Simulation, Modeling and Authoring of Glaciers

Oscar Argudo, Eric Galin, Adrien Peytavie, Axel Paris, Eric Guérin

Simulation, Modeling and Authoring of Glaciers

OSCAR ARGUDO, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France
ERIC GALIN, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France
ADRIEN PEYTAVIE, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France
AXEL PARIS, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France
ERIC GUÉRIN, Univ Lyon, INSA-Lyon, CNRS, LIRIS, France

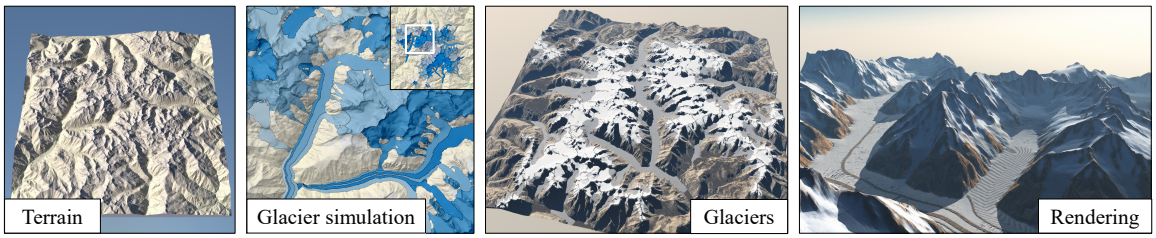

Fig. 1. Given an input terrain, user-prescribed control regions defining the average temperature and precipitation, and glacier parameters such as accumulation and ablation rates, we perform a multiresolution Shallow Ice Approximation to generate the ice thickness of the glaciers as well as physical quantities such as flow direction or shear stress. From those parameters, we procedurally amplify the ice layer with details such as icefalls, crevasses, and moraines. We implemented the simulation on graphics hardware, allowing for interactive feedback and controlled authoring.

Glaciers are some of the most visually arresting and scenic elements of cold regions and high mountain landscapes. Although snow-covered terrains have previously received attention in computer graphics, simulating the temporal evolution of glaciers as well as modeling their wide range of features has never been addressed. In this paper, we combine a Shallow Ice Approximation simulation with a procedural amplification process to author high-resolution realistic glaciers. Our multiresolution method allows the interactive simulation of the formation and the evolution of glaciers over hundreds of years. The user can easily modify the environment variables, such as the average temperature or precipitation rate, to control the glacier growth, or directly use brushes to sculpt the ice or bedrock with interactive feedback. Mesoscale and smallscale landforms that are not captured by the glacier simulation, such as crevasses, moraines, seracs, ogives, or icefalls are synthesized using procedural rules inspired by observations in glaciology and according to the physical parameters derived from the simulation. Our method lends itself to seamless integration into production pipelines to decorate reliefs with glaciers and realistic ice features.

CCS Concepts: • **Computing methodologies** → **Shape modeling**.

Additional Key Words and Phrases: Procedural terrain modeling, glacier simulation

Authors' addresses: Oscar Argudo, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France, oscar.argudo@liris.cnrs.fr; Eric Galin, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France, eric.galin@liris.cnrs.fr; Adrien Peytavie, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France, adrien.peytavie@liris.cnrs.fr; Axel Paris, Univ Lyon, Université Lyon 1, CNRS, LIRIS, France, axel.paris@liris.cnrs.fr; Eric Guérin, Univ Lyon, INSA-Lyon, CNRS, LIRIS, France, eric.guerin@liris.cnrs.fr.

1 INTRODUCTION

Glaciers take a variety of forms: from valley glaciers found in mountains, ice fields striding mountain ranges, to ice shelves floating on the ocean surface. Formed where the accumulation of snow exceeds its melting over many years, glaciers slowly deform and flow due to stresses induced by their weight, creating crevasses, seracs, and other distinguishing features. They also abrade bedrock and debris from their substrate to create landforms such as cirques and moraines. We refer to Figures 3 and 24 for illustrations of the different formations, and Section 6 for a detailed description.

The sheer variety of shapes and range of scales of glaciers presents significant simulation and modeling challenges. Despite the vast variety of methods for generating synthetic terrains [Galin et al. 2019], effectively modeling, generating, and authoring glaciers, *i.e.*, the ice sheet covering bedrock, remains an unsolved problem. The challenge stems not only from the complex physical mechanisms involved in their formation and evolution, but also from the range of scales of landform features that should be generated.

Several validated models have been proposed in geology for simulating ice-flow and glacial erosion [Braun et al. 1999; Egholm et al. 2011; Headley et al. 2012; Mahaffy 1976]. These models shape mountains and yield characteristic landforms such as tunnel valleys, cirques, and arêtes, hanging valleys at the convergence of two glaciers, and glacial lakes. Erosion simulation is beyond the scope of our work, as it addresses large-scale terrain erosion taking place over tens of thousands of years. Although different approaches for

simulating the evolution of the seasonal snow cover under the influence of the environment have been proposed, those techniques address a different and complementary phenomenon: seasonal snow accumulation, which occurs at a different time scale.

Our work comes from the observation that the general shape of glaciers can be effectively approximated by efficient simulations such as the Shallow Ice Approximation (SIA), while details at different scales can be synthesized procedurally by using generation rules derived from the many observations made in geomorphology [Huggett 2016] and glaciology [Cuffey and Paterson 2010].

In this paper, we propose a multiresolution simulation based on the SIA, allowing for easy and efficient control for representing and authoring glaciers across a range of scales (see accompanying video). Our method not only simulates the shape of glaciers but also generates small-scale characteristic landforms such as moraines, crevasses, or ogives. We use a layered data structure to represent the bedrock, ice, and snow, and simulate the ice evolution at interactive rates while providing control to the user. The output of the simulation, *i.e.* the ice thickness, the flow direction, shear stress, and other parameters, is then seamlessly streamed to a procedural amplification process that synthesizes feature density maps that allow a stochastic placement of glacier features.

The main contributions of our work include: 1) an interactive multiresolution simulation of ice mass evolution based on the physics of glaciers and models currently used in glaciology, 2) a procedural approach based on rules from geomorphology, for amplifying the data produced by the simulation and modeling detailed features such as crevasses, icefalls, or moraines and 3) a variety of intuitive control tools that allow the users to interact with the simulation or modify the glacier features placement with interactive feedback.

2 RELATED WORK

Although simulating glaciers has never been addressed in Computer Graphics, our method relates to terrain modeling and snow cover generation. We refer the reader to [Galin et al. 2019] for a complete review of terrain modeling techniques, including procedural generation, erosion simulation and example-based synthesis. In geomorphology, we refer to [Cuffey and Paterson 2010] for a more complete overview of the numerical simulation of glaciers and erosion. Existing methods for modeling snow cover fall in two categories: physical simulation, and procedural generation.

2.1 Simulation

Particle-based techniques compute a distribution of snow by simulating the effect of wind on the trajectory of snow particles so that they drift and gather onto the ground or objects. Nishita *et al.* [1997] use an implicit model to reconstruct a smooth snow surface from skeletal point elements. Fearing [2000] introduced an accumulation model by sampling surface elements with snow particles and backtracking their trajectory to source clouds while accounting for collision and stability. Subsequent work improves wind influence when determining the upward trajectory, using Navier-Stokes [Moeslund et al. 2005] or Boltzmann approaches [Wang et al. 2006]. Hinks and Museth [2009] used a dynamic wind-field to carry snow particles and generate realistic snow accumulation patterns using an implicit-surface model. Stomakhin *et al.* [2013] simulated the complex behavior of snow under different environmental conditions by using a semi-implicit Lagrangian method.

Separating static structure represented by voxels [Sai-Keung and I-Ting 2015] or a heightfield [Dagenais et al. 2016], from dynamic elements implemented with particles can also be beneficial, since it allows the interaction between snow and dynamic objects [Dagenais et al. 2016]. Particle-based methods suffer from the common limitation that they do not scale beyond small scenes (with an upper limit of about $100 \times 100\,\mathrm{m}^2$).

Physically-based heat transfer methods use the different thermal interactions between objects and the influence of environmental conditions to simulate snow accumulation, ice formation and snow melt. Muraoka and Chiba [2000] simulated snowfall using vortex fields and snow melt using heat conduction. Maréchal *et al.* [2010] introduced a complete finite volume method over a voxel grid to simulate snowfall and conductive, convective and radiative thermal transfers, and complex phenomena such phase changes where snow melts to water or water freezes into ice. Unfortunately, heat transfer methods are limited because of their high computational cost and memory-intensive volumetric representations.

Cordonnier *et al.* [2018] generated plausible snow cover over mountain ranges by using an event-based stochastic simulation operating on a layered model. The framework accounts for environmental factors such as sunlight, temperature, and wind, but is restricted to terrains of medium size ($10 \times 10\,\mathrm{km}^2$) to preserve authoring and interactivity.

2.2 Procedural approaches

Procedural methods rely on phenomenological approaches to generate a plausible snow cover. Accessibility and occlusion are commonly used to characterize the thickness of accumulated snow [Premože et al. 1999]. A direct extension for large-scale terrains consists in linking snow thickness to ambient occlusion, direct illumination [Foldes and Benes 2007; Reynolds et al. 2015; Tokoi 2006] or wind-drift approximated by directional occlusion [Moriya and Takahashi 2010]. Chen *et al.* [2003] treat differently nearby objects (using a displacement map) and more distant objects (with a volumetric texture map) to accelerate the computations.

Inspired from granular material dynamics, height span maps [Sumner et al. 1999] have been successfully used to model real-time snow accumulation with the incorporation of phenomenologically-inspired statistics [Festenberg and Gumhold 2011, 2009]. Grosbellet *et al.* [2016] introduced a general framework for the generation of details in complex scenes, by approximating the interaction of objects between them and the environment through scalar fields including temperature and humidity, therefore allowing the generation of snow cover and icicles on objects.

A general limitation of existing methods is that they only account for seasonal snow effects and do not model ice accumulation. Moreover, they are limited to small or medium size landscapes. In contrast, our multiresolution simulation method combined with a procedural amplification method for generating details allows to interactively generate large scenes, up to $100 \times 100\,\mathrm{km}^2$.

