DOI: 10.1111/cgf.15059
EUROGRAPHICS 2024
A. Aristidou and R. McDonnell
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 2
STAR – State of The Art Report

Snow and Ice Animation Methods in Computer Graphics

Prashant Goswami[†]

Blekinge Institute of Technology, Sweden

Abstract

Snow and ice animation methods are becoming increasingly popular in the field of computer graphics (CG). The applications of snow and ice in CG are varied, ranging from generating realistic background landscapes to avalanches and physical interaction with objects in movies, games, etc. Over the past two decades, several methods have been proposed to capture the time-evolving physical appearance or simulation of snow and ice using different models at different scales. This state-of-the-art report aims to identify existing animation methods in the field, provide an up-to-date summary of the research in CG, and identify gaps for promising future work. Furthermore, we also attempt to identify the primarily related work done on snow and ice in some other disciplines, such as civil or mechanical engineering, and draw a parallel with the similarities and differences in CG.

CCS Concepts

• ***Computing methodologies*** → ***Physical simulation; Procedural animation;***

1. Introduction

Snow and ice form an essential ingredient of many scenes in CG, both indoors and outdoors. The use of snow in certain scenes is crucial to convey the feeling of cold winters with snow-laden landscapes or to capture the interaction of agents with snow in these environments. Similarly, the presence of ice on frozen tree branches or other objects can enhance the realism value of scenes based on winter settings. Several Hollywood movies have leveraged the power of CG to create or enhance the visual impact of snow and ice in varied environments. Recently, a few computer games have also been developed based on winter settings, often necessitating the interaction of game characters with snow in the environment. Besides the visual impression, CG has also aided in reproducing a faithful interaction of the virtually created snow with real-world objects in these applications.

Snow and ice are made of the same material but differ in composition. A major challenge while dealing with snow is capturing the varied visual appearance and behavior under diverse conditions, such as dry or wet, chunky or powdered snow. A similar challenge is confronted in the case of ice, wherein several environmental factors govern the process of ice formation or melting, and hence its visual appearance. Furthermore, owing to the demanding nature of computations, efficient snow and ice simulation is still formidable even with modern hardware.

It is well known that various algorithms to simulate materials or matter in CG are often inspired by those in disciplines like mechanical and civil engineering, agricultural science, physics, etc. The models developed in these fields are essential to study and validate real-life test cases insofar as snow is concerned, to the required accuracy. The desired accuracy is crucial to adhere to, even at the cost of large computational overheads. However, within the purview of CG, the focus and priorities while simulating matter differs. In these applications, the overall bar for visual accuracy can be lowered and traded off for the sake of efficiency. In most cases, this is achievable without significantly sacrificing the perceived visual quality. Hence, in order to do proper justice to this state-of-the-art report, it is crucial also to obtain a glimpse into the major snow/ice-based simulation methods followed in other disciplines.

This state-of-the-art report reviews existing methods that deal with the animation of snow and ice in CG. To this end, we begin this report by first briefly identifying the existing work and state of the art in the base literature dealing with the snow and ice mechanics in Sec. 2, since most of the research on snow and ice in CG is inspired by basic and structural sciences research. To account for a wide range of methods targeted for animating snow and ice, we deal with snow and ice separately. In the second part (Sec. 3), the report reviews the existing methods on snow in CG. In the next part (Sec. 4), we discuss the existing techniques dealing with the ice processes. In all these above three sections, we have grouped the identified works into several relevant categories. We finally conclude this report after discussing (Sec. 5) our findings from the included work. This includes identifying similarities and differences in the treatment of snow in different fields and existing gaps in CG that can inspire future works.

[†] prashant.goswami@bth.se

2. Base Literature

Various disciplines in the scientific community have dealt extensively with studying mechanical properties and determining the micro-level structure of snow or ice. Some of these methods have proved instrumental in shaping the state of the art in CG. The focus of work included in this section is varied, including mechanical, civil, or agricultural engineering, but not CG. The primary motivation in these fields is to achieve accurate modeling of the phenomena to capture the physical accuracy of material behavior, followed by rigorous validation, as is accepted in different fields. An early review of snow mechanics, including several prevalent models, is given by Mellor [Mel74, Mel77]. Features like deformation, elastic properties, viscoelastic properties, failure, friction, adhesion, etc., have been discussed from the mechanical point of view. Petrovic [Pet03] also reviews the mechanical properties based on the micro-scale physical properties of snow.

Fig. 1 outlines the organization of this section based on the underlying published work in the base literature on snow and ice. We begin this section by providing a brief technical insight into the physical nature and properties of snow in Sec. 2.1 and the underlying physical models. In Sec.2.2, simulation models dealing with different scales, micro and macro, are discussed. Various types of avalanches (dry and wet), associated methods, and other special features are covered in Sec. 2.3. For brevity, we defer to introducing a more detailed physical formulation of each model to the respective explanations. In order to respect the length constraints of this report and enhance its usefulness for beginners in the field, we have kept the mathematical formulations at a higher level while introducing the foundation and deriving insights from various ideas. References are provided to delve deeper into the discussed works at all points.

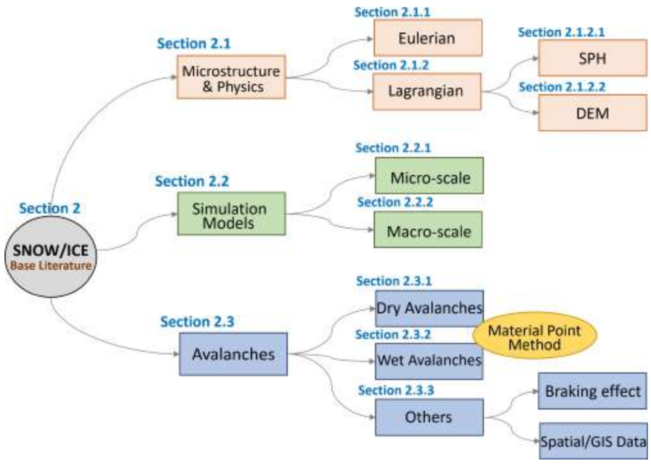


Figure 1: *Based on the published work, outline of the organization of the sections on base literature in this report.*

2.1. Microstructure and physics

Snow is a visco-elastic material exhibiting varying properties under different conditions. A majority of the literature refers to both snow and ice (and other intermediate forms) as snow. The reason for the varying properties lies in its internal crystal structure at the micro-level [Bak19] (Fig. 2a). Kinosita [Kin67] proposed that plastic deformation creates internal changes mainly in dislocations within the ice grains, while brittle deformation takes place as a result of disjointing of the network connections (Fig. 2b). Snow crystals form an interlocking aggregate structure as they precipitate. These crystals gradually transform their shapes and consistency after deposition on the ground. This further leads to the formation of ice bonds when coming into close contact with each other by the transportation of the water molecules. In a simplified sense, this physical process of bond formation between grains in the snowpack is termed as *sintering* [Ras05] (Fig. 2c). Any natural snow cover solidifies with time, resulting in a denser mixture containing ice particles consolidated due to sintering. Temperature is naturally another important physical parameter.

Apart from the hardening factor and the internal friction, the surface irregularities of deposited and pulverized snow composed of varying snow crystals are attractive to the community. In order to capture plasticity and deformation, most mathematical models require one or several parameters like stress, strain, cohesive strength, potential energy, friction (or other equivalent parameters), depending on the underlying physical model employed. In some models, consistent with other granular materials, the angle of repose (AOR), the steepest angle relative to the horizontal plane on which the snow can be piled without slumping, is required to solve equations dealing with snow dynamics. Experiments studying micrometrical properties, uniaxial compression with varying impacts and speeds, snow-object interaction (for instance, snow-wall or snow-tire), crack, and phase transition are of practical interest and, hence, standard in snow mechanics.

In the following, we briefly introduce the fundamentals of a few crucial models employed to simulate several physical phenomena in nature, forming the basis of most snow and ice simulation algorithms in the base fields and CG. To begin with, the general compressible flow for a viscid fluid is described using the Navier-Stokes equations [Bre05, McL12], Eq. 1.

$$\rho\left(\frac{\partial \boldsymbol{v}}{\partial t}+(\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v}\right)=-\boldsymbol{\nabla}p+\mu\boldsymbol{\nabla}\cdot\left\{\boldsymbol{\nabla}\boldsymbol{v}+(\boldsymbol{\nabla}\boldsymbol{v})^T-\frac{2}{3}(\boldsymbol{\nabla}\cdot\boldsymbol{v})\boldsymbol{I}\right\}$$
$$+\boldsymbol{f}_{ext},\qquad \boldsymbol{\nabla}\cdot\boldsymbol{v}=0 \tag{1}$$

Here $\boldsymbol{v}$ is the velocity of the fluid, $\frac{\partial \boldsymbol{v}}{\partial t}$ partial derivative or evolution of velocity with time, ρ density, p pressure, μ is the dynamic viscosity of the fluid, $\boldsymbol{I}$ identity tensor, and gravity and other external forces are accounted for in $\boldsymbol{f}_{ext}$. In essence, while the first equation in Eq. 1 conserves momentum, the second one ensures that the flow is divergence-free (also conserves mass). The compressible flow formulation can be simplified to obtain the incompressible variation. These equations form the basis of many snow simulation methods employing different base models.

2.1.1. Eulerian

The Eulerian or grid-based method tracks physical quantities like velocity, density, etc., on fixed points inside the fluid simulation domain. A typical Eulerian solver constitutes a loop wherein computations are made using Navier-Stokes equations on all grid cells in the following order: determining the time step Δt for marching through time, fluid advection to advect physical quantities with Δt, pressure projection to maintain incompressibility also accounting for the boundary conditions, and finally, fluid surface tracking. The use of staggered grids is common in grid-based methods to improve numerical accuracy. As the name suggests, a staggered grid stores different physical quantities in different locations of the grid cell. For instance, pressure can be stored for its value at the center location of the cell, whereas velocity is at the center of faces. Stam [Sta99] introduced a unique backward tracking method for large time steps to keep the solver unconditionally stable, which makes it somewhat hybrid of Eulerian and Lagrangian methods. Such hybrid Eulerian/Lagrangian methods, known as semi-Lagrangian, have become quite popular in many disciplines.

2.1.2. Lagrangian

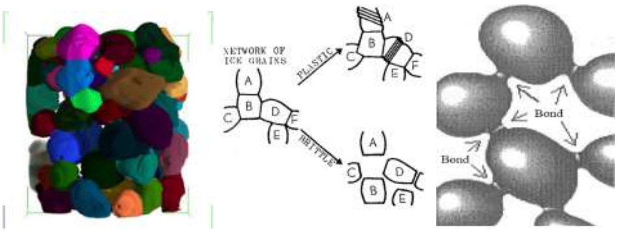
In contrast to the Eulerian approach, which deals with the concentration of a physical quantity while calculating its diffusion and convection in a given space, the Lagrangian approach deals with discretizing fluid as particles and computing particle trajectories. To the Lagrangian end, two main methods have been identified that are more popular in simulating snow and ice.

2.1.2.1. Smoothed particle hydrodynamics

Smoothed particle hydrodynamics (SPH) is a Lagrangian method that operates on particle-based discretization of the fluid. In contrast to static grid points, the particles carry physical quantities with them. A scalar quantity A at a point $\boldsymbol{r}$ is interpolated based on the weighted contributions of particles in its neighborhood $\boldsymbol{r}_j$, see below.

$$A(\boldsymbol{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\boldsymbol{r}-\boldsymbol{r}_j, h), \qquad p = k(\rho - \rho_0), \\ \boldsymbol{f}_i^p = -\sum_j m_j \frac{p_j}{\rho_j^2} \nabla W(\boldsymbol{r}-\boldsymbol{r}_j, h) \tag{2}$$

Here, j represents summation over neighboring particles falling within the core radius h, m_j is the mass of particle j, A_j the scalar quantity, ρ_j the density, and function $W(\boldsymbol{r}, h)$ is a smoothing kernel with finite support h. The density of a particle can also be obtained by substituting for A as ρ in the first equation of Eq. 2, which is used to determine the pressure p, and finally, the pressure force on particle i (derived from the Navier-Stokes equations), $\boldsymbol{f}_i^p$. Other forces, like viscosity, gravity, etc., are also incorporated into the overall formulation. Different smoothing kernels are often applied for different forces, for instance, pressure force, viscosity, etc. However, the motivation behind using these kernels is to compute the physical value of an attribute based on those of the neighboring particles within a finite support radius while also reducing their contribution with distance. The force equation in SPH is derived using the Navier-Stokes equations. An advantage of the SPH method is that mass conservation comes automatically with it. The presented formulation of equations corresponds to the weakly compressible version of SPH. Several CG methods have been developed recently to efficiently incorporate incompressibility using the SPH framework. Additionally, Lagrangian or hybrid approaches have a distinct advantage here since they are able to counteract solver inaccuracies and thus, in practice, have less volume loss than Eulerian approaches [KLTB19].



(a) (b) (c)

Figure 2: *Snow microstructure (a) snow grain structure visualized with overlapping spheres clumped together by Henry et al. [Bak19], (b) network of snow grains and deformation by Kinosita [Kin67], (c) sintering process explained in [Ras05].*

2.1.2.2. Discrete element method

Discrete element method (DEM) is a family of Lagrangian numerical methods for computing the motion of a large number of particles interacting with each other to predict bulk solids behavior [BL21]. It is often considered closely related to molecular dynamics (with simplifications) and offers the advantage of simulating millions of particles for increased resolution. Since DEM is a generic model, it provides the freedom to insert different forces' natures for different materials. This includes friction, gravity, plasticity, normal and tangential attractive or repulsive potentials, etc. In particular, DEM models have been shown to be effective in simulating a wide variety of granular materials.

More information on the Eulerian models is available in Braley and Sandu [BS10] and Stam [Sta23]. We refer the reader to the review by Monaghan [Mon12] and Koschier et al. [KBST22] for a detailed survey of SPH methods and to Guo and Curtis [GC15] for DEM-based granular simulation methods. To keep the nomenclature simple, in the remainder of this report, we will refer to the Eulerian methods as **grid-based**, Lagrangian methods as **particle-based**, and the semi-Lagrangian as **hybrid-based**.

2.2. Simulation methods

All the methods dealing with snow or ice in base fields can be broadly divided into two major categories. In the *micro-scale* approach, micrometrical physical parameters of the material and simulation play a crucial role in the process. Such methods usually focus on obtaining a more accurate behavior of the small-scale of the matter in question, often including its transformation during a particular time. On the other hand, the *macro-scale* approach is more helpful while dealing with snow deposition on large scales, such as determining the accumulation of snow on terrain. Regarding the underlying physical model, most snow mechanics methods either employ a grid-based or a DEM model based on the application and its requirements.

2.2.1. Micro-scale

Micro-scale models form the basis of early work dealing with studying the basic properties of snow. Kuroiwa et al. [KMT67] analyze the micrometrical properties of snow, such as the angle of repose, internal friction, hardening, etc. Their work aims to gain a better understanding of the consolidation and the sintering process of deposited snow. Mathematical models with various physical parameters are employed for the validation. In [Dou67], Doumani also deals with studying the micrometeorological parameters of snow. Their work, however, is centered around observing surface structures in the snow depositions, specifically created in and around the footprints, the impression of boots, etc. [MNP15] propose a micro-scale approach to predict the mechanical behavior of snow as a function of strain rate, density, and temperature applied. A validation study of dry snow throughout the ductile and brittle loading regimes based on this micro-mechanical approach is conducted in [MPN15].

A grid-based numerical method is proposed in Gaume et al. [GGT*18a] to model dynamic anticrack propagation in porous materials like snow. Their method is based on a new elastoplasticity model, assumes a large strain for crack dynamics computation, and simulates solid-fluid phase transitions in geomaterials. The model is verified through snow fracture experiments. [CJ05] propose an analytical expression to describe the temporal evolution of the microscale bonding behavior between snow particles and verify it against published experimental data. Snowmelt is of interest to many civil engineers. Eggleston et al. [EIR71] simulate the accumulation and melting processes in snowpacks. The simulation was performed on a hybrid computer and calibrated and tested using field data from multiple locations.

