice and water film growth from incoming supercooled droplets. *International Journal of Heat and Mass Transfer 42* (1999), PP2233–2242.

[MMN*94] MAENO N., MAKKONEN L., NISHIMURA K., KOSUGI K., TAKAHASHI T.: Growth rates of icicles. *Journal of Glaciology 40* (1994), 319–326.

[MS64] MULLINS W., SEKERKA R.: Stability of a planar interface during solidification of a dilute binary alloy. *Journal of Applied Physics 35*, 2 (1964), 444–451.

[OF02] OGAWA N., FURUKAWA Y.: Surface instability of icicles. *Physical Review E 66* (2002), 041202.

[OF03] OSHER S., FEDKIW R.: *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2003.

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 193–202.

[Sai96] SAITO Y.: *Statistical Physics of Crystal Growth*. World Scientific, 1996.

[SBB*05] SHORT M. B., BAYGENTS J. C., BECK J. W., STONE D. A., III R. S. T., GOLDSTEIN R. E.: Stalactite growth as a free-boundary problem: A geometric law and its platonic ideal. *Physical Review Letters 94*, 1 (2005), 018501.

[Set99] SETHIAN J.: *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.

[SL94] SZILDER K., LOZOWSKI E.: An analytical model of icicle growth. *Annals of Glaciology 19* (1994), 141–145.

[SRF05] SELLE A., RASMUSSEN N., FEDKIW R.: A vortex particle method for smoke, water and explosions. In *Proceedings of ACM SIGGRAPH 2005* (2005), pp. 910–914.

[SS92] SETHIAN J., STRAIN J.: Crystal growth and dendritic solidification. *J. Comput. Phys. 98* (1992), 231–253.

[Uen03] UENO K.: Pattern formation in crystal growth under parabolic shear flow. *Physical Review E 68* (2003), 021603.

[Uen04] UENO K.: Pattern formation in crystal growth under parabolic shear flow ii. *Physical Review E 69* (2004), 051604.

[Wet01] WETTLAUFER J.: *The Stefan Problem: Polar Exploration and the mathematics of moving boundaries*. Styria Verlag, 2001.

[WLK03] WEI X., LI W., KAUFMAN A.: Interactive melting and flowing of viscous volumes. *Proceedings of Computer Animation and Social Agents 2003* (2003).

Figure 5: ***A freezing fountain:*** *Ice forms in a fountain one morning when the temperature dips below freezing. This simulation completed in 30 minutes.*

Figure 6: ***Ice forming on a roof:*** *Icicles form from the snow melt running off down a roof. This simulation completed in 5 minutes.*

Figure 7: ***Ice forming on a roof:*** *Icicles form from the snow melt running off down a roof. This simulation completed in 5 minutes.*

Yeuhi Abe & Jovan Popović / Interactive Animation of Dynamic Manipulation

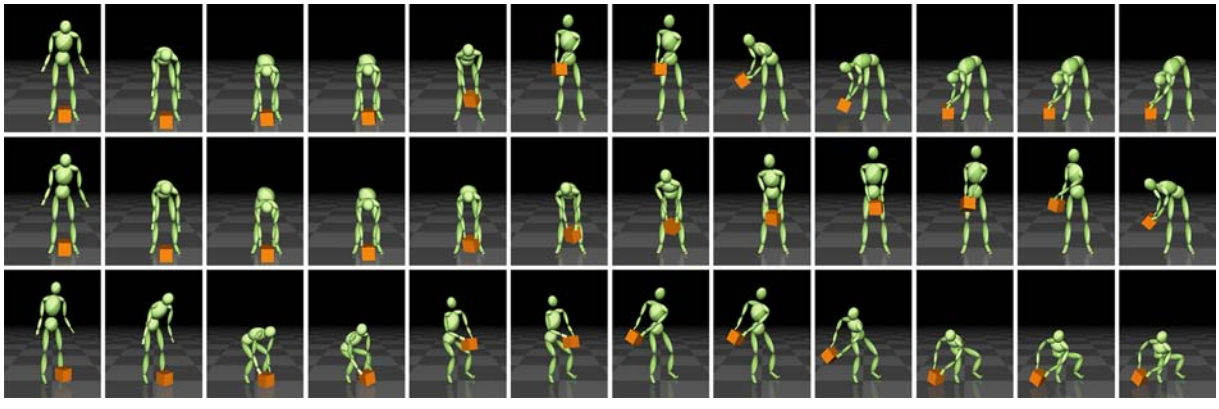

Figure 4: *Our control algorithm directs a real-time simulation of a character to accomplish manipulations, such as displacing a box (top row). Manipulations are compactly described. In the above example, only four Cartesian goal positions are used to describe the motion of the hands and the box. The missing details are filled in with a secondary posture task that incorporates recorded motion postures from a similar performance. The control adapts naturally to changes in the environment. As expected, increasing the weight of the box (second row) produces a slower lift. The performance of the task can also be changed by using a different recorded motion in the posture task (third row).*