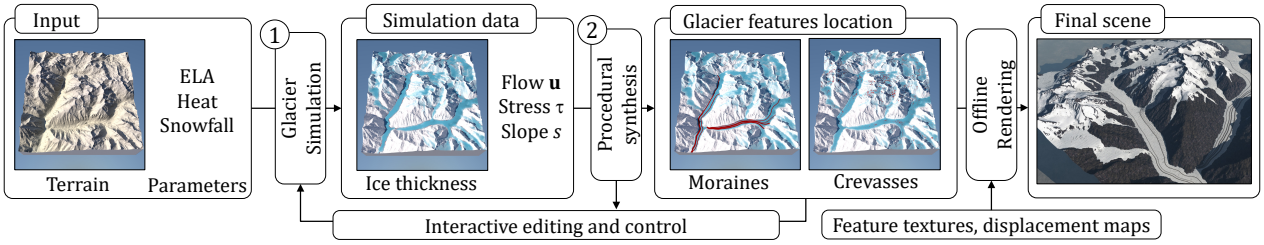

Fig. 2. Overview of our method: given an input terrain $\mathcal{T}$, climate parameters such as average temperature and temperature gradient, user-prescribed control regions for dampening an amplifying melting, our method generates a low resolution (≈ 120 m per cell) ice layer using a SIA simulation. This representation is procedurally amplified to obtain a high resolution ice model (≈ 30 m per cell) featuring details such as icefalls, crevasses, and moraines.

3 OVERVIEW

A glacier is a body of dense ice, formed in places where the snow accumulation is larger than the ablation. Recent accumulated snow exerts pressure onto older layers of snow, causing it to compact and eventually become glacier ice. This ice deforms due to gravity-induced stresses, and therefore is constantly flowing downhill under its own weight.

Glaciers can be classified into different categories according to their morphology. Valley glaciers or mountain glaciers form on many mountain ranges and are constrained by the terrain topography: they originate in upper mountain slopes and cirques (bowl-shaped valley heads) and flow along the valley (Figure 3). Ice fields are larger areas of interconnected glaciers that expand over mountain ridges, only the higher peaks and ridges are exposed. In contrast, ice caps and ice sheets completely submerge the underlying terrain, creating a dome shape that spreads out from the center. Our work mainly focuses on mountain glaciers and ice fields, although ice caps and sheets can also be generated.

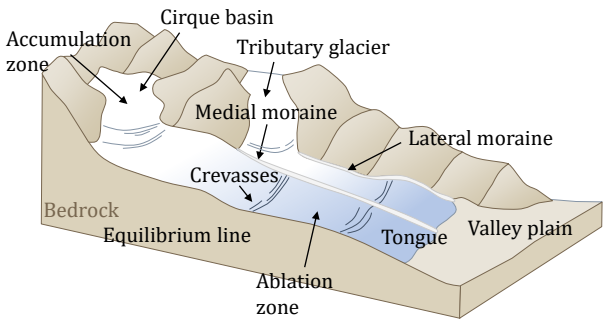

Fig. 3. Symbolic representation of a glacier and some of its major features.

Our method computes the temporal evolution of an ice field $\mathcal{H}(t)$ representing the glacier covering an otherwise static terrain $\mathcal{T}$ under the influence of temperature and precipitation. Our method combines two steps (Figure 2): a physically-based simulation for computing the evolution of the glacier over time, whose output is directly streamed to a procedural generation algorithm for synthesizing the high-resolution ice layer amplified with detailed glacier features such as crevasses, moraines, or ogives.

Simulation. Our framework uses a layered heightfield model: at any time step, the scene consists of a discrete bedrock layer $\mathcal{B}$ and a time-varying ice layer $\mathcal{H}(t)$ composed of $n \times n$ cells. Starting from an elevation map (real or synthetic) and optionally an initial ice thickness layer $\mathcal{H}$, the simulation computes the evolution of the ice field $\mathcal{H}(t + \delta t)$ from $\mathcal{H}(t)$ according to the control parameters and using the SIA model (Section 4). We employ a multiresolution method to accelerate computations and obtain a high-resolution ice layer covering the terrain. The simulation is parallelized on graphics hardware for interactive feedback (Section 5). At any time, the user can pause, resume, restart, or adjust the simulation parameters.

Modeling glacier features. Simulating the interaction between ice and flow to create complex features such as crevasses would be computationally prohibitive. Therefore, we propose a procedural approach to modeling detailed glacier features. From the simulation data, *i.e.*, the ice layer $\mathcal{H}$ of the simulation and other useful information such as the direction of the ice flow or the shear stress, we derive density maps for the different glacial features such as crevasses, ogives, or moraines (Section 6). We then use these maps to locate features and synthesize the final high-resolution landscape.

Control. From an authoring perspective, we offer two modes of control. The user can modify the simulation parameters which have very intuitive influence over the shape of glaciers. Specifically, it is possible to modify the equilibrium line, which is the main parameter influencing the shape of the glacier: the user can change it at any time in the simulation, or specify a deviation map to create interesting localized effects. The user may interactively edit the feature density maps, which provides direct control over the final feature placement.

4 GLACIER DYNAMICS

We now introduce some of the fundamental concepts required to simulate the evolution of a glacier. For a deeper understanding, we refer the reader to glaciology textbooks such as [Bennett and Glasser 2011; Cuffey and Paterson 2010].

Glaciers form over regions where more snow is accumulated than ablated (i.e. melted or sublimated) over a yearly cycle. Accumulated snow is progressively compacted into glacial ice, which then deforms under its own weight causing a slow downhill flow. The difference between accumulation and ablation at a position $\mathbf{p}$ is called the mass balance $\dot{m}(\mathbf{p})$, and varies over the surface $\mathcal{G}$ of a glacier defining the

accumulation zone $\mathcal{G}_A = \{\mathbf{p} \mid \dot{m}(\mathbf{p}) > 0\}$, the ablation zone $\mathcal{G}_B = \{\mathbf{p} \mid \dot{m}(\mathbf{p}) < 0\}$ and the equilibrium line altitude ELA $= \{\mathbf{p} \mid \dot{m}(\mathbf{p}) = 0\}$, respectively (Figure 4).

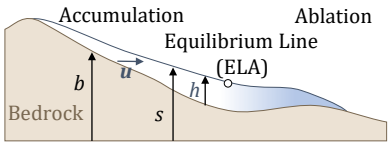

Fig. 4. Notations for the different regions of a glacier.

Gravity causes the ice to deform, creating shear stress. In response, ice flows through three mechanisms [Bennett and Glasser 2011]: internal deformation, basal sliding, and bedrock deformation. This flow can be modeled using the full Stokes equations for an incompressible, viscous, and non-Newtonian fluid as follows:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ 0 &= -\nabla p + \nabla \cdot \tau_{ij} + \rho g \\ \dot{\varepsilon}_{ij} &= A\tau^{n-1}\tau_{ij} \end{aligned} \tag{1}$$

where $\mathbf{u}$ is the velocity, p the pressure, ρ the density of ice and g the gravity acceleration. The first equation is the incompressibility condition. The second one represents the balance of forces and its left hand-side is equal to zero: ice is a very slow moving fluid and inertia is negligible. The last equation is known as the Glen-Nye flow law [Glen and Perutz 1955; Nye 1957] and models the non-linear viscosity using a power law that relates the deviatoric stress tensor τ_{ij} to the strain rate tensor $\dot{\varepsilon}_{ij}$, with $\tau = \frac{1}{2}[\mathrm{tr}(\tau_{ij})^2 - \mathrm{tr}({\tau_{ij}}^2)]$ the second invariant of τ_{ij}. The constant A depends on the composition and temperature of the ice; the exponent n is typically set to 3 for most glaciers. Finally, note that there is no time derivative in this set of equations: we do not need the previous momentum or velocity to model the evolution of ice [Bueler 2016].

Solving the previous equations directly is both complex and computationally expensive. The *Shallow Ice Approximation* (SIA) is a particular zero-order model that neglects longitudinal and transverse stresses, as well as vertical stress gradients. It was developed for modeling ice sheets, where glacier thickness is much smaller than lateral extent [Hutter 1983]. However, due to its simplicity and efficiency, it is widely used even on complex mountainous terrains [Headley et al. 2012; Kessler et al. 2006].

Following [Knap et al. 1996], and ignoring bedrock deformation, the temporal evolution of the ice thickness using SIA can be expressed as:

$$\frac{\partial h}{\partial t} = \dot{m} - \nabla \cdot F = \dot{m} - \nabla \cdot h\,(\bar{\mathbf{u}}_d + \mathbf{u}_s) \tag{2}$$

F represents the ice flux or volume discharge, and depends on the depth-averaged internal deformation velocity $\bar{\mathbf{u}}_d$ and the basal sliding velocity $\mathbf{u}_s$. We denote the bedrock b, the ice surface s and the ice thickness $h = s - b$ (see Figure 4) as functions of space ($\Omega \subset \mathbb{R}^2$ refers to the domain of the terrain) and time. To compute the deformation velocity $\bar{\mathbf{u}}_d$, let us consider the shear stress τ at a certain elevation z below the surface:

$$\tau = \rho g(s - z)\nabla s \tag{3}$$

For the particular case of $z = b$, we obtain the basal shear stress $\tau_b = \rho g h \nabla s$.

The relationship between the stress τ and the strain rate $\dot{\varepsilon}$ is expressed by the Glen flow law: $\dot{\varepsilon} = A\tau^n$ where τ^n is defined as $\tau^n = \|\tau\|^n \cdot \tau/\|\tau\|$ as τ is a vector. By integrating $\dot{\varepsilon}$ vertically, we can obtain the depth-averaged internal deformation velocity:

$$\bar{\mathbf{u}}_d = \frac{2A}{n+2} h \tau_b^n = -\Gamma_d\, h^{n+1} \|\nabla s\|^{n-1} \nabla s \qquad \Gamma_d = \frac{2A(\rho g)^n}{n+2} \tag{4}$$

The term Γ_d is commonly used to group the constant terms.