To the end of studying snow-object interactions, applications usually prefer DEM over the grid-based approach due to its power and ease of dealing with this phenomenon. The interaction between tire treads and a snow-covered road has received particular attention for obvious safety reasons. Michael [Mic14] uses an extended DEM approach to model this interaction. A micromechanical model describes the deformation behavior of snow, the growth of the bonds between grains, and the contact behavior of snow with the surrounding boundary at a granular level. In [MVP15], this DEM-FEM (finite element method) coupling is extended to simulate the interactions between tire treads and granular terrain. A DEM model for particle-wall interaction is created for dry snow in [Lin16], wherein the snow properties used were obtained from previous studies and a field test. In finite element experiments to predict the strength of snow, the underlying snow particle structure (snow type) is often essential to model [Hag11] and has a crucial impact on the failure processes. The importance of cohesion, friction, and bonding between snow granules is highlighted by Hagenmuller et al. in [HCN15]. A discrete element approach model comprising spherical granular units is developed based on the 3D microstructure captured by microtomography. The approach can model the mechanical behavior of snow under large and rapid deformations for different snow densities.

[MEC17] use *extended distinct element method* to propose a representation method of snow splitting and sliding on a roof. It is a particle-based method that simulates connection and scission between particles. Other factors like friction, adhesion between the snow and surface, density, moisture rate of snow, etc., are considered to compute this connection and scission. [ST20] focuses on a DEM framework identifying different angles of repose values at different temperatures to study the snow/ice adhesion behavior on the sensors of autonomous cars. The physics of adhesive snow accumulation on commercial vehicles has been studied using a multiphase DEM approach in [KS21]. The framework accounts for parameters like the surface material, ambient temperature, angle of repose, wind velocity field, etc.

2.2.2. Macro-scale

Only a few works in the base field of snow mechanics study the macro-scale phenomenon of snow. In [Cor04], Corripio describes a remote sensing tool for albedo estimation using conventional terrestrial photography and its validation on an Alpine glacier. The oblique photographs are first georeferenced to a digital elevation model, followed by diffuse and direct irradiation estimation using a parametric solar model. A physical and numerical model for the blowing and drifting snow in Alpine terrain is developed and validated through field measurements by Gauer [Gau98]. Drifting snow is assumed to be a turbulent multi-phase flow consisting of a continuous and dispersed phase. Three major transport modes are incorporated in their model, namely, creeping, saltation, and suspension. Each mode is described by one or more equations, including the standard Navier-Stokes equations.

2.3. Avalanches

In addition to developing the basic snow mechanics models, exclusive attention has also been paid to understanding the basics of the processes involved in avalanches [PM76, MS06]. This is because snow avalanches pose a significant hazard to people and their resources in mountainous regions worldwide. Most of the parameters that deal with the snow are still relevant here. However, several additional factors also come into play while dealing with avalanches, for example, wind, air mass, fronts, terrain topology, temperature, precipitation, snow profile, grain structure, fracture mechanics, snow quality (plates, crystals, needles, dendrites, columns, particles, etc.). Two broad categories of avalanches are slab and powder avalanches. Slab avalanches occur when the snow is solidified, but powder avalanches occur in the accumulated non-solidified snow due to piling layers. Certain avalanches are also known to have a somewhat hybrid composition [MS06]. Additionally, avalanches can vary in their water composition, as dry or wet avalanches.

An essential requirement for most avalanches to occur is the presence of a so-called weak layer or imperfection [Sto06]. Snow failure modeling and detection constitute a bulk of mechanical engineering. Sigrist [Sig06] investigates the fracture mechanics properties of snow under tension and shear, which are responsible for the failure processes in the snow cover resulting in an avalanche. The experiments are conducted on homogenous and layered snow samples and in-situ snow beams in the field. Parameters such as fracture toughness are determined by applying a range of pressure to snow masses of different densities. Avalanches have also been studied for similarity with erosion and entrainment processes

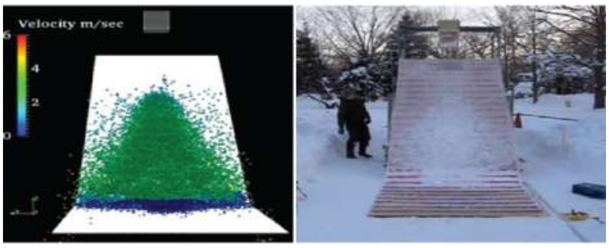


Figure 3: *Model setup and comparison between computational and experimental results of snow avalanche by [AKS14].*

in [GI04]. Computer-generated graphics are used in several applications [Atk92] for avalanche forecasting. In the community, more attention has been paid to modeling dry and wet avalanche physics mechanisms separately, as listed below.

2.3.1. Dry avalanches

It is believed that dry snow avalanches initiate from a failure in a weak snow layer below a cohesive slab. Schweizer et al. [SBJS03] focus on the formation conditions of dry snow slab avalanches, taking into account several important contributory factors like terrain, snow quality, wind, temperature, etc. Mechanical properties like snow microstructure, failure conditions, fracture propagation, etc., are also studied. Powder snow avalanches are modeled and simulated in 2D by [ERH06]. In their work, finite volume release gravity currents of large density contrast on steep slopes are simulated using a dynamic mesh adaptation technique. Reiweger uses a statistical fracture model, known as the fiber bundle model, [RSDH09] to study the failure process in the weak snow layer. The benefit of modeling snow slabs as fiber bundles is that they can accommodate a linear elastic behavior till the deformation reaches its rupture point. [DARB11] present a finite volume scheme model for simulating powder-snow avalanches in the aerosol regime. This work also studies the interaction between an avalanche and a rigid obstacle. Mass diffusion between two phases with arbitrary density ratios in a snow-powder avalanche is also handled. Steinkogler et al. [SSL14] studied the effects of snow cover properties on avalanche dynamics, such as front velocity and run-out distance.

2.3.2. Wet avalanches

Factors responsible for the formation of wet avalanches require additional treatment than dry avalanches. A primary cause of the formation of wet snow avalanches is the introduction of liquid water, leading to a decrease in snow strength. [TPS11] investigate this impact in the Eastern Swiss Alps on various snow types. Lehning et al. [LLRR08] introduce a numerical model to account for saltation, suspension, and preferential deposition processes while simulating snow transport in steep terrains. Abdelrazek and Shimizu [AKS14] simulate snow avalanches as a Bingham fluid by implementing it into SPH (Fig. 3). In the Bingham fluid model, two essential snow parameters, the cohesion and friction angle, are required to estimate the material viscosity. There have been attempts by the snow mechanics community to leverage the existing physics game engines for avalanche modeling and animation. Delparte et al. [DPPJ13] simulate avalanche flow on digital elevation model data using a game-based particle physics emulator. Many of the physical parameters (initial height, bounce friction, stickiness, damping force, etc.) for the model are determined by running genetic algorithms.

Material Point Method: Of all the methods extensively used to model or simulate avalanches in the base fields, the material point method (MPM) receives a noteworthy mention. It is a hybrid grid-based method that enables capturing a wide range of snow behaviors within the same framework. MPM is used to the end of modeling avalanches and landslides [AA10] due to a similarity in the mechanics of both. MPM is also presented as a tool to simulate avalanches and landslides in [MAMMH14], wherein the interaction with the built environment is also assessed. [GGT*18b, GGT*18c] apply MPM to reproduce the dynamics of anticracks, as well as the subsequent detachment of the slab and the flow of the avalanche. In [GvHG*19], the release and flow of snow avalanches at the slope scale using a unified model based on MPM is studied. [LSJG21] applies a 3D MPM to explore snow avalanches in different regimes on complex real-world terrain models. The idea is to account for factors such as densification and granulation, which are hard to capture in popular numerical approaches for modeling snow avalanches. Sulsky et al. [SSP*07] employed MPM to model sea-ice dynamics with a newly developed elastic-decohesion constitutive equation for modeling material failure and an algorithm to track thickness distributions and ice compactness.

2.3.3. Others

We have identified two other groups of methods that have received some attention regarding avalanche dynamics in one or more fields.

Braking effect: Some work involving snow dynamics has been dedicated to simulating and understanding the impact of forests (and other impediments) on flowing avalanches. Feistl et al. quantify the effect of forests in stopping small avalanche events [FBB*12] through several field studies. In these studies, parameters such as the starting location, deposition heights, runout distance, and forest structure were varied. The study showed that mass detrainment due to tree-avalanche interaction led to a significant deceleration of the avalanches. [FBT*14] quantify the effect of forests on small and medium avalanches. Validation is done through field studies where trees affected the runout differently. In [BVF*15], cohesion is introduced into snow avalanche dynamics calculations by incorporating additional factors like internal binding energy, Coulomb stress, and activation energy to control the release of avalanche fluidization. Fischer et al. [FKF*15] implement a depth-averaged flow model for avalanche simulation, which combines simple entrainment and friction relation. In this multivariate approach for 3D simulation, two friction parameters, namely Coulomb friction and turbulent drag, are taken into account. The developed model is validated against a documented extreme avalanche event. Rapid Mass Movements (RAAMS) [CKB10] is a popular numerical avalanche dynamics program used to simulate avalanches for such studies. RAAMS solves the depth-averaged equations governing avalanche flow and allows for the coupling of snow with objects (for example, forests, debris, etc.).

Spatial or GIS data: Some methods on avalanches perform validation over real-life spatial data or images. Corripio et al. [CDG*04] apply land-based remote sensing of snow on terrestrial images to validate their snow transport model. Their work aims to study the effects of wind on avalanche risk forecasting. In [Sto06], the stability of the mountain cover (captured using geographical information system or GIS) is investigated using a multi-dimensional, thermo-mechanical finite element model on a 2D and 3D layered snowpack. Similar to the previous works, a visco-elastic constitutive model formed the basis of this method. The model is validated through fracture experiments with snow simulation. In [DBS*14], a new physical model to model (and simulate) small and frequent avalanches using high-resolution spatial data is presented. The simulations are verified through case studies documented in the Swiss accident database. Machine learning and deep learning have recently been applied for ice flow modeling and glacier evolution on spatial data [BRG*20, JCK*22].

3. Snow in CG

Snow simulation in CG has a different focus than other disciplines discussed above. In particular, the demand for snow in CG (movies, games, etc.) arises from the need for simulation and its interaction with the objects, or realistic depiction of outdoor snow-laden landscapes in scenes with little to no interaction with snow itself, or capture the effects of snowfall in a scene or image. The main requirement across all the varied applications is that the behavior of the simulated snow or the static appearance of the snow is visibly convincing. There is a higher scope to trade off accuracy for efficiency based on the sensitivity of the human perceptual system and the application requirements. Hence, the models in CG can also concentrate on reproducing the physical effects of various types of snow (wet or dry, fresh or frozen) without directly worrying about their micrometrical properties or root causes. On the other hand, most of the desired features can be reproduced by borrowing the framework established in the snow mechanics field with appropriate modifications.

For the sake of a clear grouping and emphasis, we have classified all major published works dealing with snow animation in CG into four different sections (Fig. 4). While Sec. 3.1 on simulation below mostly corresponds to the micro-level simulations discussed in the base literature, Sec. 3.2 covers the equivalent work on avalanches. Two new groups of methods have been identified belonging more exclusively to CG. This includes techniques computing snow accumulation in outdoor scenes (Sec. 3.3), partly bearing a resemblance to the macro-scale techniques in the base literature, and those targeting to generate the effect of snowfall in the scene of images (Sec. 3.4).

3.1. Simulation

Even with the aforestated liberty while reproducing its visual behavior, snow simulation is still challenging in CG. As stated earlier, snow comes in several forms, which are formed as a result of different temperatures and other conditions. It has transforming properties, behaving as a rigid or deforming mass and even possessing fluid-like properties. The compressive nature of snow is unlike that of other substances such as sand or water and hence necessitates additional sophisticated handling. Sintering or bond formation over time is another feature unlike other commonly animated

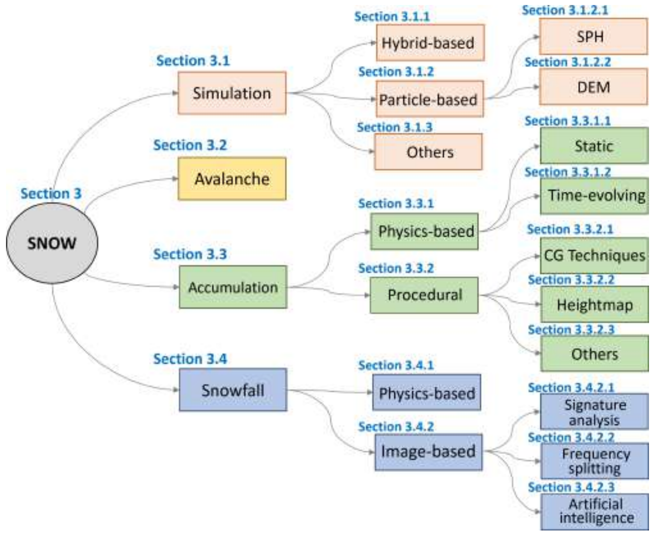


Figure 4: *Organization of the sections on snow in this report, based on the published works in CG.*

materials in CG and has to be accounted for directly or indirectly. Other aspects include boundary handling with solid objects, topological changes in complex scenes, etc. The main impediment lies in achieving all the targeted behaviors while keeping a low cost per simulation step.

Various methods with very different underlying approaches have been proposed to simulate snow in the last few years. This includes hybrid-based, particle-based, and even Lattice Boltzmann methods. Centric to all the models is their ability to capture the crucial visual properties of density, deformation, interaction with other objects, and, in some cases, phase changes of snow. Furthermore, almost all of these methods model the micro-level physical behavior of snow at a chosen granularity level to capture these effects. Most of the techniques for snow simulation in CG are inspired by those in basic science. As a result of this inheritance, most methods below make use of the fundamental physical parameters, as well as some user-defined variables or handles employed in the underlying models. In the following, we will introduce the basic mathematical details of some relevant simulation techniques at a higher level.

3.1.1. Hybrid-based

Two main methods in CG deal with snow simulation with the hybrid Eulerian/Lagrangian framework. The first one is Stomakhin et al. [SSC*13], who propose a grid-based semi-implicit snow simulation method using MPM, also see [JST*16] for further extension. Inspired by its base method, the method is continuum-based. It utilizes a user-controllable elastoplastic constitutive model (handling volume conservation, plasticity, stiffness, fracture), which enables simulation of wet/dry snow, treatment of self-collision, fracture, and even capturing of sticky effect (Fig. 5a). The reliance on the continuum model eliminates the need to model snow discretization

(a) (b) (c) (d)

Figure 5: *Snow simulation methods with their grid resolution/particles used to generate the image (a) packing snow effect captured by the MPM by Stomakhin et al. [SSC*13] (200 × 240 × 470 / 7.2M), (b) hybrid particles and grid-based friction and cohesion by [TFN14] (50 × 50 × 50 / 33K), (c) elastic deformation produced by the implicit compressible SPH solver of [GHB*20] (−/2.57M), (d) interactive/real-time snow simulation using iterative DEM solver on the GPU by Goswami et al. [GNN22] (−/96K).*

as single snow particles or grains. The fundamental equations used in [SSC*13] are

$$\frac{D\rho}{Dt} = 0, \quad \rho\frac{D\boldsymbol{v}}{Dt} = \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} + \rho\boldsymbol{g}, \quad \boldsymbol{\sigma} = \frac{1}{J}\frac{\partial\Psi}{\partial\boldsymbol{F}_E}\boldsymbol{F}_E^T \tag{3}$$

where ρ is the material density, $\boldsymbol{v}$ is the velocity, $\boldsymbol{g}$ is the gravitational constant, $\boldsymbol{\sigma}$ is the Cauchy stress (defining the state of stress at a point inside a body continuum in the deformed configuration), and Ψ is the elasto-plastic potential energy density, $\boldsymbol{F}$ is the deformation gradient defined as $\boldsymbol{F} = \frac{\partial\phi}{\partial\boldsymbol{X}}$ for the deformation ϕ, $\boldsymbol{F}_E$ is the elastic part of $\boldsymbol{F}$, and $J = det(\boldsymbol{F}) \cdot \frac{D}{Dt}$ is the material derivative, that can be expanded to $\frac{D}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{v}\cdot\nabla$ (see [JST*16] for more details). The term $\boldsymbol{\sigma}$ is essential to capture the cohesion and frictional effects of the material.