When the ice bed is frozen, the sliding velocity $\mathbf{u}_s$ equals zero. However, in glaciers such as those found in continental mountain ranges, it has a non-negligible effect over the ice flow. Modeling slide remains a challenging open problem in glaciology and different formulations exist. One possibility is to model $\mathbf{u}_s$ as proportional to some power p of the basal stress τ_b [Bindschadler 1983]:

$$\mathbf{u}_s = -k_s \tau_b^p N^{-1} \tag{5}$$

where k_s is a sliding coefficient, and $N = \tau_b - P_w$ is the effective pressure: the difference between the ice pressure τ_b and the water pressure P_w at the ice-bed interface. Using $p = n$ and grouping the constant terms into a factor Γ_s yields the formulation found in [Headley et al. 2012]:

$$\mathbf{u}_s = -\Gamma_s h^{n-1} \|\nabla s\|^{n-1} \nabla s \tag{6}$$

Putting together equations 2, 4 and 6, we obtain the equation describing the evolution of the glacier height over time:

$$\frac{\partial h}{\partial t} = \dot{m} - \nabla \cdot h^n (\Gamma_d h^2 + \Gamma_s) \|\nabla s\|^{n-1} \nabla s \tag{7}$$

A glacier is said to be in steady state when it does neither advance nor retreat, *i.e.*, $\partial h/\partial t = 0$.

5 GLACIER SIMULATION

Given initial bedrock and ice layers, and a function $\dot{m}$ defined over the simulation domain Ω, we aim at computing the evolution of the glacier mass throughout time. First, we address the numerical computations on a fixed grid resolution; and then present a multiresolution scheme that accelerates the algorithm by several orders of magnitude while keeping errors small.

5.1 Numerical model

We need to solve Equation 2 to simulate the evolution of the ice coverage. Contrary to fluid equations, there is no velocity term: speed and direction of moving ice mass are derived from the ice thickness and surface gradient. Thus, one common approach is to rewrite it as a diffusion process:

$$\frac{\partial h}{\partial t} = \dot{m} - \nabla \cdot (D\,\nabla s) \tag{8}$$

where

$$D = \Gamma_d h^{n+2} \|\nabla s\|^{n-1} + \Gamma_s h^n \|\nabla s\|^{n-1} \tag{9}$$

and solve it using a staggered grid (see Figure 6), such that $h_{i,j}$ denotes ice thickness at cell (i, j) center and $h_{i+\frac{1}{2},j}$ the interpolated value for the midpoint of its right edge.

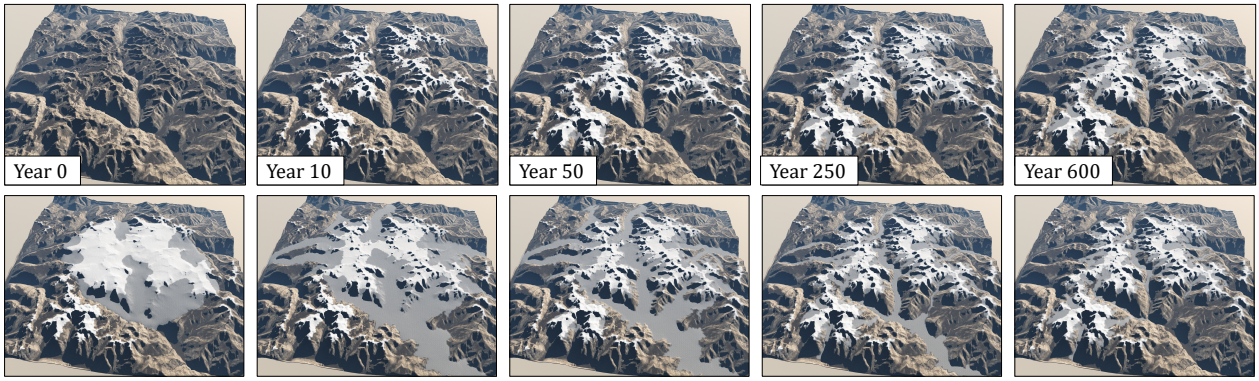

Fig. 5. Evolution of the ice coverage on a terrain, starting from bare bedrock (top) or from a user-drawn ice cap (bottom). The steady state is reached after 630 years and 583 years, respectively.

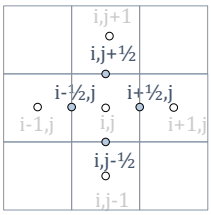

Fig. 6. Staggered grid indices.

Since the diffusivity or flux is computed at the edges of the grid, SIA simulations can suffer from mass conservation problems on mountainous terrain because steep gradients can create an outgoing flux on the upstream cell larger than the amount of ice inside it. Clamping the resulting negative ice thicknesses to 0 would create mass, and the simulation would become numerically unstable after a few iterations due to the h^{n+2} term in Equation 9.

To solve this, Jarosch *et al.* [2013] proposed using a second-order MUSCL scheme, a finite volume method that yields accurate and stable solutions even with large gradients or discontinuities [van Leer 1979]. The key concept is to use the slope-limited reconstruction of the cell states and compute the diffusion term at the cell edges. We detail the computation of the diffusion term on the right edge, $D_{i+\frac{1}{2},j}$, as done by [Jarosch et al. 2013] except they did not use a sliding velocity term (*i.e.* $\Gamma_s = 0$). The diffusion on the other edges is computed analogously.

First, we extrapolate two ice thickness values at the cell right edge $i + \frac{1}{2}$, denoted with superscripts as $i + \frac{1}{2}^-$ and $i + \frac{1}{2}^+$:

$$\begin{aligned} h_{i+\frac{1}{2}^-,j} &= h_{i,j} + \frac{1}{2}\phi(r_{i,j})(h_{i+1,j} - h_{i,j}) \\ h_{i+\frac{1}{2}^+,j} &= h_{i+1,j} - \frac{1}{2}\phi(r_{i+1,j})(h_{i+2,j} - h_{i+1,j}) \end{aligned} \tag{10}$$

where the slope

$$r_{i,j} = \frac{h_{i,j} - h_{i-1,j}}{h_{i+1,j} - h_{i,j}}$$

is limited by a function $\phi(r)$ that prevents oscillations near shocks or discontinuities. In particular, we used the superbee limiter by Roe [1986]: $\phi(r) = \max(0, \min(2r, 1), \min(r, 2))$.

We compute the two flux values $D_{i+\frac{1}{2}^-,j}$ and $D_{i+\frac{1}{2}^+,j}$ by using Equation 9 with $h_{i+\frac{1}{2}^-,j}$ and $h_{i+\frac{1}{2}^+,j}$ respectively. Finally, the flux on this cell edge is obtained as:

$$D_{i+\frac{1}{2},j} = \begin{cases} D^{\min}_{i+\frac{1}{2},j} & \text{if } s_{i+1,j} \le s_{i,j} \text{ and } h_{i+\frac{1}{2}^-,j} \le h_{i+\frac{1}{2}^+,j} \\ D^{\max}_{i+\frac{1}{2},j} & \text{if } s_{i+1,j} \le s_{i,j} \text{ and } h_{i+\frac{1}{2}^-,j} > h_{i+\frac{1}{2}^+,j} \\ D^{\max}_{i+\frac{1}{2},j} & \text{if } s_{i+1,j} > s_{i,j} \text{ and } h_{i+\frac{1}{2}^-,j} \le h_{i+\frac{1}{2}^+,j} \\ D^{\min}_{i+\frac{1}{2},j} & \text{if } s_{i+1,j} > s_{i,j} \text{ and } h_{i+\frac{1}{2}^-,j} > h_{i+\frac{1}{2}^+,j} \end{cases} \tag{11}$$

where:

$$\begin{aligned} D^{\min}_{i+\frac{1}{2},j} &= \min\left(D_{i+\frac{1}{2}^-,j},\ D_{i+\frac{1}{2}^+,j}\right) \\ D^{\max}_{i+\frac{1}{2},j} &= \max\left(D_{i+\frac{1}{2}^-,j},\ D_{i+\frac{1}{2}^+,j}\right) \end{aligned}$$

The same equations are used to compute the diffusivity on the other edges of the cell. Finally, Equation 2 is solved using forward Euler integration with the following adaptive time step:

$$\Delta t = \frac{1}{2(n+1)} \frac{\min\left(\Delta x^2, \Delta y^2\right)}{\max\left(D_{i+\frac{1}{2},j}, D_{i-\frac{1}{2},j}, D_{i,j+\frac{1}{2}}, D_{i,j-\frac{1}{2}}\right)} \tag{12}$$

Using the previous integration scheme directly leads to square-shaped glaciers on cone-shaped peaks, as shown in Figure 7. The reason is that we are only using a 4-neighborhood on a regular grid to compute slope and ice diffusion. An 8-neighborhood or a higher-order method could be used instead. However, for the sake of efficient interactive modeling, we observed that we could obtain a correct round glacier without additional computational cost by alternating between two neighborhood stencils at every simulation step: the 4-neighborhood axis-aligned and the same rotated by 45 degrees (using the diagonal cells).

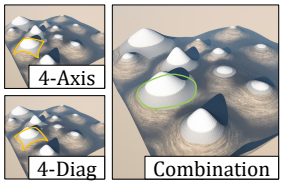

Fig. 7. Alternating between an axis-aligned and diagonal 4-neighborhood.

Finally, an interesting observation is that the steady-state of a glacier depends entirely on the bedrock morphology and the accumulation function $\dot{m}$. Figure 5 shows two cases of temporal evolutions on the same terrain and different initial conditions: bare bedrock and user-drawn ice cap. The first case shows that the glacier

starts forming in accumulation zones, and then progressively converges to a steady-state featuring connected basins. In the second case, the shape and slopes of the glacier evolve and stabilize during the first years, adapting to the valleys, excess ice is then ablated, forcing some glacier tongues to retreat or disappear and producing the same final steady-state as the first case.

5.2 Multiresolution

Simulating the ice diffusion process is complex and time-demanding. Moreover, the time step Δt for the forward integration scheme (Equation 12) is proportional to the grid cell area. Since most of the glacier body has a smooth surface, a coarse resolution simulation produces a good approximation of the high-resolution glacier cover; we adjust excess or lacking ice by the same simulation after upsampling. Therefore, we use a multiresolution approach that accelerates computation by several orders of magnitude. While this scheme can be interpreted as half of a V-cycle of a multigrid method, we are not necessarily solving Equation 2 for its steady-state, as we might be interested in an intermediate step of the glacier evolution.

First, we downsample the given terrain bedrock $\mathcal{B}$ several times $\mathcal{B} = \mathcal{B}^0, \mathcal{B}^1, \ldots, \mathcal{B}^n$ until the largest grid dimension in $\mathcal{B}^n$ has about 250 cells. We simulate the glacier evolution in $\mathcal{B}^n$ until a steady-state ice layer $\mathcal{H}^n$ is reached or the user stops the simulation, with $\epsilon^n = \partial h/\partial t$. Then, we iteratively upsample the ice layer and apply a correction step until we reach the original resolution $\mathcal{H}^0$.

To compute $\mathcal{H}^{i-1}$ from $\mathcal{H}^i$, we use bilinear interpolation to upsample the ice surface $\mathcal{S}^i = \mathcal{B}^i + \mathcal{H}^i$ and compute $\mathcal{H}^{i-1}$ as:

$$\mathcal{H}^{i-1} = (\mathcal{S}^{i-1} - \mathcal{B}^{i-1}) \cdot \mathcal{M}^{i-1}$$

$\mathcal{M}^{i-1}$ is an ice presence map bilinearly interpolated from the binary map $\mathcal{M}^i = \mathcal{H}^i > 0$. Note that we need to introduce $\mathcal{M}$ to prevent ice from being created at ice-free locations $\mathbf{p}$ from $\mathcal{H}^i$ such that $b^{i-1}(\mathbf{p}) < b^i(\mathbf{p})$.

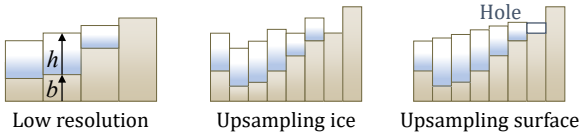

Fig. 8. Upsampling the ice thickness layer from a lower resolution (left) by directly interpolating the ice thickness layer $\mathcal{H}$ (middle), and by interpolating the ice surface $\mathcal{S}$ (right). The outlined square illustrates an example of a local minimum filled after the upsampling process.

The correction step serves to stabilize and adjust the ice layer $\mathcal{H}^{i-1}$ to the more detailed geometry of the bedrock $\mathcal{B}^{i-1}$. We run the simulation algorithm starting with $\mathcal{H}^{i-1}$ until $\epsilon^{i-1} = k\epsilon^i$, with k being a proportionality factor. Doubling the resolution not only quadruples the number of cells but also reduces the maximum time step Δt by a factor of four (Equation 12). However, only a few iterations are needed until $\partial h/\partial t \leq \epsilon^{i-1}$.

We observed that a great amount of time in the correction simulation steps was spent filling holes in to the new geometry. Therefore, we propagate the ice surface elevation to neighboring positions that are local minima of $\mathcal{S}^{i-1}$, and to ice-empty cells between an ice cell with a higher surface and a rock wall, as shown in Figure 8. Although this strategy might be adding mass to the glacier, the number of affected cells is very small in practice compared to the overall number of ice cells, and additional incorrect mass is eventually handled by the correction step. Our hole filling approximation leads to up to 6 times speed-ups on the correction step time, depending on the relief of the terrain.

6 MODELING GLACIAL FEATURES

Glaciers include a vast variety of small-scale and mesoscale landforms that are not readily captured by the glacier simulation itself: crevasses, moraines, bergschrunds, seracs, and icefalls. Our approach consists of procedurally generating those features according to the simulated data (such as stress, flow direction, flow acceleration, temperature) and to the terrain data (elevation, slope). For every type of feature, we compute a probability map defining their presence and density (Figure 9). The final high-resolution icefield is generated from those density maps by bombing textured relief models (see Section 7).

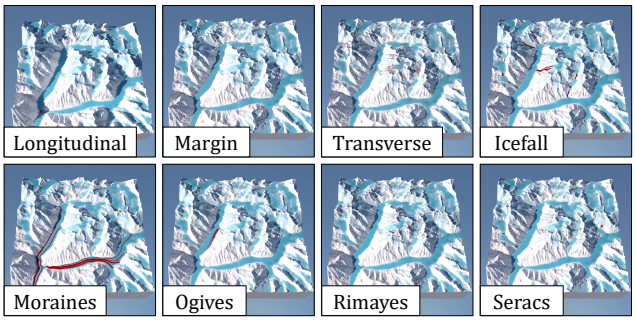

Fig. 9. Presence maps for various glacier features (in red) computed from the ice thickness (in blue) and simulation data.

6.1 Crevasses

Crevasses are fractures on the surface of a glacier, opening from some centimeters up to a few meters wide, and reaching depths up to 50 m. Since pressure increases with depth under the glacier surface, the uppermost part of a glacier mass behaves as a layered solid material, whereas deeper inside, once pressures are larger than 50 kPa, it starts behaving as a viscous plastic flow. Consequently, accelerations of this underlying moving mass, sudden changes in slope, or obstacles in the bedrock can cause the top layers to fracture.

For every different type of crevasse (see Figure 10 and details in the subsequent paragraphs), we compute a probability map over the glacier surface based on its physical properties and observations made by glaciologists. These maps are used to stochastically place and model crevasses over the ice field.

The underlying crevasse formation processes take place at much smaller timescales than the full glacier simulation. Moreover, modeling crevasses remains an active research direction in glaciology, with prototypes based on fracture mechanics or continuum damage mechanics [Colgan et al. 2016]. Therefore, we rely on a set of procedural placement rules inspired by observations according to glacier type and appearance.

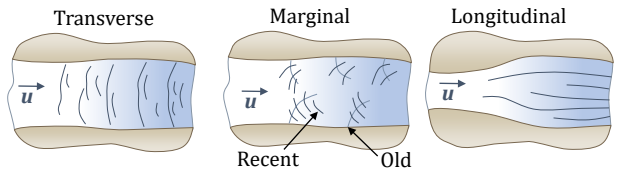

Fig. 10. Different types of crevasses (adapted from [Clowes and Comfort 1987]): transverse crevasses are perpendicular to the direction of the flow, marginal crevasses include old crevasses bent in direction of the ice flow and recently formed ones, and longitudinal crevasses appear when the valley becomes wider and ice widens to fill it.

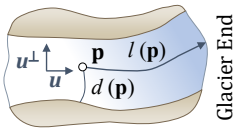

Fig. 11. Notations for crevasse generation.

Transverse crevasses are the most commonly found type. They form across glaciers, perpendicular to the direction of flow, due to tensile stresses when the ice accelerates or the slope steepens. The margins of the glacier body are usually free of this type of crevasse. Let $\nabla_{\hat{\mathbf{u}}}$ denote the directional derivative in the flow direction $\hat{\mathbf{u}} = \mathbf{u}/\|\mathbf{u}\|$. We compute a probability map $\mathcal{P}_{\mathcal{T}}$ across the ice surface by factoring in the flow acceleration $\nabla_{\hat{\mathbf{u}}}\mathbf{u}(\mathbf{p})$, the bedrock slope changes in the flow direction $\nabla^2_{\hat{\mathbf{u}}} b(\mathbf{p})$, the ice surface slope angle $\alpha(\mathbf{p})$, and the distance to the closest margin $d(\mathbf{p})$ (Figure 11):

$$\mathcal{P}_{\mathcal{T}}(\mathbf{p}) = \left(f_a(\nabla_{\hat{\mathbf{u}}}\mathbf{u}(\mathbf{p})) + f_b(\nabla^2_{\hat{\mathbf{u}}} b(\mathbf{p})) + f_s(\alpha(\mathbf{p}))\right) \cdot f_d(d(\mathbf{p}))$$

if $\tau(x) > 50\,\text{kPa}$, and $\mathcal{P}_{\mathcal{T}}(\mathbf{p}) = 0$ otherwise. The different f_i are functions that map the different domains of each magnitude to the unit range. In our implementation, we used cubic functions.

Longitudinal crevasses form when the valley widens and the ice slows down, due to compressive stresses in the flow direction and expansive in the perpendicular direction. The fracture is parallel to the flow direction.

We compute their probability of longitudinal crevasses $\mathcal{P}_{\mathcal{L}}$ across the ablation area of the glacier $\dot{m}(\mathbf{p}) < 0$, where ice is slowing in the flow direction $\nabla_{\hat{\mathbf{u}}}\mathbf{u}(\mathbf{p}) < 0$, accelerating perpendicularly $\nabla_{\hat{\mathbf{u}}^\perp}\mathbf{u}(\mathbf{p}) > 0$ and the valley width increases down flow. The probability is proportional to both the flow deceleration and parallel acceleration. Finally, since these crevasses occur more frequently near the end of glaciers, we use the flow distance l between the point $\mathbf{p}$ and the glacier limit to gradually reduce the probability according to $l(\mathbf{p})$. This distance is computed along with the flow, and differs from the Euclidean distance:

$$\mathcal{P}_{\mathcal{L}}(\mathbf{p}) = \left(f_a(-\nabla_{\hat{\mathbf{u}}}\mathbf{u}(\mathbf{p})) + f_p(\nabla_{\hat{\mathbf{u}}^\perp}\mathbf{u}(\mathbf{p}))\right) \cdot f_e(l(\mathbf{p}))$$

Marginal crevasses form due to shear stress in the ice close to the glacier margins, as it moves slower than the ice at the glacier center due to friction with the bedrock walls. They open at oblique angles with the flow direction, at approximately 45 degrees.

We place these crevasses below the equilibrium line, on nearly flat areas with low stress, close to the glacier margins. The probability decreases at the center of the glacier, and is proportional to the variation of stress towards the walls, *i.e.* perpendicular to flow direction:

$$\mathcal{P}_C(\mathbf{p}) = f_t(\nabla_{\hat{\mathbf{u}}^\perp}\tau(\mathbf{p})) \cdot (1 - f_d(d(\mathbf{p})))$$

if $\dot{m}(\mathbf{p}) < 0$, $\alpha(\mathbf{p}) < 10$ and $\tau(\mathbf{p}) < 50\,\text{kPa}$, otherwise $\mathcal{P}_C(\mathbf{p}) = 0$.