The plastic behavior of snow is obtained by using the concept of principal stretches instead of principal stresses to define a new plastic yield criteria, as well as a simplification of the hardening behavior. Ψ is formulated as a function of $\boldsymbol{F}_E$ and $\boldsymbol{F}_P$. Interestingly, Lamé parameters, which appear in the formulation on Ψ, are modified in the hyperelastic energy density (for the principal stretching), with both of them being a function of the plastic deformation gradients. The look and dynamics of the simulated snow are controlled by a bunch of other parameters that appear in the formulation of Ψ, including critical compression θ_C, critical stretch θ_S, hardening coefficient η, initial density E_0, Poisson ratio μ. The user can tune these properties to obtain a range of various combinations of snow behavior, such as chunky or powdery, hard or soft, brittle or ductile, etc. At the beginning of the animation loop, particle data is rasterized to the grid, where the grid forces and velocities are solved. The updated grid parameters are used to calculate the deformation gradient and particle velocity for all the particles.

The advantage of MPM over purely Eulerian methods is that the former uses particles to track mass, momentum, and deformation gradient ($\frac{D\rho}{Dt}$ and $\rho\frac{D\boldsymbol{v}}{Dt}$). To this end, each particle needs to store $\boldsymbol{x}$, $\boldsymbol{v}$ and $\boldsymbol{F}$. On the other hand, the spatial derivatives of the stress tensor ($\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}$) are harder to calculate on the Lagrangian particles and are hence calculated on the Eulerian grid. The presented implementation is CPU-based and, hence, offline and ignores the interaction of snow with the surrounding air.

Since the introduction of [SSC*13], MPM has become popular for simulating snow in CG. Fang et al. [FHHJ18] have presented a temporally adaptive parallel MPM with regional time stepping targetting various materials that the method can handle, including snow. A realistic simulation using MPM to assess the interaction between snow and rubber tires is employed in [Sch16]. Additionally, different shapes of ice crystals are modeled and placed on the surface to form the appearance of snow crystals. A massively parallel scalable framework of MPM on a multi-GPU platform to simulate the physical behaviors of several materials is proposed by Wang et al. in [WQS*20]. This includes a new particle data structure to promote coalesced memory access patterns on the GPU, a kernel fusion approach using a grid-to-particles-to-grid scheme to reduce GPU kernel launches, and optimized algorithmic designs for sparse grids in a shared memory context. Hu et al. [HLA*19] proposed a data-oriented programming language for efficiently accessing and maintaining spatially sparse data structures. To this end, they have shown the power of their high-performing data structure for MPM on CPU and GPU. Although [WQS*20] have not explicitly demonstrated the case of snow in their experiments, we expect that their framework would still work for snow since it enables generic multi-platform scalability for physical behaviors of various materials simulated using MPM.

The second method is by Takahashi et al. [TFN14], who simulate the dynamics of snow employing the *fluid-implicit-particle* (FLIP) method [ZB05], also taking the continuum approach and the hybrid grid-based framework, see also [AO11] for similarities in the force model. However, their approach's basic mathematical formulation is derived with the help of more traditional Navier-Stokes equations suited for granular behavior (Eq. 4).

$$\frac{D\rho}{Dt} = 0, \quad \rho\frac{D\boldsymbol{v}}{Dt} = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot\boldsymbol{s} + \boldsymbol{f}_{ext}, \quad \|\boldsymbol{s}\|_F \leq \sqrt{3}\alpha p + c \tag{4}$$

Here p is the pressure, $\boldsymbol{s}$ the deviatoric or frictional stress, $\boldsymbol{f}_{ext}$ the external forces, $\|\boldsymbol{s}\|_F$ the Frobenious norm of $\boldsymbol{s}$, α a function of the angle of repose, and c is a cohesion criterion. The frictional stress tensor replaces the Cauchy stress tensor in the MPM formulation, while the pressure also appears in the equation here. The compressible behavior of snow due to interaction with solid or otherwise is accounted for by approximating snow mass as a collection of porous snow particles (Fig. 5b). To account for this, in addition to the compressibility caused by collision with objects, a variable termed as *durability* for each particle is assigned, representing the level of compression already achieved starting from a porous state. In each simulation step, particle attributes (velocity and durability) are transferred to the grid, followed by the FLIP solver updating pressure, friction, and cohesion, finally leading to the interpolation of grid velocity and durability from the grid onto the particles. The yield condition on a pile of snow is determined by comparing $\boldsymbol{s}$ against cohesion coefficients c to check if the snow is in the plastic composition.

3.1.2. Particle-based

Like the base literature, Lagrangian methods to simulate snow in CG adopt one of the following two approaches. Even though both SPH and DEM use a particle-based approach to simulate snow, their underlying physical approaches differ significantly, as discussed below.

3.1.2.1. SPH

Few methods existed that employed SPH or another purely particle-based approach to the end of snow simulation in CG for a long time. Recently, an implicit compressible SPH solver for snow simulation has been proposed by Gissler et al. [GHB*20] (Fig. 5c). It is inspired by the base method implicit incompressible SPH (IISPH) [ICS*14], an SPH-based approach to simulate incompressible fluid flow. An advantage of the base implicit solver of IISPH (first equation in Eq. 5) is that it allows large time steps in the context of a particle-based solver. The method models snow as an elastoplastic continuous material that can capture both small and large snow volumes, including snowfall and accumulation on surfaces. However, against the base method IISPH, which is designed to handle incompressible fluids efficiently through an implicit solver, the snow solver required some modifications to handle the compressible properties of snow. The formulation of IISPH is obtained by discretizing the basic continuity equation using a forward difference at Δt of the density (first equation in Eq. 5) and solving the implicit equation iteratively for all the particles. Each snow particle is assumed to have an elastic component that resists deformation, beyond which the deformation is plastic. The acceleration of snow due to elastic deformation is accounted for by using two separate solvers, each corresponding to acceleration caused due to one Lamé parameter used, λ and G. To this end, the first solver iteratively computes the pressure term (second equation in Eq. 5), which in turn computes the acceleration $\boldsymbol{a}^{\lambda}$ countering the compression of snow. In essence, this is achieved by solving an equation of state, wherein $\lambda^{t+\Delta t}$ at time $t+\Delta t$ is set equal to λ^{t} and $\rho_0^{t+\Delta t}$ to ρ_0^{t} for the elastic computation (λ physically representing material stiffness). The shear accelerations inside snow volume are accounted for using the other Lamé parameter G and strain tensor $\boldsymbol{\epsilon}$. The Cauchy stress is used to refine the particle velocity computed so far, considering the acceleration due to all other factors to incorporate the shear acceleration in the second solver.

$$\begin{aligned}
&\frac{\rho_i(t+\Delta t)-\rho_i(t)}{\Delta t}=\sum_j m_j \boldsymbol{v}_{ij}(t+\Delta t)\nabla W_{ij},\\
&p^{(t+\Delta t)}=\lambda^{t+\Delta t}\left(\frac{\rho^{t+\Delta t}}{\rho_0^{t+\Delta t}}-1\right),\\
&\boldsymbol{a}^{\lambda}=-\frac{1}{\rho}\nabla p, \quad \boldsymbol{\sigma}^{(t+\Delta t)}=2G^t\boldsymbol{\epsilon}^{(t+\Delta t)}
\end{aligned} \tag{5}$$

In [GHB*20], each snow particle is additionally subjected to plastic deformation to account for the hardening of snow. The basic idea is to update the Lamé parameters λ and G in each simulation loop to reflect this change. To put it simply, this is achieved by computing elastic deformation gradient $\boldsymbol{F}_E$ for each snow particle using updated velocity and allowing a maximum elastic deformation, marking all larger deformations as permanent (adopted from [SSC*13]). To this end, [GHB*20] adopt some user-tuneable parameters in their solver to obtain varied behaviors of snow. In addition to other snow properties like deformation, breaking, compression, hardening, etc., two-way coupling with rigid bodies, boundary handling, and phase change are also supported. For boundary computations, all solid objects are sampled with boundary particles and involved in the solvers/computations wherever required [PGBT18].

Takahashi and Fujishiro [TF12] strive to simulate interactions between snow and objects, taking the sintering effect into account, assuming a simple thermal conduction model. Rigid bodies are sampled into particles, and SPH is used to approximate the behavior of snow as a fluid by modifying the viscosity term in the Navier-Stokes equations. At the same time, the adhesion of snow particles to rigid bodies is managed within the same framework by appending the sintering term to the Navier-Stokes equations. Wang et al. [WWX*23] investigate the driving behavior of tires on snow-covered terrain. The SPH and FEM are employed to model the behaviors of the snow layers and the pavement, respectively.

3.1.2.2. DEM

DEM is another particle-based technique to predict bulk solids behavior, more popular in structural or solid mechanics. It was only recently that some DEM-based methods to simulate snow animation have been developed in CG. This development is inspired by the fact that DEM continues to be a popular and simple method to simulate snow/ice animation in the base fields, especially when dealing with snow-object interaction. Inspired from [HCN15], Goswami et al. [GMH19] have proposed a particle-based DEM simulation on the GPU, which computes density and other variables without the need for SPH kernel-based interpolation in the simulation. The hybrid and SPH-based methods bear a small or large resemblance in terms of the underlying model (Navier-Stokes equations) and the conceptual parameters employed. However, the DEM method uses an altogether different approach.

In [GMH19], a particle starts as pure snow r_s with air entrapped in it and is allowed to transform into ice r_i as it shrinks in its radius gradually due to compression. Therefore, the density, and hence physical state η of any particle, can be determined based on its radius. η is used to interpolate the physical quantities (Young's modulus, etc.) of any particle based on its state between that of snow and ice. Any particle exerts a normal force $\boldsymbol{f}_n$ (along direction vector $\boldsymbol{n}$) and tangential force $\boldsymbol{f}_t$ (see Eq. 6) on its neighbors. Here E is Young's modulus of the particle, r_d its radius, ζ_n is the cohesive strength in the normal direction, δ the overlap between particles, $\boldsymbol{u}_t$ is the accumulated shear displacement, and φ is the angle of repose. $\boldsymbol{f}_n$ between any two neighboring particles is attractive when they do not overlap ($\delta \leq 0$) and repulsive otherwise ($\delta \leq 0$). It is worth mentioning that the neighborhood set of a particle is much reduced in DEM than in SPH. Furthermore, [GMH19] takes the average of normal and tangential forces when considering a pair of particles to follow Newton's third law of equal and opposite forces between particles.

$$\boldsymbol{f}_n = \begin{cases} -(Er_d\delta)\boldsymbol{n} & \text{if } -Er_d\delta < 4\zeta_n r_d^2 \\ 0 \text{ and cohesion is broken} & \text{if } -Er_d\delta \geq 4\zeta_n r_d^2 \end{cases} \tag{6}$$
$$\boldsymbol{f}_t = (\boldsymbol{u}_t/|\boldsymbol{u}_t|)|\boldsymbol{f}_n| tan(\varphi)$$

Unlike the hybrid and SPH-based methods, DEM methods need to explicitly account for the bonds between snow particles to reproduce the corresponding behavior. Hybrid and SPH methods in CG control this phenomenon indirectly using the additional user-defined parameters. As seen in Eq. 6, the normal cohesive forces between neighboring particles are attractive for as long as the bonds between them are not broken. This is dealt with in [GMH19] by marking particles with broken bonds (instead of tracking particles with created bonds) when their number of neighbors is reduced drastically within a short time, primarily due to some impact. Such particles are excluded from exerting attractive normal forces on their neighbors, which have also been marked with broken bonds. Each snow particle also undergoes a compression in its radius based on the impinging forces by adopting the durability principle in [TFN14]. At the end of each simulation loop, all the forces acting on each particle update its velocity and position for the next frame. The method is efficient, operates at high frame rates, and also incorporates lightweight bonding approximation between the particles, as well as the effect of thermodynamics. Despite a high frame rate, a limiting factor in their case is the low time step needed to maintain the stability (0.1 ms).

[GNN22] extended and improved their original method in [GMH19] by employing a GPU-based iterative discrete element solver on the base method (Fig. 5d). This iterative method incorporates time steps that are ten times larger and can also simulate various behaviors, such as dry and wet snow, and even tending to ice within the same framework by merely varying the physical properties of simulated particles. In addition to the basic physical parameters, which change with the radius of particles, the bonding threshold is also adjusted to reproduce varied compositions of snow. The interaction of snow with solid objects is handled using boundary particles, and a user study is conducted to validate the realism value of their approach for applications such as computer games. This method operates at high frame rates while applying time steps in agreement with [GHB*20], which uses a larger smoothing kernel over a more extensive neighborhood set of particles. It is also important to mention that both these DEM methods, [GMH19] and [GNN22], are targeted to provide a more efficient than accurate physical snow simulator. To this end, the simulation relies on empirical values for some physical parameters (for instance, E and σ_n), which are much smaller than their actual values. This was essential to keep a large time step for applications like games. Also, the motivation for a user study originated from these approximations and the need to investigate if the users found the simulation realistic enough with the modified parameters in the solver. Based on the user study results reported in [GNN22], the participants found their simulation realistic for games and other similar real-time applications.

3.1.3. Other methods

Some methods to animate snow employ simplified physics only partially in the overall pipeline to achieve a limited task. [WF15] model snow as a hybrid structure that handles movable snow and static snow separately (Fig. 6a). Whereas movable snow is represented as a set of particles connected using springs, static snow is modeled as grid cells, staying more like passive units. The method handles the conversion between particles and grid cells to account for changing activity. The brittle property of the snow is captured by manipulating strings that connect movable particles.

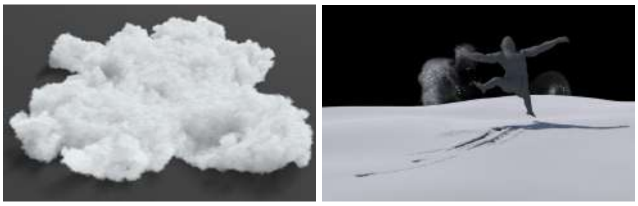
In some cases, the impressions generated by trampling on snow can also be considered a pure surface phenomenon. Dagenais et al. [DGP16] decompose snow accumulation into three layers to generate snow imprints on interaction with dynamic objects (Fig. 6b). The base layer is the non-interacting layer influencing the deformation due to object imprint and is stored as a level set. The second layer comprising snow particles is added to model the interaction between the snow and the dynamic objects and is animated using adapted granular simulation. A thin layer composed of powdery snow is also added during the object-snow interaction to model the airborne snow mist, which is animated using fluid simulation. The framework controls the transfer of snow from the base layer to the snow particles and then to the snow mist. Noise is added to the base layer to make the interaction marks look more realistic, and factors such as air resistance, friction, etc., are also accounted for. Even though this method uses a lean component of physical simulation, it is largely governed by procedural mechanisms.

Sumner et al. [SOH99] develop a simulation model composed of a uniform grid forming a height field. This height field defines the ground surface, which solid objects can deform. The model can simulate the approximate behavior of multiple materials like sand, mud, and snow and supports parallel CPU-based processing. A particle system enables modeling portions of the simulated material thrown into the air and adhesiveness between the object and the material.

Discussion: The discussed physics-based snow simulation methods in CG follow very different approaches to capture the visible behavior of snow, and hence, they have their different advantages, also see Tab. 1. Apart from other forces, a key factor that governs snow dynamics is the prevalent stress force. The discussed models differ in their treatment to capture this stress force, especially the discussed DEM technique, which has a completely different approach to account for the stress force. Both MPM [SSC*13] and

Method	*Snow/Ice*	*Type*	$\Delta t(ms)$	*Resolution*	*Time/Step(s)*	*Properties*
Stomakhin et al. [SSC*13]	snow	hybrid	0.5 0.5	700×120×210(0.58M) 200×240×470(7.2M)	180 2142	MPM, elasto-plastic constitutive model (CPU)
Takahashi et al. [TFN14]	snow	hybrid	1.0 1.0	40×40×40(128K) 60×60×60(1.9M)	0.8 3.3	FLIP, durability (CPU)
Gissler et al. [GHB*20]	snow	particles	0.1 3.1	5.04M(5 mm) 5.07M(8 cm)	522 194	IISPH, small and large volumes of snow (CPU)
Goswami et al. [GNN22]	snow	particles	0.6 1.0	0.5M(1.26 cm) 1.0M(1 cm)	0.036 0.104	DEM, iterative, perceptual user study (GPU)
Liu [LCZ*21]	snow avalanche	particles	1.0 1.0	2.7M 90K	26 0.11	Position-based dynamics (GPU)
Iwasaki et al. [IUDN10]	ice	particles	3.0 3.0	97K(1.1 mm) 200K (1.1 mm)	0.017 0.084	SPH, melting, photon emission (GPU)
Lii & Wong [LW14]	ice	particles	0.01 0.01	32×32×32(13.3K) 64×64×64(53.4K)	0.066 0.685	Virtual water particles, density field (GPU)
Miao & Xiao [MX15]	ice	particles	16	35K 100K	0.2 0.5	PBD, freezing, air bubbles (CPU)
Ishikawa et al. [IYW*15]	glazed frost	hybrid	1.0 1.0	128×128×128(20K)	0.02	FLIP, freezing, heat transfer (GPU)

Table 1: *Comparison of various physics-based snow and ice simulation methods in CG. For hybrid methods, both the maximum grid and particle resolution are provided. For the particle-based approaches, particle sizes used in the simulation are stated (if provided in the publications).*



(a) **(b)**

Figure 6: *(a) Snow spread generated using hybrid particles and grid-based snow simulation by Wong and Fu et al. [WF15], (b) snow imprints created by a three-layered approach by Dagenais et al. [DGP16].*

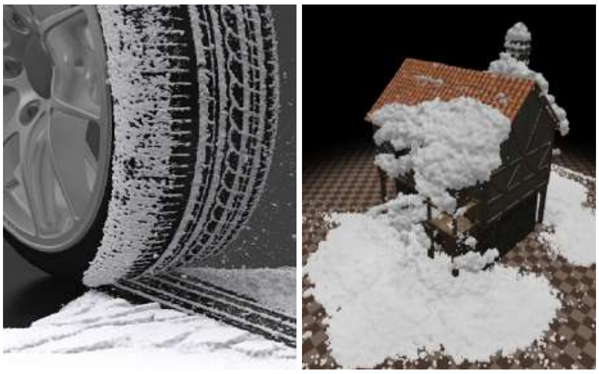
implicit compressible SPH [GHB*20] methods for snow have some similarities in their fundamental underlying model; they also follow a more rigorous physical approach and demonstrate their methods for reasonably large scenes. Stomakhin et al. have demonstrated that the improved MPM approach can not only incorporate several features like volume preservation, stiffness, plasticity, and fracture together within the same framework, but it also allows time steps about 100 times larger than the explicit update scheme. On the other hand, Gissler et al. not only extend the IISPH framework to incorporate the compressible characteristics of snow with their proposed approach but also inherit the advantages of particle-based methods (including handling different snow volumes, snowfall, boundary handling with other objects, etc.).

In addition to what these methods can achieve visibly, a desirable feature is their ability to operate with large time steps. However, even with large time steps these methods can support, the per-step simulation time is relatively high to allow anything close to a real-time behavior with a fine resolution. Since the methods were developed on different hardware spanning over a period gap of about seven years and might have used different levels of parallel computations (or no parallelism at all), an exact comparison between the reported time cost per simulation time would be hard to establish. Efficient GPU implementations are generally known to accelerate computations significantly compared to the CPU equivalents [GSSP10, GEF15]. On the other hand, the existing DEM methods [GMH19, GNN22] have allowed approximations to alter some physical parameters, making them somewhat less accurate. However, the DEM methods can operate at similarly large time steps and at a much reduced per-step simulation overhead, making them more viable for applications demanding efficiency over accuracy. An advantage of both particle-based methods, SPH or DEM, is their capability to handle the interaction between snow and objects more detailedly to generate small-scale effects. Both these methods also account for the drag force caused by air on the snow particles, which becomes prominent in the case of snow due to its lower density. Fig. 7 shows the interaction between snow-object particles generated using implicit compressible SPH and iterative DEM solvers, with both methods employing boundary particles and modified physics for boundary handling.

3.2. Avalanche

Similar to basic snow mechanics, some specialized solutions inheriting the fundamental snow simulation framework for large scales have been developed to convey the visual feel of avalanches in CG. Two major physical models to capture avalanche dynamics with different focuses exist to this end.

Tsuda et al. [TYDN10] simulate a mixed-motion avalanche consisting of snow smoke and liquified snow. Such avalanches travel down the slope at high speed and often create interesting visual effects. The interacting flow is divided into three layers: accumulated snow (deepest layer), dense-flow layer (liquified or moving snow),



(a) **(b)**

Figure 7: *Particle-based physics-based snow simulation methods can handle a more detailed interaction with objects (a) implicit compressible SPH-based snow solver by [GHB*20] (5.5M particles), (b) DEM-based iterative snow solver by [GNN22] (1M particles).*

(a) **(b)**

Figure 8: *Avalanche simulation methods (a) physically-based avalanche simulation by Lui et al. [LCZ*21], (b) avalanche generated using Houdini particles in [Kap03].*

and suspended snow mass (snow smoke). Of these, the snow smoke is simulated by a standard grid-based approach, while the other two layers are simulated using SPH. To model the avalanche dynamics, two types of interactions are accounted for: between the suspension and dense-flow layers and between the dense-flow and accumulated snow layer. Additional factors like drag, buoyancy, and vorticity forces are employed while dealing with the suspension layer of snow. This also allows for the mass exchange between the suspension and dense-flow layer. The interaction between dense-flow and accumulated snow layers is treated as probabilistic mass entrainment using strain rate, density, slope angle, etc.

Liu et al. [LCZ*21] propose a particle-based algorithm (Eq. 7) to simulate dynamic avalanches under the position-based dynamics (PBD) framework [BKCW14] (Fig. 8a) on the GPU. The PBD method iteratively predicts particle positions for the next physics step such that the framed constraints on those positions are mutually maintained. By treating snow as a viscoplastic material, the PBD formulation is translated to reducing the potential energy function for particles by framing the elastic and plastic constraints. To this end, the potential energy is obtained from the energy density formulation similar to MPM by Stomakhin et al. [SSC*13], with the same deformation gradient $\boldsymbol{F} = \boldsymbol{F}_E\,\boldsymbol{F}_P$. The elastic/plastic range and yield criteria are computed by following the *Bingham* model, wherein the material exhibits solid properties for stress less than the yield stress (shear stress in this case) and plastic flow otherwise. The stress-strain tensor formulation on the elastic component is given in Eq. 7. Here, the additional parameters are the $\boldsymbol{C}$ which is a PBD constraint (framed on reducing potential energy), and $\dot{\gamma}$ the shear strain computed from the velocity tensor field (subscript e represents the elastic component).

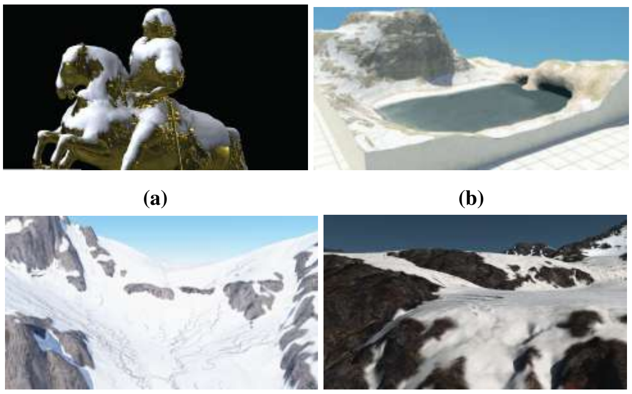
$$\epsilon_e = \frac{1}{2}(\boldsymbol{F}_E^T\boldsymbol{F}_E - \boldsymbol{I}), \quad \boldsymbol{\sigma} = \boldsymbol{C}\epsilon_e, \quad \dot{\gamma} = \frac{1}{2}(\nabla\boldsymbol{v} + \nabla(\boldsymbol{v})^T) \tag{7}$$

In the modified Bingham model, the computed variables are coupled with other theories to determine the critical shear stress, which in turn determines the yielding point between elasticity and plasticity. A two-way fluid-solid coupling model based on a parallel level-set approach handles the interaction between snow flow in the avalanche and the surrounding objects. The method can handle wet as well as dry avalanche cases. The method is GPU-based, and drag force is used for the air resistance. In some sense, Liu et al. [SSC*13] have proposed the PBD variation of MPM. It is interesting to note that neither of these works, however, studies the triggering factors or conditions that initiate avalanches. This is in contrast to the base disciplines wherein several works target to determine the weak layer or triggering conditions.

Some other works in CG also target simplified variants of avalanche animation. A simple non-Newtonian fluid model based on a combination of SPH and rheological models simulates the snow avalanche flow in [Kro10]. The method achieves nearly 215 iterations per second for 64K particles on an NVIDIA Fermi GPU. The technique can, however, handle limited cases. In [DPJP12], a simple model to simulate avalanche flow for gaming is created with the help of OpenGL particles and a digital elevation model for terrains. Using a genetic algorithm approach on a training dataset composed of 30 known avalanche paths, avalanche path outlines are generated on a provided terrain and starting zone. A smart particle system interacting with voxels in Houdini is designed in [Kap03] for avalanche animation (Fig. 8b). The particles were obtained from Houdini points, reducing the computational overhead with their ability to cross-communicate with the voxels.

Discussion: Avalanche simulation methods bear a strong resemblance to conventional snow simulation methods in CG when considering the adopted physics framework. This could potentially imply that the physics of many micro-level snow simulation methods could be extended to incorporate avalanche dynamics and vice-versa. The time steps and the computational time per simulation step are in agreement with the corresponding state-of-the-art snow simulation methods as well. In addition to the physical parameters previously stated, one aspect that clearly distinguishes in the case of avalanches is the scale of simulation involved, which is usually much larger even when compared with a typical outdoor area containing simulated snow. The following part of this report discusses another CG application wherein large-scale outdoor scenes are considered for computing snow accumulation. These methods, however, differ considerably in their requirements and, hence, underlying methods with the simulation techniques discussed so far.

Figure 9: *Comparison of the snow accumulation methods presented in [HM09]. Wind-driven accumulated snow modeled in left: [FO02], middle: [MMAL05], right: [HM09].*



(a) **(b)** **(c)** **(d)**

Figure 10: *Physics-based snow accumulation models (a) diffusion-based cover by [FG11], (b) heat transfer simulation for modeling realistic winter scenes in [MGG* 10], (c) time-evolving snow-covered landscape and ski tracks generated by Cordonnier et al. [CEG* 18], (d) physically-based glacier evolution and rendering by Argudo et al. [AGP* 20].*

3.3. Accumulation

Computing realistic accumulation of freshly fallen snow on objects in outdoor environments has been of interest in CG to generate viable backgrounds in snow-laden settings. Contrary to the simulation methods for snow animation, simplified techniques have been developed in CG that focus more on portraying realistic static snow accumulation over larger terrains and other outdoor scenes. As discussed below, the accumulation problem can be modeled over terrains represented using a simpler micro-physics than simulation or entirely procedurally through the use of heightmap, mesh-based layer, etc. The elimination or suitable modification of the micro-level physics in these applications is essential to avoid long computational times, given that the scale of terrain and, hence, the involved computations are often quite extensive. Additionally, many physics-based snow accumulation methods accommodate various environmental factors to provide the snow cover transformation or evolution as a function of time. All the snow accumulation techniques in CG can be categorized into two major types, physics-based and procedural, as outlined below.

3.3.1. Physics-based

An interesting observation with the snow accumulation methods in CG is that these methods can vary from using much physics to capture several physical phenomena to very little physics to capture just a few. These physics-based accumulation methods can be classified based on a few different categories. We have loosely divided them into *static* and *time-evolving* accumulations below for brevity. By time-evolving, we refer to accumulation methods that target the transformation of snow over time, for instance, due to melting or freezing. It is worth mentioning that several methods outlined as static do treat mass evolution as a function of wind or account for snow accumulation stability on surfaces based on factors like temperature. However, they do not focus on the explicit evolution of the landscape due to phase change, which requires the inclusion of several environmental factors together with the concept of time.

3.3.1.1. Static

Several static methods attempt to obtain a realistic snow accumulation cover on limited landscapes containing obstacles, such as buildings, trees, etc. The common purpose of employing physics here is to model the behavior of snow accumulation with blowing wind or to account for stability when interacting with objects in the scene. A simple 2D cellular automata-based parallel model to simulate snow transport by wind is proposed in [MC95]. The method comprises two major steps, namely, collision and propagation, to govern the movement of snow particles. The interaction between snow particles is captured using a cohesion force that attracts any falling particle to an existing deposited snow mass. However, the presented interaction forces are too simplistic to allow any complex snow behavior.

Snow accumulation and drifts created due to the wind-blown snow around obstacles are proposed in Feldman and O'Brien [FO02]. A three-dimensional flow of air in the volume around the obstacles is computed using the Navier-Stokes equations to model the convection, deposition, and lift of snow by the wind field. A similar problem is targetted by Moeslund et al. [MMAL05] wherein the falling and accumulated snow is modeled using simplified physics. It is one of the few works in CG that models the size and density of snowflakes as a function of the temperature. Snowflakes are modeled by combining triangular polygons with the help of concentric spheres, with several layers governing the size of the snowflake. The movement of each snowflake is dictated by gravity, buoyancy, lift, and drag forces, which are similar in nature to those employed in avalanche dynamics. In order to model lift and drag forces on the falling and accumulating snow realistically, a wind field is generated by the Navier-Stokes equations. The falling snowflakes are collected on the ground, and obstacles are represented as adaptive triangle meshes wherein areas receiving a higher accumulation are refined further. The stability of deposited snow on various triangles is determined using the angle of repose, which also considers the temperature.

[HM09] introduce the concept of *snow packages* (discrete volumes of snow) for modeling progressive snow accumulation on static surfaces. In contrast to [FO02, MMAL05], Hinks and Museth handle snow accumulation on topologically complex shapes with the help of dual-level set structures. Furthermore, a local particle-level set is used to deposit stable amounts of snow on surfaces. A wind field is established and solved within the domain cells with the help of Navier-Stokes equations. The stability of snow build-up is determined by crucial factors such as physically-based stability criterion and global temperature value. The three accumulation methods are visually compared in [HM09], see also Fig. 9.

A different approach for generating snow cover is proposed in [FG11]. The mathematical framework is based on the diffusion process for computing snow distribution. It considers the physical properties of falling and fallen snow, as well as adhesive particle impingement (Fig. 10a). Features like snow bridges and snow overhangs, in addition to textures, are used to convey the appearance of accumulation on scene geometries.

3.3.1.2. Time-evolving

Methods in CG attempting to provide a time-varying accumulation of snow often target larger landscapes and timespans (for instance, seasonal changes in snow cover over terrain or mountains) than the *static* category. Furthermore, the idea is to compute the visual look of a given terrain during a time of the year and its transformation approximately without getting down to the micro-level of physics. Therefore, more often than not, small artificial structures do not appear in the overall problem description. A common feature in these methods is the use of some environmental profile, often not required in the *static* accumulation classification, to maintain and modify the environmental variables affecting the overall process with time.

Maréchal et al. [MGG*10] simulate the evolution of winter landscapes by considering the thermal or heat transfer between scene elements. Weather parameters such as air and dew point temperatures, precipitations, etc., are contained as a part of an environmental profile, which is allowed to evolve based on the weather model (Fig. 10b). The scene is represented with a voxel-based schema wherein each voxel could contain a single material (snow, water, ice, air, rock) or more than one material in some cases (snow-air, ice-water). Based on the current environmental profile, the simulation loop computes snowfall and its accumulation on the ground or scene voxels. This is followed by computing the heat flows of different types through the faces of the voxels, leading to the determination of new voxel temperatures and possible phase change to ice or water. The heat exchange between voxels simplifies the overall thermal transfer process, constituting conduction, convection, and radiation. The technique ignores the effect caused by evaporation or wind change in the whole process.