6.2 Rimayes

A rimaye, also known as *bergshrund*, is a particular type of crevasse that forms on the upper part of a cirque, at the interface between the moving body of the glacier and stagnant ice attached to the steep headwall (see Figure 12). Because they formed due to the downward flow of the glacier body, the opening is parallel to the walls and forming a right angle with the glacier flow direction.

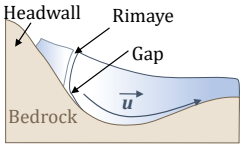

Fig. 12. Symbolic representation of a rimaye.

Contrary to other types of crevasses that are stochastically distributed over their presence region, we constrain rimaye models to locations $\mathbf{p}$ satisfying the following conditions: (1) $\mathbf{p}$ is located in the accumulation zone: $\dot{m}(\mathbf{p}) > 0$, (2) the surface slope angle is steep: $\alpha(\mathbf{p}) > 30$, (3) the basal stress is in an intermediate range between static and flowing ice: $30\,\text{kPa} < \tau(\mathbf{p}) < 60\,\text{kPa}$, (4) at least 3 of the 8 cell neighbors are located upstream and are also steep, (5) at least 2 of the 8 cell neighbors have a slope lower than a given threshold, 30 degrees in our implementation.

6.3 Moraines

Moraines are the accumulation of rocks, debris, and sediments transported by the glacier that emerge to the surface. It is possible to differentiate between different types of moraines: *lateral* moraines appear at the borders of the glacier, *medial* moraines form at the interface between two glacier flows that meet in a valley and travel side by side, and *terminal* moraines form at the limit of a glacier as the ice melts and deposits the transported materials.

To place medial moraines, we need to identify the different tributary flows in a glacier and detect where they meet. Moraines only appear in the ablation zone since fresh ice is continuously produced on top of the accumulation zone, covering debris.

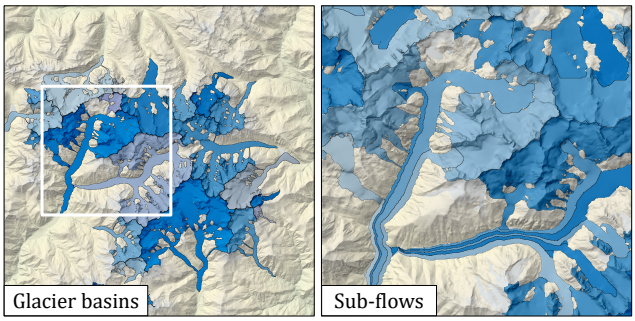

Fig. 13. Segmentation of glacier basins (left) and identified flows on one of the areas (right).

First, we segment and separate the different glacier drainage basins, *i.e.*, every subset of the ice surface that reaches the same

glacier limit. Existing drainage basin computation algorithms [Barnes et al. 2014] were designed for watersheds and are therefore not adapted to ice flow. Our method can be outlined as follows: starting from the lowest unlabeled point on the ice surface, we assign a new ice basin index to this point and propagate it to all higher glacier neighbors until no more points can be reached. By iterating this process, we separate all the different ice basins (see Figure 13, left).

Next, we subdivide each glacier basin into different flows (Figure 13, right). This is more challenging as we cannot rely on a flooding strategy as before: two tributary flows may have originated from the same upper area - for example, a glaciated mountain top flowing towards two valleys that merge downstream - or a flow might divide and meet again downstream creating a moraine from there, so we want to identify both branches.

To effectively separate the different flows in a basin, we compute the morphological skeleton S_G of the binary mask representing the glacier in ablation zone plus the glacier under a plastic flowing regime: $\{\dot{m}(\mathbf{p}) < 0\} \vee \{\tau(\mathbf{p}) > 50\,\text{kPa}\}$. Unlike river networks, the surface elevation of a glacier is not necessarily monotonically decreasing downstream so we cannot build the tributaries tree from the skeleton elevation. We define the main glacier sub-flow as the longest minimal path in S_G between its lowest node (near the glacier tongue) and any other node in S_G. Then, we iteratively classify tributary glaciers as the longest minimal path between an unvisited node in S_G and the set of visited nodes that form the current glacier network. This process stops when a maximum number of sub-flows is reached (10 in our experiments), or when the longest possible sub-flow is smaller than a threshold ($3-5$ km in our scenes).

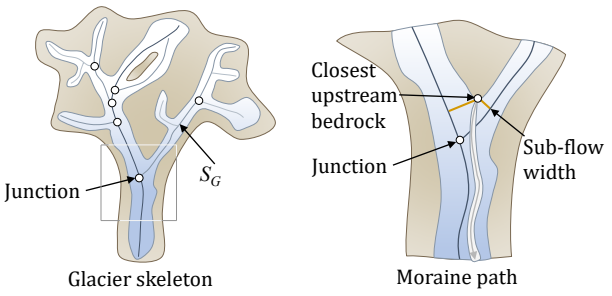

Fig. 14. Subdivision of the intersecting area of two glacier flows: split is proportional to the ratio between sub-flow widths at tributary points.

Once the tributary glaciers and their corresponding junction points are identified (see Figure 14), we recursively visit these junctions in a bottom to top order, *i.e.* starting from the one closest to the glacier tongue. We locate the upstream rock wall that is closest to the junction point, and split the downflow glacier surface into two parts such that the respective widths are proportional to the widths of each tributary before merging. These widths are estimated as twice the minimum distance between the upstream rock wall point and the skeleton of each sub-flow. Finally, a medial moraine is placed at each interface between different sub-flows.

6.4 Seracs and icefalls

Seracs are fractured blocks of ice, formed at the intersection of several crevasses or abrupt discontinuities and steep slopes over the surface of the glacier such as cliffs or icefalls.

We define a probability map proportional to the number of neighbors of a cell that present ice discontinuities in the stacked grid representation. In particular, there is a discontinuity between a cell $\mathbf{p}$ and its neighbor $\mathbf{n}$ if $\mathcal{S}(\mathbf{p}) < \mathcal{B}(\mathbf{n})$. Icefalls are defined as the ice surface regions with slope and basal stress τ above a given threshold (we use 15 degrees for the slope, and 100 kPa for basal stress).

6.5 Ogives

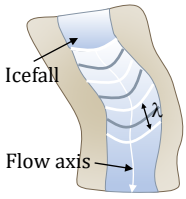

Fig. 16. Symbolic representation of ogives.

Ogives, also called Forbes bands, are alternating arc-shaped wave crests and valleys on the ice surface that visually form light and dark bands. Always forming below icefalls, although not all icefalls produce ogives, they propagate down the glacier until they eventually fade out or reach the glacier limit [Cuffey and Paterson 2010]. The period between two consecutive bands λ corresponds to the distance the glacier progresses in a year (Figure 16).

We use the ice falls map (Section 6.4) to estimate ogives location. An ogive forms if the base of an icefall is wide enough (> 100 m in our implementation) and if the icefall flow direction is aligned with that of the glacier beneath it. We use the geodesic distance to the base of the icefall $l(\mathbf{p})$ to parametrize the band pattern, and combine it with the distance to the ogive medial axis $d(\mathbf{p})$ to warp it backward to obtain arc-shaped patterns. Their effect progressively disappears as ogives travel downflow.

Ogives that form on a glacier sub-flow do not affect other parallel flows separated by moraines: therefore we use the segmentation used for placing medial moraines (Section 6.3) to limit the side extent of ogives.

7 RESULTS

We implemented our glacier simulation in a compute shader embedded in a C++ and OpenGL application, allowing interactive simulation and authoring of glaciers. The user may pause, resume or restart the simulation at any moment, modify the physical parameters (equilibrium line, accumulation rates) at any time and observe how the glacier grows or retreats, and interact with brush tools allowing to sculpt the bedrock layer, add or remove ice locally or edit the control maps (see accompanying video). The feature placement was implemented in Python using OpenCV and SciPy. Features maps were used to define the materials, textures and displacement maps, which in turn were streamed to E-on Vue® software to produce the photorealistic renderings (Figures 1, 5, 15, 17, 21, 22, 23, 24). Features were instantiated using a texture bombing approach to change the albedo of crevasses, rimayes and ogives, carve the surface elevation (moraines, crevasses), add noise onto the surface (seracs, icefalls), or change the material (moraines).

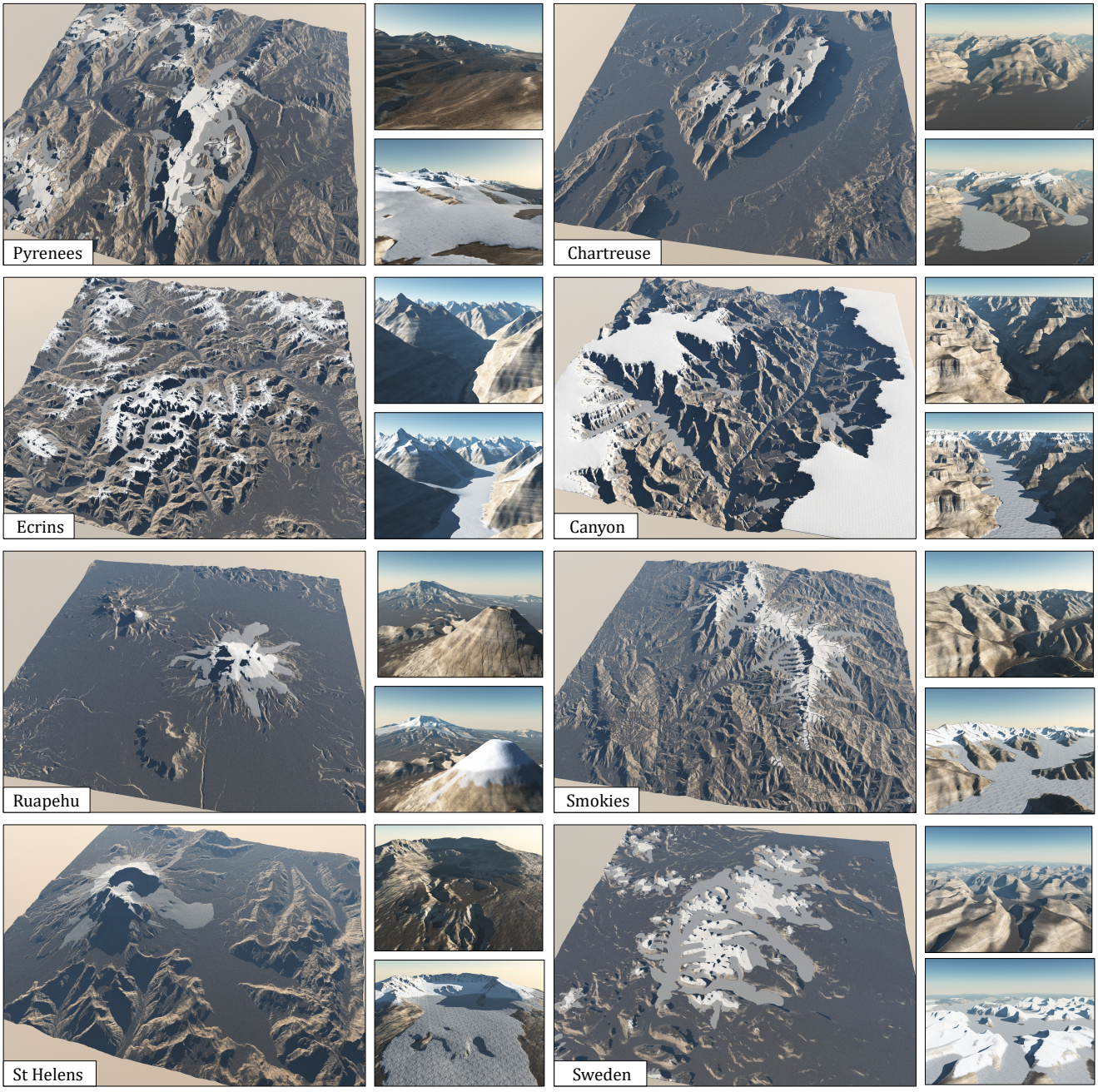

Fig. 15. Glaciers simulated over a variety of reliefs: close-ups show the bare terrain (top) and with the ice sheet generated by our simulation (bottom).

In our context, texture bombing consists of stamping features onto a map (depth, albedo) using alpha blending masks for delimiting the footprint of the features. Texture stamps used for bombing crevasses were generated manually from real pictures, other glacier features procedurally. Our objective was to achieve reasonable rendering quality and demonstrate the plausibility of our placement maps. Industrial production quality and more physically accurate modeling would require application and artists' intervention.

Our implementation can be found at the following repository: https://github.com/oargudo/glaciers.

Table 1. Comparison of execution time and relative error according to the initial resolution. t_s denotes the computation time to reach steady state for the simulation on the starting, while t_{up} denotes the time of upsampling steps. Root Mean Square Error (RMSE) and Volume difference were computed with respect to the ice layer obtained from the 20 m simulation.

Res.	Grid	Years	$t_s(s)$	$t_{up}(s)$	RMSE	Volume
120 m	250^2	631	17	103	2.38 m	+1.2%
60 m	500^2	629	287	62	1.69 m	−0.6%
40 m	750^2	632	1530	55	0.95 m	−0.3%
20 m	1500^2	635	26722	–	–	–

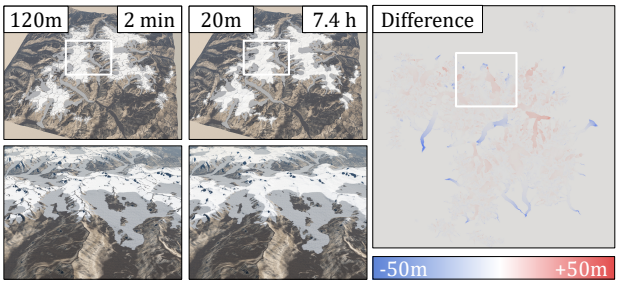

Fig. 17. Comparison between the glacier computed with a coarse grid (120 m cells) and refined using our multiresolution approach, and the glacier directly computed on a precise grid (20 m cells). The right image shows the ice thickness difference between the multiresolution and precise grids.

7.1 Performance and validation

Known as computationally intensive, glacier simulations are usually restricted to small domains with limited grid sizes. Our interactive multiresolution approach accelerates computations by several orders of magnitude and generates high-resolution ice thickness data at the expense of negligible approximations, compared to a direct brute force implementation (see Figure 17). This allows both to process large terrains and use higher grid resolutions.

Table 1 reports the timings and errors for different initial resolutions (final resolution is always 20 m) and demonstrates the effectiveness of the multiresolution approach. From a given initial cell resolution, we simulated the ice evolution until reaching a steady-state such that $\partial h/\partial t \leq \epsilon^n = 1$ mm. For the successive correction steps during upsampling, we relaxed the steady-state condition to $\epsilon^i = r^n/r^i \cdot \epsilon^n$, where r^n refers to the low-resolution cell size (120 m in our experiments) and r^i to the upsampled size. This linear factor serves as a trade-off between expensive correction steps if we keep $\epsilon^i = \epsilon^n$, and poor accuracy if ϵ^i is scaled quadratically like the number of cells. Moreover, we always simulate at least one year of evolution during the correction step.

As expected, the error decreases when using higher resolutions, but at a much slower rate than the computation time increases. Figure 17 shows a comparison between the ice cover obtained using our multiresolution simulation, and a direct brute force high-resolution simulation. While the ice covers slightly differ, the error remains small ($\approx$ 1%) and the multiresolution performs orders of magnitude faster (×220).

Table 2 reports the performance of the simulations presented throughout the paper, indicating the time t_s until steady state is reached in the initial resolution grid, as well as the time t_{up} required for the upsampling and correction steps. In all cases, we used 2 or 3 upsampling steps. Recall that the simulation time step Δt depends on cell resolution, as well as the maximum ice diffusivity over the terrain, which in turn depends on the ice thickness. Moreover, the number of years required to reach a steady-state depends on its terrain morphology and the glaciology parameters. Therefore, we also indicate the number of years simulated until steadiness, the maximum ice thickness of the glacier, and the total ice volume, as they are related to execution times.

While the steadiness thresholds defined above seemed reasonable for most tested scenes, cases arose where the simulation did not show any apparent changes before the stop criteria where met as well as cases were some parts of the glacier were still advancing. In an interactive editing session, however, the user may intervene and stop or keep running the simulation as desired. We noticed that for vast landscapes, some glacier parts may have reached steady-state while others are still advancing, which suggests that a spatially-adaptive simulation scheme could be worth investigating as future work to accelerate simulations.

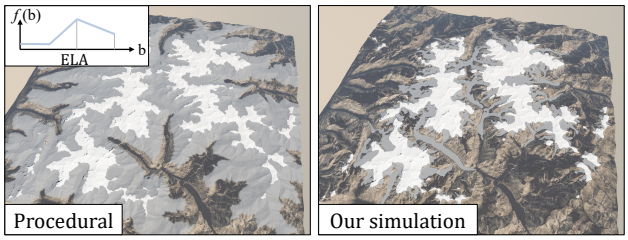

Fig. 18. Comparison of our method to a procedural approach.

Figure 18 compares our method to a procedural approach defining the ice thickness as $h(\mathbf{p}) = f(b(\mathbf{p})) \cdot g(\alpha(\mathbf{p}))$, where $f(b(\mathbf{p}))$ is a two-step linear function with a maximum reached for $b(\mathbf{p}) = ELA$, and $g(\alpha(\mathbf{p})) \in [0, 1]$ linearly reduces ice thickness to 0 as the bedrock slope increases and reaches a $\alpha \geq 60$ degree limit. Procedural method characterizing the ice thickness only according to the position, elevation, and slope fail to capture the downflow of ice tongues constrained by valley walls.

From a visual perspective, our method is capable of generating realistically looking glaciers on a variety of terrains, as illustrated in Figure 15. Depending on the choice of parameters, we can obtain mountain glaciers (Ecrins, Pyrenees, Smokies), ice fields (Chartreuse, Sweden), or ice caps (Ruapehu). The procedural rules for feature placement were improved after some iterations with an expert on glaciers, to generate placement maps as consistent as possible with reality. Figure 24 shows a side-by-side comparison of several glacier forms and modeled features.

7.2 Control

From an authoring perspective we offer two forms of user control: simulation parameters and ice and terrain sculpting. Glaciologists adjust the models to real glaciers by finding an appropriate set

of values corresponding to the different physical parameters and constants required in the simulation, especially the mass balance function $\dot{m}$ which describes where and how much snow is accumulated or melted, and the constants Γ_s and Γ_d that regulate the sliding and deformation speeds of the ice mass. We take advantage of this by introducing simple and intuitive controls over the simulation.

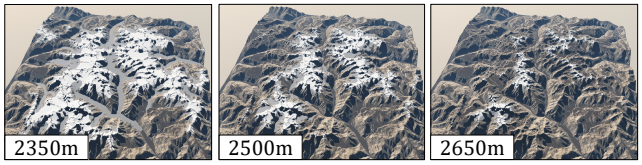

Fig. 19. Influence of the ELA values over the final glacier shape: slight modifications drastically change the glacier extents in the valleys.

The parameter with the most noticeable impact is the *equilibrium line altitude* ELA that determines the elevation above which snow accumulates and creates glaciers. Control of this parameter can be achieved by setting a constant value or providing a control map that allows the deviation from this value. Slight changes to the equilibrium line produce substantial effects on the final glacier shape as depicted in Figure 19. The control map can be either provided by the user or computed from yearly sunlight irradiance onto the terrain (Figure 20). Studies in glaciology estimate deviations due to orientation in the order of 70 to 320 m [Evans and Cox 2005], so we used similar ranges.