A GPU-based interactive approach to generate snow-covered landscapes together with user control to evolve them dynamically over time is presented in [CEG*18] (Fig. 10c). The terrain is modeled as a 2D height grid (the static ground height), stacked by other materials, such as compacted, stable, unstable, and powdery snow, in that order. Various phenomena are handled by manipulating these layers. For instance, snowfall modifies powdery, unstable, and stable layers, while snow diffusion happens to the powdery layer. Events such as snow accumulation, avalanches, skiing tracks, etc., are modeled in the framework with the help of user control and environmental variables. The events are classified and modeled separately as long-term (for instance, snow cover accumulation) or short-term Poisson (for instance, avalanches, ski tracks) stochastic events. The temperature at all grid cells is evaluated based on the direct sunlight impinging on the point and the ambient or indirect exposure component. Similarly, the wind velocity field is generated as a 2D field and enhanced, considering other parameters affecting wind flow, such as the terrain slope. With its supported events, the framework aims to model snow features at interactive frame rates across terrain scales of 10×10 km while validating the method with a lightweight user study.

Argudo et al. [AGP*20] simulate the temporal evolution of glaciers by a hybrid method that combines physics and proceduralism (Fig. 10d). Similar to [CEG*18], a layered heightfield model in 2D with static bedrock and time-varying ice layers is adopted. The evolution of ice coverage is solved in the established 2D grid with a variation of Navier-Stokes equations and the environmental profile by modeling the ice movement as a diffusion process between neighboring cells. The diffusion computation is accelerated by a multiresolution approach that uses a coarser resolution for smooth areas and selective upsampling for other areas. In the next step following the diffusion computation, small-scale 3D procedural landforms like crevasses, moraines, icefalls, etc., are generated based on the simulation data. To this end, procedural placement rules or stochastic distribution, considering the basic terrain, are leveraged to augment the terrain with small-scale land features at a slower timeline than the glacial evolution. The overall framework reports running their tests over mountainous areas spanning up to 84 km, with a maximum height of 557 m, over a time span of up to nearly 3200 years, and with a coarse/fine grid resolution of size 250/2800.

Discussion: By this point, we have covered all the major approaches in CG targetting accumulated snow mass at different scales by employing different levels of physics. The scale of the problem and the involvement of the time dynamics clearly have a substantial impact on the nature of physics employed in different problems. The Navier-Stokes equations applied in the case of vast outdoor landscapes to model snow's movement or phase change appear in a different formulation than when simulating a limited mass of snow for a more rigorous physical behavior. Interaction with objects can often be simplified or wholly ignored at large scales; the time evolution factor, on the other hand, becomes more prominent to capture.

3.3.2. Procedural

The impression of snow accumulation in outdoor settings can be conveyed through purely procedural methods with no or very lean physics component, especially if the application does not require object-snow interaction or to capture the transformative properties of snow. In contrast to techniques that employ various levels of physics, these approaches focus on producing a plausible or convincing appearance of accumulation over surfaces by altogether skipping any physics whatsoever. In this section, we cover the major procedural techniques adopted in CG to achieve snow accumulation in various scenes.

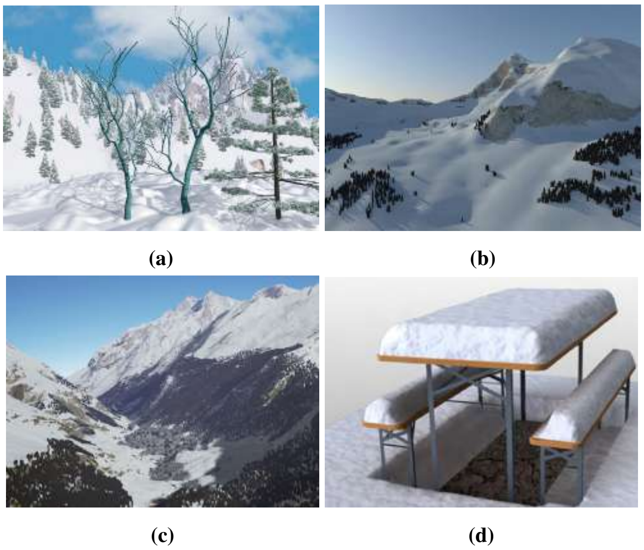


Figure 11: *Procedural snow accumulation and deposition by different methods (a) hybrid multi-mapping technique for modeling snow scenery in [YSHW03], (b) seasonal snow cover generated on the alpine terrain in [PTS99], (c) seasonal snow cover in Zermatt region approximated in Neukom et al. [NAS18], (d) statistical snow deposition model by [FG09].*

3.3.2.1. Standard CG techniques

Standard techniques or algorithms commonly used in CG have also been applied to achieve efficient snow accumulation covers on objects. Accessibility to the sky for computing realistic snow deposition quantity on the targetted objects is a common focus among several of these approaches. Hence, the focus of the approaches in this category is to gather the relative (and not accurate) contribution of snowfall on different objects in the scene. Some of these methods also attempt to capture the stability of deposited snow mass on objects with the help of lightweight physics.

Fearing [Fea00] presents a method to model the incremental accumulation of snowfall on various surfaces and also account for the stability of accumulated snow. The targetted surface is meshed to generate progressive level-of-detail (LOD) refinements with time. To account for accumulation on the mesh and its accessibility to the open sky, snow particles are shot upwards towards the sky from the *launch sites*. Whereas parameters such as density, accuracy, computation, importance ordering, etc., are easily controlled by the shooting approach, the amount of snow distribution on potential launch sites is facilitated by dividing the sky into a grid as constant-size buckets. The stability test on the surface geometry is tested iteratively based on the angle of repose, edge proximity, etc. Yanyun et al. [YSHW03] propose a hybrid multi-mapping technique to the end of creating a snow cover in natural scenes (Fig. 11a). Whereas the snow cover on near objects in any scene is obtained through displacement mapping, snow deposition on distant objects is approximated utilizing a volumetric texture. The amount of snow deposited on different objects is determined by first adding snow particles to the scene in the vertical direction and then computing the probability of particles sticking to the voxels. Both the displacement map and volumetric texture are concurrently rendered using ray tracing.

Techniques popular for computing lighting and shadow effects in CG have also been successfully applied to approximate snow deposition on objects. A part of the lighting or shadow computation pipeline entails determining the amount or absence of light or photons reaching from the source to each rendered polygon. To this end, the contribution computing algorithms can be mapped to calculate the approximate quantity of snow reaching different polygons in the scene. Ohlsson and Seipel [OS04] compute snow accumulation on objects as a per-pixel effect with the help of a method similar to shadow mapping. The accumulation received by various triangles of any polygonized surface depends on factors such as normal, inclination, etc., and is guided by a prediction function. Vertex displacement and noise computation are performed on a GPU using the shader programming language. Foldes and Beněs [FB07] present a GPU-oriented method to simulate occlusion-based accumulated snow on the terrain for distant views. Snow is modeled as a 3D layer added to the input scene. The standard ambient occlusion lighting technique is used to predict the shape and location of snow accumulation and direct illumination from the skylight to simulate the dissipation of accumulated snow with melting. [Tok06] applies the shadow buffer method for simulating snow cover on objects that block snowfall. In addition, the influence of irregular snow motion is accounted for, and the final snow-covered shape is determined by moving the snow to a stable position. [MT10] employ the pre-computed radiance transfer (PRT) technique to model the accumulation of fallen snow while also incorporating effects such as flying, hitting, and falling snowflakes due to the wind and obstacles. Snow accumulation and melt are generated using a depth map in [CR15]. An *exposure map* of the environment is constructed from occlusion and shadow area computation. Factors such as dispersed wind direction and occluder slope are considered to generate snow accumulation.

3.3.2.2. Heightmap

Another procedural approach to compute snow accumulation in 3D scenes, small or large, is using a heightmap. Given a base ground height value for each part of the targetted terrain, the heightmap techniques entail determining the increment in the height value caused by snow deposition. Some methods in this classification target large-scale terrains wherein the accessibility of snow reaching different polygons in the scene is not as important as in the previous classification of standard CG techniques, and hence, often ignored. Snow cover on the alpine terrain is generated using image-based clues in Premože et al. [PTS99] as shown in Fig. 11b. With the help of the environment's geometry, the method employs grayscale aerial imagery to produce color views of alpine scenes. Large-scale snow accumulation, ablation, and melting behavior are simulated by dividing the height field into various elevation levels. A real-time multi-scale snow cover approximation and visualization on GIS-based terrain data is proposed in [NAS18] on the GPU (Fig. 11c). The precipitation content over the landscape is derived from the GIS data, and snowmelt is computed using a simplified solar radiation model, which is approximated using LOD. For the close-up regions, the terrain height is modified to account for snow cover, and a zero-thickness snow cover generates the illusion of snow for distant views.

A practical use case of heightmaps is in generating snow deformation covers in small outdoor scenes that could capture limited interaction of objects with snow, for instance, the creation of footprints as characters walk on the snow. Different procedural techniques have been used to this end; many of them have a prior step dealing with a particle-based emitter (covered in Sec. 3.4) to generate particles that are deposited. Haglund et al. [HAH02] simulate snow accumulation in different stages. Starting with a snow-free environment, snowflakes are deposited using a particle system on a 2D matrix that corresponds to the ground heightfield. Special care is taken during triangulation to avoid zig-zag patterns and at the snow edges. [FG09] propose a statistical snow distribution model to derive the shape formation of freshly fallen snow over the objects (Fig. 11d). The method produces an enhanced height span map on targetted objects, which is validated against some real-life cases of snow deposition. Reynolds et al. [RLD15] achieve accumulation of snow on 3D object surfaces in dynamic scenes in real-time on the GPU. To this end, surface-bound accumulation buffers are mapped to each object in the scene (represented as mesh), each of which effectively acts as a height map to accumulate snow. For the garden scene with 250K polygons, a performance of 65 FPS is achieved. [Sve19] generate and render deformable snow covers to model the interaction of snow with objects, such as tracks from wheels. A dynamic heightfield is used for bookkeeping the terrain and snow height, and the midpoint displacement algorithm is used to displace the snow.

3.3.2.3. Others

Apart from using standard CG techniques and heightmaps, other methods have tried a different procedural approach to generate snow accumulation. [NVI04] uses an entirely procedural method for calculating the snow coverage in a scene and also uses this information to modify the mesh and lighting to reflect this snow accumulation. This is achieved by physical parameters such as the exposure and incline coefficient and post-processing effects such as edge and depth aliasing. [KRP19] create a realistic snow accumulation effect in screen space for natural environment rendering on the GPU. In their method, the LOD of available terrain mesh is enhanced by extracting geometrical details from a decal texture of the terrain model. [JP20] describes a real-time method (using compute shaders) to implement dynamic manipulation of meshes to simulate the behavior of loosely packed snow when manipulated by objects or agents in a virtual environment. In [Han21], a GPU-based real-time snow rendering technique using compute shaders is presented. The deformation is achieved with the help of a sliding window that moves with the player, allowing the technique to be of use in open-world games.

Discussion: Although only a few listed methods demonstrate the procedural deformation of snow accumulation covers, most computer games find it easiest to adopt a similar approach for their snow environments in real-time with some pre-stored computations (for instance, mesh deformation) and a few variables. Tab. 2 compares some of the above-discussed methods for their nature, simulation scale, CPU/GPU support, maximum resolution demonstrated, simulation time, and properties. Some of these techniques [RLD15, JP20] demonstrate interactive or real-time frame rates for limited outdoor environments. Even with the increasing strength of hardware over the past several years, most techniques targetting landscape-scale snow cover generation require much higher computational times owing to the scale of the problem.

Figure 12: *(a) Physics-based winter storm captured in [HHP*21], (b) snowfall modeled through artist-generated textures in [WW04].*

3.4. Snowfall

In the context of snow-related methods in CG, producing the effect of falling snow is also a requirement in many applications. Many accumulation methods require a snowfall simulator that leads to a realistic snow deposition on the scene objects over time. The application of snowfall varies from adding the effect of falling snowflakes in the scenes to possibly detecting or removing the presence of snowflakes in the images (or videos) in computer vision, image processing, and other related disciplines. Since the process of snowfall is simple, the use of previously discussed simulation methods is an overkill of computation for this purpose. In addition to physics-based methods (Sec. 3.4.1), snowfall is shown to be successfully modeled by various image-based inexpensive methods (Sec. 3.4.2). Both space and image-based procedural techniques can be employed to this end.

3.4.1. Physics-based

Almost all the physically based snowfall methods employ particles to reproduce the effect of falling snow in CG scenes. Additionally, several snowfall simulators require physics for various purposes, for instance, wind field generation, weather simulation, snow adhesion, etc. Some of the presented methods also adopt simplistic models to the end of computing snow accumulation on the scene objects caused by falling snow particles. If accounted for, the shape and size of snow crystals can vary, and these varied shapes are also incorporated in some works.

Simplistic simulators (termed as *particle systems*) are often employed to emit snow particles with varying spatial distribution and temporal frequency in given 3D scenes. Such simulators handle the basic particle movement by accounting for gravity, buoyancy, vortex field, other forces, and, in some cases, simple accumulation models as well. The wind field itself can be generated using different physics models. Muraoka and Chiba [MC00] describe a method to reproduce the behavior of snowflakes in the surrounding environment (air currents, gravity, etc.) using virtual snow particles. The

Method	*Type*	*Scale*	*CPU/GPU*	*Maximum Resolution*	*Simulation Time*	*Properties*
Hinks & Museth [HM09]	physical	limited	CPU	256×256×256	14,400 s	wind-driven level set, limited physics
Maréchal et al. [MGG*10]	physical	landscape	CPU	3.2M voxels	18,000 s	heat transfer over nine days period
Festenberg & Gumhold [FG11]	physical	limited	CPU/GPU	700×700	654 s	diffusion-based snow cover
Cordonnier et al. [CEG*18]	physical	landscape	GPU	1024×1024	1,172 s	110 frames of snow cover evolution, weather events
Argudo et al. [AGP*20]	physical	landscape	GPU	1500×1500	120 s	Glacier evolution over 635 years
Reynolds et al. [RLD15]	procedural	limited	GPU	250K polygons	0.0154 s/step	occlusion-based, heightmap
Neukom et al. [NAS18]	procedural	landscape	GPU	166×118	0.145 s/step	GIS-based cover, limited physics
Junker & Palamas [JP20]	procedural	limited	GPU	2048×2048 texture	0.02 s/step	compute shaders

Table 2: *Various snow accumulation methods compared for: the type of method (physical or procedural), whether it is based on the landscape- or limited-scale outdoor settings, CPU- or GPU-based, the maximum resolution used, and the time required (or frame rates) to generate the maximum resolution. Correction added on 9 May 2024, after first online publication: There were a few numerical errors in the Method for "Argudo et al. [AGP*20]" in Table 2 which are now corrected.*

method also creates accumulated shapes of the fallen snow while accounting for heat propagation (from sunlight, ground, and radiation) that leads to snowmelt. Real-time snowing simulation leveraging Boltzmann equations to model the interaction of falling snow and wind is proposed by Wang et al. in [WWXP06]. The deposition and erosion processes in 3D space are influenced by the wind and captured with the help of a height field. The particle density is controlled with LOD-based distribution in the scene, and fuzzy motion is leveraged to enhance the visual effect. Simple particle systems subjected to Newtonian physics (gravity, buoyancy, horizontal wind force) are simulated in [TZWZ09, TL15].

[ZZL11] propose a simple particle-based snowfall simulation effect in virtual scenes based on *OpenSceneGraph*. [FZ12] generate snow and rain in virtual environments based on simple particle kinematics, wherein the particles are texture-mapped to convey the appearance of falling snowflakes. [DZC*13] use a particle system to simulate the rain and snow effect in real-time, together with large textures that help reduce the particle number. Herrera et al. [HHP*21] propose a unified physics-based framework to simulate weather at interactive rates (Fig. 12a). Different precipitation types, such as snow, rain, and graupel, are modeled by introducing a microphysics scheme. The main focus, though, is on the microphysics of cloud formation covering snow/ice crystal growth occurring in mixed-phase clouds.