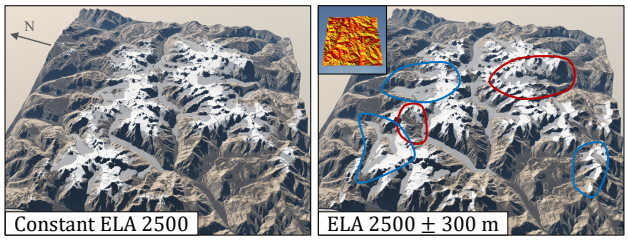

Fig. 20. Using a sunlight map to locally deviate the ELA. Glaciers on north faces appear thicker and longer, whereas fewer glaciers start from south faces.

The mass balance function $\dot{m}$ also has a significant impact, as it determines how far a glacier extends below equilibrium line. Although this function can be very complex to account for all the phenomena that cause accumulation and ablation, we selected the approach in [Oerlemans 1986] and defined a two-step linear function:

$$\dot{m}(\mathbf{p}) = \begin{cases} \beta \cdot (s(\mathbf{p}) - e(\mathbf{p})) & \text{if } s(\mathbf{p}) > e(\mathbf{p}) \\ \gamma \cdot (s(\mathbf{p}) - e(\mathbf{p})) & \text{if } s(\mathbf{p}) < e(\mathbf{p}) \end{cases}$$

where the constants β and γ are the accumulation and ablation rate gradients, above and below ELA, respectively. This follows the intuition that more snow accumulates at higher altitudes whereas ice melts at lower altitudes.

The glacier extent principally depends on the ratio between these two slopes. Strong accumulation leads to longer glacier tongues extending below the accumulation area, whereas strong ablation produces glaciers quickly melting and disappearing after leaving the accumulation area. Scaling both parameters by the same constant produces similar glacier coverages, simply with a greater overall mass of ice (see Figure 21). It is also possible to define local variations of β using a *precipitation map* $\mathcal{P}(\mathbf{p})$ to scale the accumulation rate $\beta(\mathbf{p}) = \beta \cdot \mathcal{P}(\mathbf{p})$, as illustrated in Figure 22.

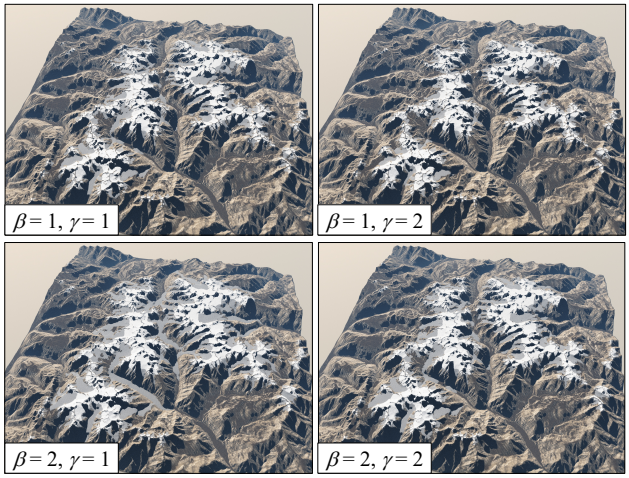

Fig. 21. Effects of accumulation and ablation ratios β and γ: higher values of accumulation lead to longer glacier tongues, whereas strong ablation values lead to smaller glaciers by speeding-up melting. Values in $\text{mm} \cdot \text{year}^{-1} \cdot \text{m}^{-1}$.

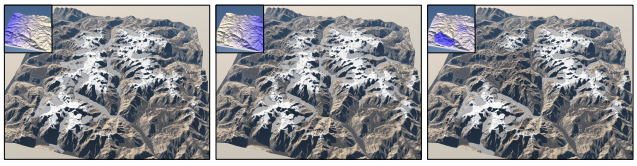

Fig. 22. Influence of accumulation maps: left and middle linear ramps represent conditions where snow fronts enter from either the north or the south. The right map was painted as a sketch to modify the accumulation locally on each massif, confirming the authoring ability of our method.

Finally, regarding the constants for ice speed, we fixed their value following [Headley et al. 2012] as $\Gamma_s = 3.27\,\text{y}^{-1}\text{m}^{-1}$ and $\Gamma_d = 7.26 \cdot 10^{-5}\,\text{y}^{-1}\text{m}^{-3}$ that correspond to alpine-like glaciers. Polar or cold glaciers can also be simulated by adjusting these constants and reducing the sliding contribution. We provide intuitive control over these parameters with a simple scalar that properly scales the constants from cold to temperate glacier. Figure 23 shows a comparison of two simulations with the same parameters except that only one velocity component is used in each case. Cold glaciers (deformation only) produce more voluminous ice masses that also advance more slowly, whereas temperate glaciers (sliding only) are more affected by steep relief and create thinner but longer flows.

Most of the examples shown through the paper did not require user intervention or edition. However, we developed brush tools to control and modify the bedrock layer, the ice layer, and both

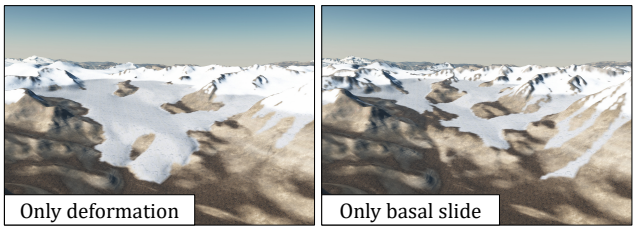

Fig. 23. Simulating only the deformation or the sliding components of ice velocity influences the overall ice volume and velocity of the glacier.

the ELA and accumulation maps directly on the terrain while the simulation is running. Recall that the steady-state of a glacier does not depend on the initial or current amount of ice (Figure 5), but only on the relief morphology and the glaciology constants. Therefore, the impact of the ice tool progressively disappears throughout time.

7.3 Limitations

One of the biggest limitations of the SIA model for glaciers is that it represents the bedrock and ice layers as continuous derivable functions. This implies that overhanging blocks of ice, like seracs, are impossible to obtain through the simulation. Although our procedural amplification locates where such features occur, we used a high-resolution layered model combined with textures for our renders, thus our scenes do not show overhangs. The complex geometry of crevasses and seracs would be better captured by placing three-dimensional models; for example, procedurally defined implicit surfaces allowing for blending and carving operations. However, modeling individual feature geometries was out of the scope of this paper.

We restricted our work to valley glaciers and ice fields and did not investigate the simulation of ice shelves that are ice sheets spreading into the ocean, forming floating ice platforms still attached to the grounded ice. Simulating ice shelves would require accounting for the interactions between the ocean water and ice, which was beyond our target scope.

8 CONCLUSION

In this paper, we proposed a hybrid technique combining a novel multiresolution SIA with procedural methods for generating high-resolution glaciers featuring realistic details such as moraines, ogives, icefalls, and crevasses. To the best of our knowledge, we are the first to introduce a fully interactive glacier simulation system compatible with existing terrain generation frameworks and pipelines. Our work opens several avenues for future research.

Although our system allows for simulating the formation and the evolution of glaciers throughout time, the considered time scales are not large enough to model bedrock erosion that produces specific landforms such as hanging valleys. Combining the ice flow simulation with erosion would be an interesting research direction worth investigating.

In this work, we focused on a subset of glacier features connected with the ice flow. Retracing glaciers usually create a frontal moraine from the accumulation of transported debris on its limit, which acts as a dam retaining the water melting, forming a lake. Modeling such frontal moraines and lakes could be a direct extension of this work: our simulation could record geomorphological parameters to determine how far glaciers advanced and where to generate lakes.

Glacier features are instantiated at a given time step according to parameters of the mass of ice. In reality, crevasses are in constant movement: as the glacier flows, new crevasses open whereas others close due to compression stress, some break apart into seracs that fall along cliffs or icefalls, eventually triggering avalanches. Simulating the complex ice dynamics would require accurate spatial and temporal scales (less than one meter, less than one hour), and therefore remains a challenging realistic animation problem.

Finally, from a rendering perspective, interesting challenges include modeling the complex shape of crevasses and seracs, or capturing the changing material properties of ice depending on its level of compactness.

ACKNOWLEDGMENTS

We thank the reviewers for their helpful comments, Jordi Camins for his advice on modeling the glacier features, and credit Jürg Alean and Michael Hambrey for allowing to use their photographs. This work is part of the project HDW ANR-16-CE33-0001, supported by Agence Nationale de la Recherche. We would like to credit E-On software for providing Vue for rendering our terrain models and Adobe Substance for providing us Substance 3D licenses.

REFERENCES

Richard Barnes, Clarence Lehman, and David Mulla. 2014. Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models. *Computers and Geosciences* 62 (2014), 117–127.

Matthew M Bennett and Neil F Glasser. 2011. *Glacial geology: ice sheets and landforms*. John Wiley & Sons.

Robert Bindschadler. 1983. The Importance of Pressurized Subglacial Water in Separation and Sliding at the Glacier Bed. *Journal of Glaciology* 29, 101 (1983), 3–19.

J. Braun, D.Zwartz, and J.H. Tomkin. 1999. A new surface-processes model combining glacial and fluvial erosion. *Annals of Glaciology* 28 (1999), 282–290.

Ed Bueler. 2016. Numerical modelling of glaciers, ice sheets, and ice shelves (course notes).

Yanyun Chen, Hanqiu Sun, Hui Lin, and Enhua Wu. 2003. Modelling and rendering of snowy natural scenery using multi-mapping techniques. *Journal of Visualization and Computer Animation* 14, 1 (2003), 21–30.

Aland Clowes and Peter Comfort. 1987. *Process and landform: an outline of contemporary geomorphology*. Oliver & Boyd.

William Colgan, Harihar Rajaram, Waleed Abdalati, Cheryl McCutchan, Ruth Mottram, Mahsa S. Moussavi, and Shane Grigsby. 2016. Glacier crevasses: Observations, models, and mass balance implications. *Reviews of Geophysics* 54, 1 (2016), 119–161.

Guillaume Cordonnier, P. Ecormier, Eric Galin, James Gain, Bedrich Benes, and Marie-Paule Cani. 2018. Interactive Generation of Time-evolving, Snow-Covered Landscapes with Avalanches. *Computer Graphics Forum* 37, 2 (2018), 497–509.

Kurt M Cuffey and William Stanley Bryce Paterson. 2010. *The physics of glaciers*. Academic Press.

François Dagenais, Jonathan Gagnon, and Eric Paquette. 2016. An Efficient Layered Simulation Workflow for Snow Imprints. *The Visual Computer* 32, 6-8 (2016), 881–890.

David L Egholm, Mads F Knudsen, Chris D Clark, and Jerome E Lesemann. 2011. Modeling the flow of glaciers in steep terrains: The integrated second-order shallow ice approximation (iSOSIA). *Journal of Geophysical Research: Earth Surface* 116, F2 (2011).

Ian S. Evans and Nicholas J. Cox. 2005. Global variations of local asymmetry in glacier altitude: separation of north-south and east-west components. *Journal of Glaciology* 51, 174 (2005), 469–482.

Paul Fearing. 2000. Computer Modelling of Fallen Snow. In *Proceedings of Siggraph*. ACM Press/Addison-Wesley Publishing Co., 37–46.

Niels Festenberg and Stefan Gumhold. 2011. Diffusion-Based Snow Cover Generation. *Computer Graphics Forum* 30, 6 (2011), 1837–1849.

Fig. 24. Side by side comparison of different modeled features. Our procedural placement combined with texture bombing can successfully reproduce a vast variety of small-scale glacier features. Picture credits: Alaska NPS pictures shared as public domain (https://www.flickr.com/alaskanps), Jürg Alean and Michael Hambrey pictures used with permission (https://www.swisseduc.ch/glaciers), Jordi Camins picture used with permission (http://www.gelicehielo.com), Céline Mounier picture shared as CC-BY (https://www.flickr.com/celinemyriam/7787161490).

Table 2. Execution parameters, timings and results for the different glaciers simulated, grid size in meters (m).

Figure: terrain	km	ELA (m)	β	γ	Coarse grid	Years	t_s (s)	years/s	Fine grid	t_{up}(s)	Max h (m)	Ice km^3
15: Chartreuse	50	1400 ± 50	3	1	250 (200)	946	76	12.4	1250 (40)	152	341.5	14.0
15: Canyon	30	2000 ± 150	2	1	250 (120)	1407	145	9.7	1500 (20)	366	264.5	27.0
15: Ecrins	84	2600 ± 100	1	1	350 (240)	671	109	6.1	2800 (30)	2032	417.9	54.0
15: Pyrenees	30	2500 ± 150	2	1	250 (120)	416	25	16.6	1500 (20)	218	258.6	6.7
15: Ruapehu	30	2000 ± 150	4	2	250 (120)	323	23	13.8	1500 (20)	242	186.1	3.2
15: Smokies	30	1400	1	1	250 (120)	902	19	46	1500 (20)	52	190.9	3.3
15: St Helens	20	2000 ± 100	5	1	250 (80)	256	21	12	1000 (20)	74	115.3	1.2
15: Sweden	60	1400	5	1	200 (300)	3209	94	34	2400 (25)	815	556.4	81.0
19: left	30	2350	1	1	250 (120)	777	40	19.3	1500 (20)	192	223.8	11.0
19: middle	30	2500	1	1	250 (120)	621	7	92.7	1500 (20)	36	143.5	3.6
19: right	30	2650	1	1	250 (120)	794	3	240.8	1500 (20)	16	105.8	0.6
20: left	30	2500	2	1	250 (120)	631	17	36.1	1500 (20)	107	179.8	6.3
20: right	30	2500 ± 300	2	1	250 (120)	659	29	22.8	1500 (20)	184	187.9	7.1
21: top-left	30	2500	1	1	250 (120)	621	7	92.7	1500 (20)	36	143.5	3.6
21: top-right	30	2500	1	2	250 (120)	454	4	126.1	1500 (20)	29	119.5	2.7
21: bottom-left	30	2500	2	1	250 (120)	631	17	36.1	1500 (20)	107	179.8	6.3
21: bottom-right	30	2500	2	2	250 (120)	607	14	44.7	1500 (20)	83	173.8	4.7
22: left	30	2500	2	1	250 (120)	667	24	27.4	1500 (20)	126.8	185.2	6.2
22: middle	30	2500	2	1	250 (120)	642	22	28.9	1500 (20)	148.8	188.9	6.3
22: right	30	2500	2	1	250 (120)	544	21	25.8	1500 (20)	111.3	181.7	4.7
23: left	30	2500	1	1	250 (120)	1359	19	71.2	1500 (20)	85	203.7	9.2
23: right	30	2500	1	1	250 (120)	639	6	99.8	1500 (20)	30	148.9	3.6

N. V. Festenberg and S. Gumhold. 2009. A Geometric Algorithm for Snow Distribution in Virtual Scenes. In *Proceedings of EG Conference on Natural Phenomena*. 17–25.

David Foldes and Bedrich Benes. 2007. Occlusion-Based Snow Accumulation Simulation. In *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2007)*. The Eurographics Association, 35–41.

Eric Galin, Eric Guérin, Adrien Peytavie, Guillaume Cordonnier, Marie-Paule Cani, Bedrich Benes, and James Gain. 2019. A Review of Digital Terrain Modeling. *Computer Graphics Forum (proceedings of Eurographics 2019 STAR)* 38, 2 (2019), 553–577.

J. W. Glen and Max Ferdinand Perutz. 1955. The creep of polycrystalline ice. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 228, 1175 (1955), 519–538.

Francois Grosbellet, Adrien Peytavie, Éric Guérin, Eric Galin, Stéphane Mérillou, and Bedrich Benes. 2016. Environmental Objects for Authoring Procedural Scenes. *Computer Graphics Forum* 35, 1 (2016), 296–308.

Rachel M. Headley, Gerard Roe, and Bernard Hallet. 2012. Glacier longitudinal profiles in regions of active uplift. *Earth and Planetary Science Letters* 317-318 (2012), 354–362.

Tommy Hinks and Ken Museth. 2009. Wind-driven snow buildup using a level set approach. In *EG Ireland Workshop Series*. 19–26.

Richard Huggett. 2016. *Fundamentals of geomorphology*. Routledge Fundamentals of Physical Geography.

Kolumban Hutter. 1983. *Theoretical glaciology; material science of ice and the mechanics of glaciers and ice sheets*. D. Reidel Publishing Company, Dordrecht, The Netherlands.

A. H. Jarosch, C. G. Schoof, and F. S. Anslow. 2013. Restoring mass conservation to shallow ice flow models over complex terrain. *The Cryosphere* 7, 1 (2013), 229–240.

Mark A. Kessler, Robert S. Anderson, and Greg M. Stock. 2006. Modeling topographic and climatic control of east-west asymmetry in Sierra Nevada glacier length during the Last Glacial Maximum. *Journal of Geophysical Research: Earth Surface* 111, F2 (2006).

Wouter Knap, Johannes Oerlemans, and Martin Cabée. 1996. Climate sensitivity of the ice cap of King George Island, South Shetland Islands, Antarctica. *Annals of Glaciology* 23 (1996), 154–159.

MW Mahaffy. 1976. A three-dimensional numerical model of ice sheets: Tests on the Barnes Ice Cap, Northwest Territories. *Journal of Geophysical Research* 81, 6 (1976), 1059–1066.

N. Maréchal, E. Guérin, E. Galin, S. Mérillou, and N. Mérillou. 2010. Heat Transfer Simulation for Modeling Realistic Winter Sceneries. *Computer Graphics Forum* 29, 2 (2010), 449–458.

Claus Moeslund, Thomas anvisud Madsen, Michael Aagaard, and Dennis Lerche. 2005. Modeling falling and accumulating snow. In *Vision, Video and Graphics*. The Eurographics Association.

Tomoaki Moriya and Tokiichiro Takahashi. 2010. A Real Time Computer Model for Wind-Driven Fallen Snow. In *ACM SIGGRAPH ASIA 2010 Sketches*. 26:1–26:2.

Kazunobu Muraoka and Norishige Chiba. 2000. Visual Simulation of Snowfall, Snow Cover and Snowmelt. In *Proceedings of the Parallel and Distributed Systems: Workshops (ICPADS '00)*.

Tomoyuki Nishita, Hiroshi Iwasaki, Yoshinori Dobashi, and Eihachiro Nakamae. 1997. A Modeling and Rendering Method for Snow by Using Metaballs. *Computer Graphics Forum* 16, 3 (1997), C357–C364.

John Frederick Nye. 1957. The distribution of stress and velocity in glaciers and ice-sheets. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 239, 1216 (1957), 113–133.

J Oerlemans. 1986. An attempt to simulate historic front variations of Nigardsbreen, Norway. *Theoretical and applied climatology* 37, 3 (1986), 126–135.

Simon Premože, William B. Thompson, and Peter Shirley. 1999. Geospecific Rendering of Alpine Terrain. In *Eurographics Workshop on Rendering (EGWR'99)* (Granada, Spain). 107–118.

D. T. Reynolds, S. D. Laycock, and A. M. Day. 2015. Real-time Accumulation of Occlusion-based Snow. *The Visual Computer* 31, 5 (2015), 689–700.

P L Roe. 1986. Characteristic-Based Schemes for the Euler Equations. *Annual Review of Fluid Mechanics* 18, 1 (1986), 337–365.

Wong Sai-Keung and Fu I-Ting. 2015. Hybrid-based Snow Simulation and Snow Rendering with Shell Textures. *Computer Animation and Virtual Worlds* 26, 3-4 (2015), 413–421.

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A Material Point Method for Snow Simulation. *ACM Transactions on Graphics* 32, 4 (2013), 102:1–102:10.

Robert W. Sumner, James F. O'Brien, and Jessica K. Hodgins. 1999. Animating Sand, Mud, and Snow. *Computer Graphics Forum* 18, 1 (1999), 17–26.

Kohe Tokoi. 2006. A shadow buffer technique for simulating snow-covered shapes. In *International Conference on Computer Graphics, Imaging and Visualisation*. IEEE Computer Society, 310–316.

Bram van Leer. 1979. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.* 32, 1 (1979), 101 – 136.

Changbo Wang, Zhangye Wang, Tian Xia, and Qunsheng Peng. 2006. Real-time Snowing Simulation. *The Visual Computer* 22, 5 (2006), 315–323.