Few works place exclusive emphasis on capturing the crystal structures of falling snowflakes in addition to applying basic physics to handle the movement of snow particles as they fall. The shape of snowflakes is modeled by compositing various possible ice crystal shapes in hydrology by Moeslund et al. [MMAL05], wherein the size and density are simulated as a function of the air temperature. A complex crystal shape is obtained by modeling each snowflake within a concentric layered structure to combine triangular polygons. The movement of created snowflakes is guided by the basic physics forces. The wind field itself is described by the Navier-Stokes equations and consistent with the standard semi-Lagrangian fluid solvers; convection and projection steps are employed. In [ZXZ10], an algorithm to generate snowflake texture on the GPU is presented. The shape and attributes of snowflakes are controlled by weather parameters (temperature, humidity, etc.), similar in principle to [MMAL05]. Real-time rendering of falling snowflakes is achieved using various elementary particle types to simulate different snow structures in [TF11], and the wind field is simulated with the help of Boltzmann equations. A large and near-to-exhaustive variety of crystal forms falling from the winter clouds in the form of snowflakes are covered in [Lib16].

Parallel methods with a combination or variation of the techniques mentioned above to simulate snowfall have also been investigated since particle emitters can often be easily parallelized. A simple Newtonian kinematics-based model on particle systems simulates rain and snow in [YZMC08] on the GPU. Saltvik et al. [SEN06] parallelize snowflakes and their interaction with the wind on multi-core processors. For this purpose, Navier Stokes equations are used to model the falling snow. A snow texture on objects exposed to the sky is generated on the GPU to model accumulated snow; hence, no physics is used. A CUDA-based framework on the GPU for snow simulation is proposed in [Eid09]. The effect of wind is generated using Navier-Stokes equations in the grid setting, and a terrain height map model aids in computing snow accumulation. The falling snowflakes follow a particle-based approach insofar as the force treatment is concerned, with gravity, buoyancy, lift, and drag forces incorporated into the model. [ZCL10] simulate whirling snowflakes in the sky and the snow deposition effect on objects using particles on the GPU.

3.4.2. Image-based

Several approaches have been developed that can add, detect, or remove snow phenomena in the image space in the frequency domain without needing any physics or object-domain manipulation. This mainly includes snow (rain) detection, removal, or addition in a given image or video. Although most of these methods fall under the image or signal processing domain, they are still somewhat relevant to this report since they allow adding or removing snow-related artifacts in the scene. An advantage of such image-space approaches is that snowfall can be added to any given scene in a postprocessing step without requiring its three-dimensional representation.

Only some works have proposed adding image-space snowfall to a given video. Falling snow is rendered using an inverse Fourier transform by Langer and Zhang [LZ03] in images/videos. To this end, a set of surfaces is first generated in the frequency domain, followed by an inverse 3D Fourier transform step. [WW04] model falling snow with four artist-generated textures, mapped onto a double cone with the camera centered in between them (Fig. 12b). In addition to blending the four textures, other tricks, such as texture elongation and scrolling, can achieve effects such as blurring, gravity, etc. A spectral-particle hybrid method for rendering falling snow is presented by Langer et al. [LZK*04]. The method captures the dynamic texture properties of falling snow by using a sparse particle system and a dynamic texture fill inspired by the particle system. The texture is generated using an image-based spectral synthesis method and allows texture variation in both the space (image) and time domain. Camera motion can also be matched, and adding snow effects to static and dynamic scenes is demonstrated.

On the contrary, much more research exists that targets the detection or removal of snow in images or videos. A primary application of many snow removal algorithms is creating clearer outdoor environments than those captured by cameras under snowy conditions. We have identified three major categories of manipulation algorithms, which target to remove snow or rain from images.

3.4.2.1. Signature analysis

Snow can be detected in images by identifying the signature of snow streaks (for instance, using an existing or on-the-fly database) and matching them against pixels of the provided image. [BNK10] develop a model of a single rain/snow streak in the image space, the statistical characteristics of which are used to generate the effect of rainy/snowy weather in the frequency space. This method can detect, increase, or remove snow streaks in the provided image or video. A guidance image method to remove rain and snow in a single image is proposed in [XZLT12a]. The guidance image is derived from the imaging model of a snowflake through a charge-coupled device camera and refined later. The snow removal based on the intensity of the image pixels is carried out with the help of the refined guidance image. Cai et al. [CPS*06] propose an algorithm to detect moving snow in video streams. This is achieved with the help of an automatic incident detection (AID) system, which improves the detection rate in the area of video frames containing snow. The algorithm consists of a series of processing steps (glare processing, background generation, snow sample correlation, and final snow map production) to create a snow map indicating whether pixels contain snow or not.

3.4.2.2. Frequency splitting

In many cases, it is possible to sieve out the characteristic frequency differences between snow and other background components in the image instead of looking for a particular signature. In [RM13], snowfall is removed from an image by decomposing the image into low-frequency and high-frequency components using a bilateral filter. The high-frequency part is further decomposed into the snow and non-snow components utilizing other techniques like dictionary learning and sparse coding. Zheng et al. [ZLG*13] propose a method for snow and rain removal from a given image by separating the image into high and low-frequency components. The low-frequency component captures the phenomena of rain or snow, which is then removed with the help of a guided filter. Bossu et al. [BHT11] present a system based on computer vision that detects the presence of rain or snow in image sequences. After separating the foreground from the background, selection rules based on photometry and size are used to select the potential rain streaks. After that, a histogram of orientations of rain or snow streaks (HOS) is estimated, which represents the orientation of the rain or the snow. With the help of HOS, pixels containing rain or snow in the foreground image can be detected. Xu et al. [XZLT12b] apply a guided filter for snow removal from the image, which does not require any pixel-based statistical information for detecting rain or snow. In [PTL14], image features based on saturation and visibility are leveraged to remove the snowflakes and raindrops from given images.

3.4.2.3. Artificial intelligence

The use of artificial intelligence (AI) is gaining more popularity in image manipulation. In the context of snow, AI enables more straightforward automation of one or more of the techniques discussed above. An algorithm to remove snow and rain from a given image is presented in [WLCZ17]. Their method decomposes the image into low-frequency or snow-free and high-frequency or snow components. Thereafter, a 3-layered hierarchical scheme is applied on the high-frequency component, wherein an overcomplete dictionary, guided filtering, and variance sensitivity are employed on respective layers to remove snow or rain from this image component. AI has been used to remove snow from images in [LJHH18]. This method eliminates the need for handcrafted features for this purpose and employs a multistage network *DesnowNet* to handle the removal of translucent and opaque snow particles. [VBVB19] simulate photo-realistic snow and fog on existing image data sets for enhanced convolutional neural network training and evaluation.

4. Ice in CG

Ice, like snow, is made up of water. Even though the process of ice formation shares some similarities with that of snow, there are marked differences in the physical appearance and properties of ice. Ice is usually denser than snow and has a different crystalline structure. It can be formed directly due to the freezing of water or the exposure of snow to pressure. Hence, contrary to the processes of accumulation or interaction with other solid bodies in the case of snow, the processes of growth, melting, etc., and hence thermodynamics are the focus areas when dealing with ice animation in CG. Consequently, research in CG has paid more attention to capturing these physical features of ice and the underlying physics separately from snow. While most methods rely on a simplified physics-based foundation to develop the algorithms to this end, some procedural methods do exist, as discussed below. The structure of the remainder of this section on ice is outlined in Fig. 13.

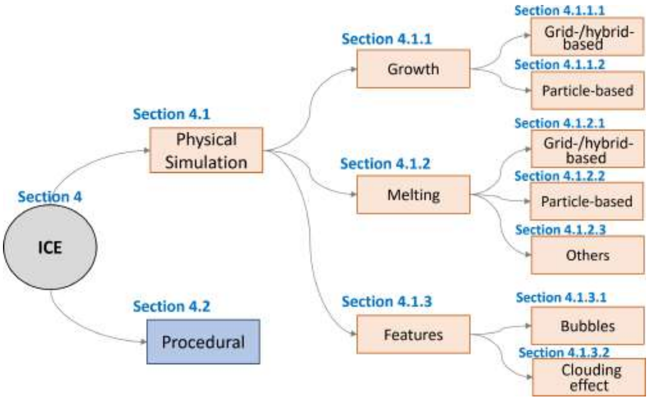


Figure 13: *Organization of the sections on ice in this report, based on the published works in CG.*

4.1. Physical Simulation

In addition to reproducing the physical phenomena of freezing (growth or formation) and melting, certain visible features like bubbles trapped within the volume have been the focus in the case of ice. Interestingly, the techniques designed for simulating snow (Sec. 3) do not suffice here. In the context of ice, we will cover the physical simulation models to achieve the growth and melting of ice in the following, followed by outlining the few existing procedural works. In the actual world, freezing and melting of ice are considered phenomena opposite to each other. Consequently, one would assume that a physical method designed for simulating freezing should be able to handle melting efficiently as well, and vice-versa. However, as discussed below, dedicated CG techniques are more efficient in handling either of the two processes separately. Additionally, unlike most snow simulation methods, most freezing or melting methods need to account for the heat exchange with the environment through temperature fields, thermal photons, or something similar.

4.1.1. Growth

Some work has been done to reproduce the behavior of ice, icicle, and glazed frost using grid-, hybrid-, and particle-based methods. Icicle refers to the tapering mass of ice formed by freezing, dripping water. On the other hand, glazed frost is a smooth, transparent, and homogenous ice coating formed on a surface. Unlike approaches targetting to capture the melting phenomenon, freezing or growth is confronted with reproducing the relatively minute crystal structure generated by water freezing on various surfaces.

4.1.1.1. Grid-/hybrid-based

Multiple grid-based and hybrid methods have successfully demonstrated the process of ice growth. Kim and Lin [KL03] use the phase field method to achieve interactive ice crystal growth on the GPU. In the phase field method, the undercooled liquid is represented implicitly as a 2D or 3D grid, containing a temperature and phase field for each grid cell. In their work, thin 2D crystals are simulated for growth instead of 3D (for efficient computation), followed by adding the thickness dimension. Banded computation is performed wherein adaptivity prioritizes the growth on the ice-water front. The seed crystal is supplied to their model by extracting visually salient features of a target object, and growth is influenced by manipulating the freezing temperature. The simulation can also introduce postprocessing effects, such as creases into the ice at visually expected locations. The phase field method is inspired by Kobayashi's formulation [Kob93], which can achieve a wide variety of 2D ice growth patterns by controlling a few parameters in the reaction-diffusion partial differential equations (PDEs) containing the temperature and pressure.

Kim et al. [KHL04] propose a hybrid algorithm for the physically-based modeling of ice formation wherein three techniques, namely, diffusion-limited aggregation, phase field methods, and stable fluid solvers, are combined (Fig. 14a). The benefit of combining these methods is the ability to capture the entire crystalization process using a grid-based solver. The PDE in [KL03] is modified to include the rate of phase change due to diffusion additionally. An efficient solution method to solve the simplified formulation that accelerates the phase field method by more than two times is also presented. The method incorporates factors like fluid flow, boundary/obstacles, humidity, density, temperature, etc., in the crystal growth.

Kim et al. [KAL06] further proposed a physically-based algorithm for modeling ice formation by using a level set approach to the thin-film *Stefan problem* (Fig. 14b). The classic Stefan problem is a phase transition problem that describes the evolution of the boundary between the two phases undergoing a phase change. Even though the level set method is chosen to track complex surface features that emerge during ice formation, it has been observed that even a subscale or adaptive grid is insufficient to resolve some features that are orders of magnitude smaller than the overall domain (for instance, the tip of an icicle). This is handled by an analytical solution for the dynamics in combination with a curvature-dependant evolution equation for the ice away from this tip. A ripple formation model (Ueno model) is introduced to generate ripples or non-smooth features occurring in ice formations.

Apart from leveraging the grid-based physics, there are similarities between the above-discussed works [KL03, KHL04, KAL06] for surface ice growth. They all rely on a PDE-based formulation of the form $\frac{\partial T}{\partial t} = c_1 \nabla^2 T + o$ involving temperature T and other various factors o (c_1 is a constant) to solve the grid equation. They allow for the user intervention to shape or manipulate the growth process. [KL03, KHL04] even generate ice features that resemble microscale crystal details.

Glaze: Unlike obtaining conventional ice freezing on surfaces, glaze formation has been the focus of some grid-based approaches. Glazed frost is the crystal clear form of ice formed from supercooled rain droplets when they hit the surfaces of objects. [IDY*13, IYW*15] present a hybrid grid-particle method (FLIP [ZB05]) to simulate this freezing phenomenon (Fig. 15a). In their work, raindrops and objects are represented using particles, which simplifies advection, and the velocity is updated on the grid cells (Eq. 8, where κ represents the adhesion coefficient, $\boldsymbol{p}$ are the particle positions, Q is the transferred heat, $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity, and k is the thermal conductivity). The heat transfer between

(a) (b) (c) (d)

Figure 14: *Ice growth algorithms (a) snowflake growth on a chilled glass simulation simulated using the physically-based hybrid method in [KHL04], (b) physics-based modeling ice formation by Kim et al. [KAL06], (c) particle-based physical freezing model by Miao and Xiao [MX15], (d) procedural icicle modeling on a tree branch by Gagnon and Paquette [GP11].*

particles and the outside grid is also accounted for using standard laws of thermodynamics. The adhesion force ($\boldsymbol{f}_{adhesion}$) between water and ice particles is computed using surface and interfacial tension.

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \boldsymbol{\nabla})\boldsymbol{v} = -\frac{1}{\rho}\boldsymbol{\nabla} p + \nu \boldsymbol{\nabla}^2 \boldsymbol{v} + \boldsymbol{f}_{ext}, \quad \boldsymbol{\nabla} \cdot \boldsymbol{v} = 0,$$
$$\boldsymbol{f}_{adhesion} = \sum_j \kappa \frac{\boldsymbol{p}_i - \boldsymbol{p}_j}{\left\|\boldsymbol{p}_i - \boldsymbol{p}_j\right\|}, \quad \Delta Q = -kA\frac{dT}{dx} \tag{8}$$

4.1.1.2. Particle-based

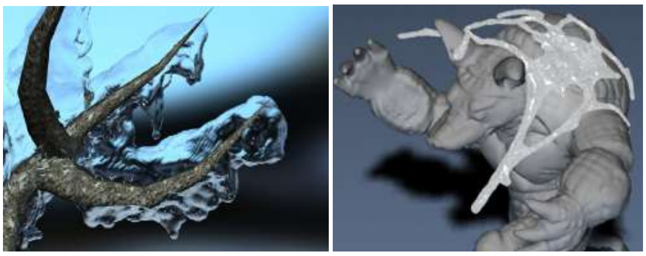
Several approaches use SPH or other particle-based frameworks in CG to simulate various ice phenomena. Assuming a particle-based discretization often makes mass conservation, heat transfer, phase change from solid to liquid (or vice versa), etc., much easier to handle when compared to the grid-based methods. Kharitonsky and Gonczarowski [KG93] present physically-based modeling for the naturally occurring process of icicle growth. Their model considers the mechanical (surface tension, contact angle, wetting) and thermodynamical (heat transfer, conductivity, phase changes) variables involved in the process. Icicles are created separately using particle-like drops and then joined to form clusters to imitate natural formations in nature. The entire process is simulated iteratively, thereby combining the simplified mechanical and thermodynamical processes together with the clustering.

In [MX15], an SPH-based approach inspired by position-based dynamics [BKCW14] is leveraged to simulate the freezing phenomena in flowing water (Fig. 14c). The heat transfer equation (similar to PDE used in the grid-based methods) controls the freezing speed and direction, and the appearance of opaque frozen ice is enriched by employing smoothly diffused air bubbles. To this end, the diffusion of air bubbles is computed from the recently solidified particle to its liquid neighbors. The method also accounts for the time-based volume expansion and deformation of ice by altering the density of particles according to their phase by employing XSPH density. In [Blo14], the temperature and heat conduction are included within the framework of SPH to simulate phase transitions (freezing, melting) of fluids.

Glaze and icicles: Like grid-based approaches, particle-based methods have also been proposed to animate glazed frost formation or similar phenomena. Im et al. [IKK*17] present a particle-based framework to simulate the freezing of flowing water in glaze or directional icicles on the GPU. The method considers factors such as the direction of the flowing water, humidity on object surfaces, phase transformation through nucleation energy, etc., to guide the growth and direction of icicles. The concept of nucleation energy, which depends on the humidity and water flow, determines whether a water particle at the water-ice boundary should be converted to ice. Kim and Lee [KL21] present a particle-based method to express the directional ice formation and glaze effects caused by the freezing of flowing water. It can generate both freezing ice and glaze effects (Fig. 15b). The freezing solver is based on the IISPH [ICS*14] method to solve for pressure forces. The modified solver integrates fluid motion into the growth direction of ice and also integrates improved surface tension and viscosity terms into the solver to alleviate the effect of scattering motion due to impact by collision of freezing ice. The ice growth direction is determined from the fluid motion direction, which is calculated based on the velocity of the fluid particles. This approach generates a virtual film by transferring fractional mass from the liquid to adjacent solid particles. A nucleation energy function based on humidity, water flow, and transferred mass is proposed to turn water particles into ice. This work relies on the fundamentals of SPH to compute several physical quantities to capture the border physics between the fluid and ice particles, which is where the freezing process happens. It is interesting to note that contrary to the grid-based methods, particle-based methods targetting glaze formation place a special emphasis on the directional aspect of ice formation based on the fluid velocity.

4.1.2. Melting

Melting ice is of interest to certain applications in CG. As opposed to the growth process wherein the appearance of formed ice patterns on surfaces is crucial, the focus here lies in reproducing the melting effect for an already frozen shape (volume or protruded surface) of ice. Therefore, some research solely deals with the convincing melting of ice rather than the entire freezing and melting process. Nevertheless, the freezing and melting approaches share many similarities, especially regarding the underlying physical tools employed.



(a) **(b)**

Figure 15: *Physics-based models for glaze growth (a) glazed frost simulated by Ishikawa et al. [IDY*13], (b) ice surface and glaze effects produced when water is thrown in the left direction by Kim and Lee [KL21].*

(a) **(b)**

Figure 16: *Physics-based algorithm for melting (a) interactive particle-based melting of ice by Iwasaki et al. [IUDN10], (b) ice simulation with water flow handling by Lii and Wong [LW14].*

4.1.2.1. Grid-/hybrid-based

Matsumura and Tsuruno [MT05] generate melting ice animations considering the thermal energy transfer between the ice and the surrounding air. For this purpose, the air calculation field, derived from the Navier-Stokes equations, and the object calculation field, representing ice, is used. The thermal energy transfer is calculated between the two fields using a similar but reverse process employed in freezing. Fujisawa and Miura [FM07] introduce a technique similar to the photon mapping for heat calculation in physically-based ice melting and solve for the thermal radiation phenomenon by storing (and accumulating) the thermal energy in each photon. The use of photon mapping allows for thermal diffusion and convective thermal transfer in the scene. The method accounts for thermal diffusion, convective thermal transfer, and thermal radiation. Thus, it can handle various melting scenarios, including ice objects in hot water, air, etc. The fluid behavior itself is simulated with the help of Navier-Stokes equations for an incompressible fluid.

4.1.2.2. Particle-based

Iwasaki et al. [IUDN10] simulate the effect of melting and freezing of ice objects using heat transfer on a particle-based framework (Fig. 16a). The heat from the source to the object is carried by emitting and receiving thermal photons. The heat exchange is facilitated by the SPH variation of the PDE employed in the grid-based methods, $\frac{\partial T_i}{\partial t} = c_d \sum_j m_j \frac{(T_j - T_i)}{\rho_j} \nabla^2 W(\boldsymbol{r}_i - \boldsymbol{r}_j, h)$ where c_d is the thermal diffusion coefficient and T represents the temperature of a particle). In addition to capturing the meltwater flow on ice and the formation of water droplets, an interfacial tension (similar in nature to [IYW*15]) is proposed to handle the water-ice and water-water interaction based on the SPH. The method is implemented on CUDA and can operate at interactive frame rates with simulation and rendering combined.

A particle-based approach for ice melting simulation is demonstrated in [LW14], see Fig. 16b. Each particle tracks the amount of surrounding water content through an attribute called *virtual water*. This virtual water transfer between the exterior ice particles enables the simulation of the thin layer of water flow on the ice surface. During the transition of ice melting, the virtual water and water particles are allowed to exchange their states. The heat transfer from any heat source to the particles is performed through the photon mapping method.

4.1.2.3. Others

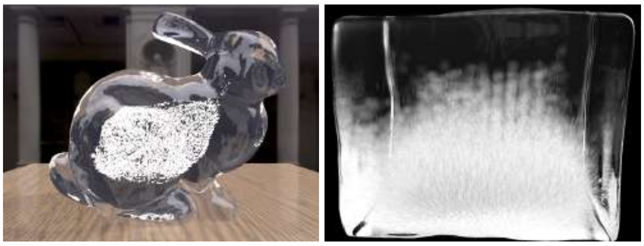
Some other physically-based methods related to ice that do not target freezing or melting exist. For instance, an adapted Lattice Boltzmann-based free surface flow solver simulator on the GPU is presented in [JMR17] for handling fluid-ship-ice interactions. The rigid body interaction is limited to already broken ice floes and the ship hull and is intended to optimize a ship hull's capability to clear the ice. To this end, the method uses a volume-of-fluid (VOF) interface capturing approach.

4.1.3. Features

The appearance of generated ice can be further enhanced by adding other features like bubbles and scratches. These effects are best modeled within the physical framework of simulation by accounting for the relevant variables and processes. Nevertheless, these features are often not obtained automatically but are added separately during or after the main physics pipeline.

4.1.3.1. Bubbles

Most methods accounting for bubble formation in ice volume rely on the grid-based framework. In these methods, the phenomenon of ice formation is augmented by additional processes that can account for the entrapping air mass leading to bubble formation. [MO08] simulate ice and bubble formation processes using a grid-based framework to construct geometric representations of ice that integrate bubbles in an ice cube. Their model is physically based, and factors such as the velocity of ice formation, levels of air concentration, and simplified heat transfer are accounted for to evolve the ice-water interface. Madrazo et al. [MTSK09] propose a physically-based method for creating a photorealistic animation sequence of simulating freezing ice with bubbles using a grid-based method. The technique considers phase change from liquid to solid because bubbles nucleate in the ice-water interface during this process. The interface is tracked by a particle level-set method driven by the Stefan conditions. Nishino et al. [NIDN12] present a fast simulation of freezing ice, taking into account air bubbles using



(a) (b)

Figure 17: *Bubbles formation in freezing ice simulation by the (a) grid-based method of Nishino et al. [NIDN12], (b) particle-grid method of Im et al. [IPKK13].*

CUDA (Fig. 17a). Though a grid-based framework is employed to simulate the freezing, bubbles trapped in the water-ice interface are represented with spheres. To this end, the weight of dissolved air at each grid point is computed. In [IPKK13], a particle-grid method to simulate the generation of opaque ice with trapped air bubbles is presented (Fig. 17b). The dissolved air is exchanged between ice and water particles, and the diffused temperature is maintained over the grid, which can handle spatially varying heat transfer rates. A particle is assumed to be an air bubble if it has sufficient air; otherwise, it is dissolved into the surroundings. The use of the signed distance field function (SDF) enables the generation of opaque ice volumes varying vastly in geometric shapes.

4.1.3.2. Clouding effect

Another way to make ice appear more realistic is by reproducing the damage effects. To this end, the generation of cloudy effects and scratches on ice due to collision with objects are handled in [KIKL16]. These effects are produced and diffused in proportion to the intensity of a collision, and scratch generation is also achieved using the density gradient of cloudy effects. For this purpose, a grid-projection technique is combined with the boundary particle method.

Tab. 1 summarizes the various snow and ice simulation methods on various attributes like the underlying model used, time step employed, support for GPU, maximum resolution demonstrated, time taken to compute per simulation step and other properties. It is evident that both particle- and grid-based methods can support larger time steps (even if the magnitude does not differ) in the case of ice simulation when compared to the snow methods.

4.2. Procedural

Some other works target one or more aspects of ice relevant to the CG community with purely procedural methods. Gagnon and Paquette [GP11] propose a procedural approach for interactive icicle modeling and glaze ice formation (Fig. 14d). Their algorithm consists of four sequential stages: computation of the water flow on the provided input mesh, determining the dip points, computation of the icicles' trajectories, and finally, computation of the icicles' surfaces. The user has a certain degree of interactive freedom through each stage of the algorithm execution. [FA01] take a different approach to ice thawing by using mathematical morphology and cellular automaton to the end of volume modeling. Effects such as thawing, dilation, erosion, and regelation are accomplished through morphological operations, thus taking no ice physics into account.

[ADP*15] design and implement a parallel ice simulator to capture the behavior of a ship operating in pack ice. Ice floes are modeled as a set of discrete objects with simple properties, and the system mechanics as a set of discrete contact and failure events. This approach comprising objects and events is termed event mechanics modeling.

Discussion: The discussed particle-based methods for ice growth and melting in CG closely resemble the grid-based methods in their underlying physics and differ significantly from their snow counterparts. Temperature field is a common factor shared by both processes, while other specific variables like humidity in case of ice growth and heat transfer from the source in case of melting need to be incorporated in any targetted physical method. Additionally, the amount of heat an ice unit is exposed to must be stored for melting. The heat is modeled using various means, including temperature field, thermal photons, etc. Factors like deformation and plasticity in real-time interaction with other objects are not of much relevance for ice as in the case of snow. Whereas specialized growth algorithms are devised for the glaze or icicle formation, standard algorithms could be used to capture their melting process. Whereas the micro-level crystal structure is important in shaping the look of freezing water, the bond formation and relative movements between constituting discrete ice units are often ignored. This is a big difference when comparing it to the snow physics in CG. In many applications, the problem of ice formation can be reduced to obtaining a thin icy layer on object surfaces or to a two-dimensional layer with some depth addition.

5. Discussion

Based on the findings of this report, we have been able to make several observations. Fig. 18 lists the major landmark physically based methods developed to simulate snow, ice, or accumulation in CG. It is evident that most research in these fields has been done within the last two decades or so. Furthermore, the focus and approach of research work in several disciplines (civil engineering, snow mechanics, etc.) is different than in CG. In CG, capturing the simulated material's visual properties is essential, making it appear similar to snow. In addition to the basic physical properties, these methods often take the liberty to use a set of higher-level variables or handles to reproduce and modify the visual properties of snow and ice. In some cases, even the physical parameters employed can be manipulated significantly to push for higher efficiency if it does not hurt the apparent value of visual realism. In contrast, other disciplines work closer to the actual variables involved in the simulation and emphasize a rigorous validation process in order to replicate a more accurate process. On the other hand, the validation aspect in CG is often much weaker or only needs to suffice to the visual perception. Grid-based or hybrid models have been the earlier choice for snow simulation in the basic sciences and CG. However, other models have recently been shown to be as powerful in capturing snow-object interaction and other properties as traditional grid-based models.

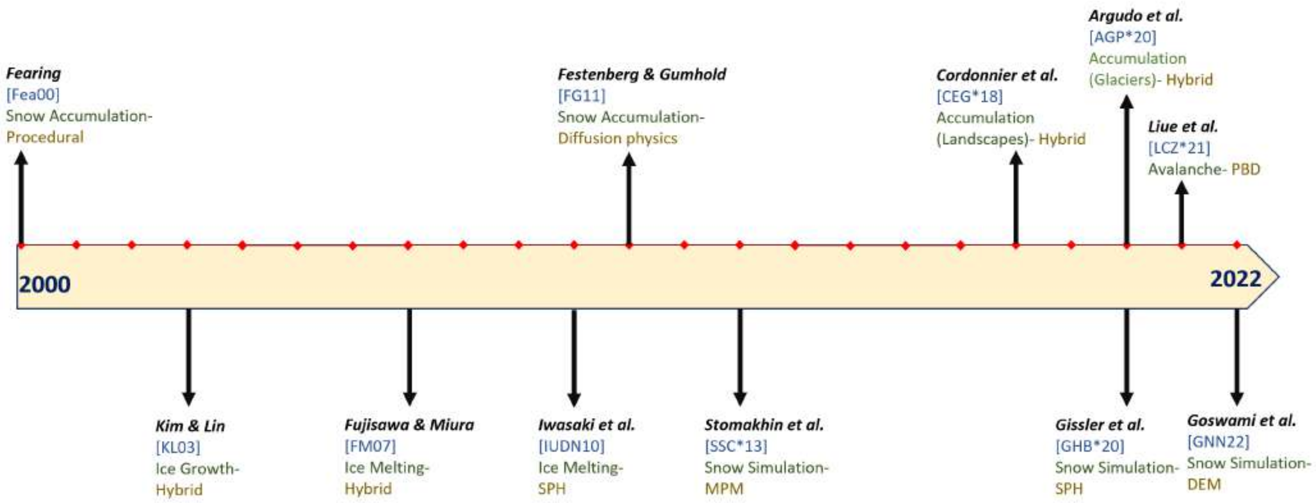


Figure 18: *Chronology of the major landmark papers in the field of CG to simulate snow, ice, or accumulation.*

Snow-tire interaction is identified to be of importance both in the base scientific community and in CG. In the former, DEM models are known to capture this best. In CG, particle-based models exist to animate various snow-related phenomena. Some SPH-based models [GHB*20] recently successfully captured desired effects. Interestingly, most of the physics-based snow and ice simulation methods rely on the established fundamentals of fluid simulation, for example, Navier-Stokes equations. Whereas the divergence is preserved in the grid-based methods, most existing particle-based methods do not enforce divergence.

Similarly, much more work has been done to understand and simulate avalanches in civil and agricultural engineering. This is due to the apparent reason for minimizing the damage caused by avalanches in risk-prone areas. Few works on this topic exist in the CG community, most of which have been done in the last decade. Very little to no work in CG has explored the braking effect of forests and other obstacles on the flow and intensity of avalanches. Also, there is a more extensive scope to simulate various types of avalanches both procedurally and physically. The material point method has gained popularity for snow and avalanche simulation. More interesting is that this is the case both in the base scientific community and in CG. This reflects the power of this method to enable simulations with lower-level scientific variables and higher-level handles, as is often required in CG.

Procedural methods to handle snow and ice simulations exist, but the functionality is often limited to model shape formation, collision response, etc. On the other hand, procedural methods (without or with limited physics) to generate a snow cover on objects or landscapes in outdoor scenes are popular. This is in tune with the requirements of applications that do not require small-scale simulation details but focus more on achieving a realistic cover in a limited computational time. Similarly, the effect of falling snow in scenes can be achieved with the help of physics-based, object-domain particle systems or image-based manipulations.

Due to the larger scale, physics-based methods of computing snow accumulation in outdoor scenes are often much more computationally demanding than simulation methods. Such methods in CG often offer a higher possibility to relax the constraint of reproducing a more accurate micro-scale snow behavior. Despite this accounted relaxation, the high computational time can be attributed to large outdoor settings and time-varying physical processes necessitating multiple variables and simulation loops. Much simpler mechanisms are preferred to detect, produce, or tamper snowfall presence in the scenes. These include either the particle-based generators in 3D or image-based methods that operate entirely on the image pixels.

In the base disciplines, some methods treat snow and ice within the same physical framework by merely changing the involved parameters, for instance, DEM. However, the methods developed for snow in CG are different than for ice as they vary in their focus, too. Dedicated approaches target special forms of ice, for instance, glaze or icicle. Both grid-based (including hybrid) and particle-based methods have been shown to be effective in simulating snow and ice. The performance aspect is yet to receive more attention when simulating snow. There are higher-performing methods dealing with ice than snow. Several models developed with position-based dynamics as their base solver for ice growth or melting can take very large time steps and reasonable time per frame to provide more efficient solutions. However, most of these methods' focus is to simulate only the formation or melting of ice, which involves very gradual relative displacement between particles. The need in snow models is to simulate fracture, deformation, etc., or snow-object interactions, which necessitate stress computation. For this reason, the snow-related methods are not only complex but much slower computationally. Similar to its base solver, the implicit compressible solver can handle relatively large time steps for animating snow. However, the total time required to generate each simulation step limits it to only offline solutions as it is operating on the CPU.

More investigation by porting it on the GPU could provide a better comparison. Comparatively, the ice-object interactions are embedded in the overall pipeline and are captured in a limited way or not captured at all.

The scale of snow simulations often stands at a coarser level in CG than in base sciences. Some methods go to the crystal-level structure in CG only for specific visualizations. However, it is standard in the base science to leverage these structures in mathematical formulations, too. A much finer resolution and, hence, millions of particles are required to reproduce a realistic effect with snow, even in scenes of small scale. This is exacerbated further in outdoor scenes. Even though the accumulation methods find approximate workarounds to eliminate micro-level physics, avalanches can often not escape this predicament. Very few works in CG have focussed on avalanche simulation, and not many procedural approaches have been explored. Overall, to the best of our knowledge, apart from image-based techniques to add the effects of snowfall, no work has exploited the power of artificial intelligence to simulate snow, ice, avalanche, or accumulation.

6. Conclusions

The development of visually appealing and efficient snow simulation methods in CG has gained more momentum recently. Many snow and ice simulation techniques have been proposed in the last few decades but have not been reviewed comparatively. This report has identified and categorized the existing relevant work and presented the advantages and disadvantages of these methods. Our findings suggest that snow and ice can be successfully simulated using different methods based on grids, particles, or hybrid formulations. CG could benefit more from the existing methods in the base scientific field to extend state-of-the-art solutions. Furthermore, future research could aid the development of efficient, real-time methods that could potentially be incorporated in SDKs of other real-time materials. Another promising direction would be to explore the use of artificial intelligence to the end of snow and ice simulation or snow cover generation. The current scope of AI-based methods is limited to the image domain to detect, remove, or add snow streaks in images and videos.

Acknowledgements

This research was funded partly by the Knowledge Foundation, Sweden, through the Human-Centered Intelligent Realities (HINTS) Profile Project (contract 20220068).

References

[AA10] ANDERSEN S., ANDERSEN L.: Modelling of landslides with the material-point method. *Computational Geosciences 14*, 1 (2010), 137–147. 5

[ADP*15] ALAWNEH S., DRAGT R., PETERS D., DALEY C., BRUNEAU S.: Hyper-real-time ice simulation and modeling using GPGPU. *IEEE Transactions on Computers 64*, 12 (2015), 3475–3487. 21

[AGP*20] ARGUDO O., GALIN E., PEYTAVIE A., PARIS A., GUÉRIN E.: Simulation, modeling and authoring of glaciers. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–14. 12, 13, 16

[AKS14] ABDELRAZEK A. M., KIMURA I., SHIMIZU Y.: Numerical simulation of snow avalanches as a bingham fluid flow using SPH method. *River Flow 2014* (2014), 1581–1587. 5

[AO11] ALDUÁN I., OTADUY M. A.: SPH granular flow with friction and cohesion. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2011), pp. 25–32. 7

[Atk92] ATKINS R.: Computer graphics applications in avalanche forecasting. In *Proceedings of the International Snow Science workshop* (1992), Citeseer, pp. 116–125. 5

[Bak19] BAKER I.: Microstructural characterization of snow, firn and ice. *Philosophical Transactions of the Royal Society A 377*, 2146 (2019), 20180162. 2, 3

[BHT11] BOSSU J., HAUTIERE N., TAREL J.-P.: Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International Journal of Computer Vision 93*, 3 (2011), 348–367. 17

[BKCW14] BENDER J., KOSCHIER D., CHARRIER P., WEBER D.: Position-based simulation of continuous materials. *Computers & Graphics 44* (2014), 1–10. doi:10.1016/j.cag.2014.07.004. 11, 19

[BL21] BARRETO D., LEAK J.: A guide to modeling the geotechnical behavior of soils using the discrete element method. In *Modeling in Geotechnical Engineering*. Elsevier, 2021, pp. 79–100. 3

[Blo14] BLOM T.: *Solidification using Smoothed Particle Hydrodynamics*. Master's thesis, Game and Media Technology, Utrecht University, 2014. 19

[BNK10] BARNUM P. C., NARASIMHAN S., KANADE T.: Analysis of rain and snow in frequency space. *International Journal of Computer Vision 86*, 2 (2010), 256–274. 17

[Bre05] BRENNER H.: Navier–stokes revisited. *Physica A: Statistical Mechanics and its Applications 349*, 1-2 (2005), 60–132. 2

[BRG*20] BOLIBAR J., RABATEL A., GOUTTEVIN I., GALIEZ C., CONDOM T., SAUQUET E.: Deep learning applied to glacier evolution modelling. *The Cryosphere 14*, 2 (2020), 565–584. 6

[BS10] BRALEY C., SANDU A.: Fluid simulation for computer graphics: A tutorial in grid based and particle based methods. *Virginia Tech, Blacksburg* (2010). 3

[BVF*15] BARTELT P., VALERO C. V., FEISTL T., CHRISTEN M., BÜHLER Y., BUSER O.: Modelling cohesion in snow avalanche flow. *Journal of Glaciology 61*, 229 (2015), 837–850. 5

[CDG*04] CORRIPIO J. G., DURAND Y., GUYOMARC'H G., MÉRINDOL L., LECORPS D., PUGLIÉSE P.: Land-based remote sensing of snow for the validation of a snow transport model. *Cold Regions Science and Technology 39*, 2-3 (2004), 93–104. 5

[CEG*18] CORDONNIER G., ECORMIER P., GALIN E., GAIN J., BENES B., CANI M.-P.: Interactive generation of time-evolving, snow-covered landscapes with avalanches. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 497–509. 12, 13, 16

[CJ05] CRESSERI S., JOMMI C.: Snow as an elastic viscoplastic bonded continuum: a modelling approach. *Italian Geotechnical J 4* (2005), 43–58. 4

[CKB10] CHRISTEN M., KOWALSKI J., BARTELT P.: Ramms: Numerical simulation of dense snow avalanches in three-dimensional terrain. *Cold Regions Science and Technology 63*, 1 (2010), 1–14. doi:10.1016/j.coldregions.2010.04.005. 5

[Cor04] CORRIPIO J. G.: Snow surface albedo estimation using terrestrial photography. *International Journal of Remote Sensing 25*, 24 (2004), 5705–5729. 4

[CPS*06] CAI J., PERVEZ M., SHEHATA M., JOHANNESSON R., BADAWY W., RADMANESH A.: On the identification of snow movements on roads. In *IEEE Workshop on Signal Processing Systems Design and Implementation* (2006), IEEE, pp. 357–361. 17

[CR15] CHANG J.-K., RYOO S.-T.: Real-time rendering of snow accumulation and melt under wind and light. *Int J Multimedia Ubiquitous Eng 10*, 12 (2015), 395–404. 14

[DARB11] DUTYKH D., ACARY-ROBERT C., BRESCH D.: Mathematical modeling of powder-snow avalanche flows. *Studies in Applied Mathematics 127*, 1 (2011), 38–66. 5

[DBS*14] DREIER L., BÜHLER Y., STEINKOGLER W., FEISTL T., BARTELT P.: Modelling small and frequent avalanches. In *Proceedings of the International Snow Science Workshop ISSW, Banff* (2014), vol. 29, pp. 135–138. 6

[DGP16] DAGENAIS F., GAGNON J., PAQUETTE E.: An efficient layered simulation workflow for snow imprints. *The Visual Computer 32*, 6 (2016), 881–890. 9, 10

[Dou67] DOUMANI G. A.: Surface structures in snow. *Physics of Snow and Ice: proceedings 1*, 2 (1967), 1119–1136. 4

[DPJP12] DELPARTE D., PETERSON M., JACKSON J., PERKINS J.: Modeling and visualizing avalanche flow using genetic algorithms and OpenGL. In *Proceedings, International Snow Science Workshop, Anchorage, Alaska* (2012). 11

[DPPJ13] DELPARTE D., PETERSON M. R., PERKINS J., JACKSON J.: Integrating gaming technology to map avalanche hazard. 5

[DZC*13] DING W., ZHU Z., CHEN X., ZHANG C., LIANG Y., LI S., FAN H., FENG H.: Real-time rain and snow rendering. In *2013 Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics)* (2013), IEEE, pp. 32–35. 16

[Eid09] EIDISSEN R.: *Utilizing GPUs for real-time visualization of snow*. Master's thesis, Institutt for datateknikk og informasjonsvitenskap, NTNU, 2009. 16

[EIR71] EGGLESTON K. O., ISRAELSEN E. K., RILEY J. P.: Hybrid computer simulation of the accumulation and melt processes in a snowpack. 4

[ERH06] ETIENNE J., RASTELLO M., HOPFINGER E. J.: Modelling and simulation of powder-snow avalanches. *Comptes Rendus Mécanique 334*, 8-9 (2006), 545–554. 5

[FA01] FUJISHIRO I., AOKI E.: Volume graphics modeling of ice thawing. In *Volume Graphics* (2001), Springer, pp. 69–80. 21

[FB07] FOLDES D., BENES B.: Occlusion-based snow accumulation simulation. In *VRIPHYS* (2007), pp. 35–41. 14

[FBB*12] FEISTL T., BEBI P., BÜHLER Y., CHRISTEN M., TEICH M., BARTELT P.: Stopping behavior of snow avalanches in forests. In *Proceedings of the International Snow Science Workshop ISSW. Anchorage, Alaska* (2012), pp. 420–426. 5

[FBT*14] FEISTL T., BEBI P., TEICH M., BÜHLER Y., CHRISTEN M., THURO K., BARTELT P.: Observations and modeling of the braking effect of forests on small and medium avalanches. *Journal of Glaciology 60*, 219 (2014), 124–138. 5

[Fea00] FEARING P.: Computer modelling of fallen snow. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), pp. 37–46. 14

[FG09] FESTENBERG N. V., GUMHOLD S.: A geometric algorithm for snow distribution in virtual scenes. In *Eurographics workshop on natural phenomena* (2009), The Eurographics Association, pp. 15–25. 14, 15

[FG11] FESTENBERG N. V., GUMHOLD S.: Diffusion-based snow cover generation. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1837–1849. 12, 13, 16

[FHHJ18] FANG Y., HU Y., HU S.-M., JIANG C.: A temporally adaptive material point method with regional time stepping. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 195–204. 7

[FKF*15] FISCHER J.-T., KOFLER A., FELLIN W., GRANIG M., KLEEMAYR K.: Multivariate parameter optimization for computational snow avalanche simulation. *Journal of Glaciology 61*, 229 (2015), 875–888. 5

[FM07] FUJISAWA M., MIURA K. T.: Animation of ice melting phenomenon based on thermodynamics with thermal radiation. In *Proceedings of the 5th international conference on computer graphics and interactive techniques in Australia and Southeast Asia* (2007), pp. 249–256. 20

[FO02] FELDMAN B. E., O'BRIEN J. F.: Modeling the accumulation of wind-driven snow. In *ACM SIGGRAPH conference abstracts and applications* (2002), pp. 218–218. 12

[FZ12] FAN N., ZHANG N.: Real-time simulation of rain and snow in virtual environment. In *International Conference on Industrial Control and Electronics Engineering* (2012), IEEE, pp. 29–32. 16

[Gau98] GAUER P.: Blowing and drifting snow in alpine terrain: numerical simulation and related field measurements. *Annals of Glaciology 26* (1998), 174–178. 4

[GC15] GUO Y., CURTIS J. S.: Discrete element method simulations for complex granular flows. *Annual Review of Fluid Mechanics 47* (2015), 21–46. 3

[GEF15] GOSWAMI P., ELIASSON A., FRANZÉN P.: Implicit Incompressible SPH on the GPU. In *12th Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS 2015), Lyon* (2015), Eurographics-European Association for Computer Graphics. 10

[GGT*18a] GAUME J., GAST T., TERAN J., VAN HERWIJNEN A., JIANG C.: Dynamic anticrack propagation in snow. *Nature communications 9*, 1 (2018), 1–10. 4

[GGT*18b] GAUME J., GAST T., TERAN J., VAN HERWIJNEN A., JIANG C.: Unified modeling of the release and flow of snow avalanches using MPM. In *Proceedings of the International Snow Science Workshop 2018* (2018), no. CONF. 5

[GGT*18c] GAUME J., GAST T., TERAN J., VAN HERWIJNEN A., JIANG C.: Unified modeling of the release and flow of snow avalanches using the material point method. In *Geophysical Research Abstracts* (2018), vol. 20, pp. 1–5. 5

[GHB*20] GISSLER C., HENNE A., BAND S., PEER A., TESCHNER M.: An implicit compressible SPH solver for snow simulation. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 36–1. 7, 8, 9, 10, 11, 22

[GI04] GAUER P., ISSLER D.: Possible erosion mechanisms in snow avalanches. *Annals of Glaciology 38* (2004), 384–392. 5

[GMH19] GOSWAMI P., MARKOWICZ C., HASSAN A.: Real-time particle-based snow simulation on the GPU. In *Eurographics Parallel Graphics and Visualization (EGPGV)* (2019), Eurographics-European Association for Computer Graphics. 8, 9, 10

[GNN22] GOSWAMI P., NORDIN A., NYLÉN S.: Iterative discrete element solver for efficient snow simulation. In *Eurographics Symposium on Parallel Graphics and Visualization (EGPGV)* (2022). 7, 9, 10, 11

[GP11] GAGNON J., PAQUETTE E.: Procedural and interactive icicle modeling. *The Visual Computer 27*, 6 (2011), 451–461. 19, 21

[GSSP10] GOSWAMI P., SCHLEGEL P., SOLENTHALER B., PAJAROLA R.: Interactive SPH simulation and rendering on the GPU. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA), Eurographics Association* (2010), p. 55–64. 10

[GvHG*19] GAUME J., VAN HERWIJNEN A., GAST T., TERAN J., JIANG C.: Investigating the release and flow of snow avalanches at the slope-scale using a unified model based on the material point method. *Cold Regions Science and Technology 168* (2019), 102847. 5

[Hag11] HAGENMULLER P.: *Experiments and simulation of the brittle failure of snow*. PhD thesis, Master's thesis, WSL Institute for Snow and Avalanche Research SLF, 2011. 4

[HAH02] HAGLUND H., ANDERSSON M., HAST A.: Snow accumulation in real-time. In *Special Effects and Rendering. Proceedings from SIGRAD 2002; Linköpings universitet; Norrköping; Sweden; November 28th and 29th; 2002* (2002), no. 007, Linköping University Electronic Press, pp. 11–15. 15

[Han21] HANÁK D.: *Real-time Snow Deformation*. Master's thesis, Department of Informatics, University of Masarykiana, 2021. 15

[HCN15] HAGENMULLER P., CHAMBON G., NAAIM M.: Microstructure-based modeling of snow mechanics: a discrete element approach. *The Cryosphere 9*, 5 (2015), 1969–1982. 4, 8

[HHP*21] HERRERA J. A. A., HÄDRICH T., PAŁUBICKI W., BANUTI D. T., PIRK S., MICHELS D. L.: Weatherscapes: nowcasting heat transfer and water continuity. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–19. 15, 16

[HLA*19] HU Y., LI T.-M., ANDERSON L., RAGAN-KELLEY J., DURAND F.: Taichi: a language for high-performance computation on spatially sparse data structures. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–16. 7

[HM09] HINKS T., MUSETH K.: Wind-driven snow buildup using a level set approach. In *Eurographics Ireland Workshop Series* (2009), vol. 9, pp. 19–26. 12, 13, 16

[ICS*14] IHMSEN M., CORNELIS J., SOLENTHALER B., HORVATH C., TESCHNER M.: Implicit incompressible SPH. *IEEE Transactions on Visualization and Computer Graphics 20*, 3 (2014), 426–435. doi:10.1109/TVCG.2013.105. 8, 19

[IDY*13] ISHIKAWA T., DOBASHI Y., YUE Y., KAKIMOTO M., WATANABE T., KONDO K., IWASAKI K., NISHITA T.: Visual simulation of glazed frost. In *ACM SIGGRAPH Posters*. 2013, pp. 1–1. 18, 20

[IKK*17] IM J., KIM J.-H., KIM W., PARK N., KIM T., KIM Y. B., LEE J., KIM C.-H.: Visual simulation of rapidly freezing water based on crystallization. *Computer Animation and Virtual Worlds 28*, 3-4 (2017), e1767. 19

[IPKK13] IM J., PARK H., KIM J.-H., KIM C.-H.: A particle-grid method for opaque ice formation. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 371–377. 21

[IUDN10] IWASAKI K., UCHIDA H., DOBASHI Y., NISHITA T.: Fast particle-based visual simulation of ice melting. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 2215–2223. 10, 20

[IYW*15] ISHIKAWA T., YUE Y., WATANABE T., IWASAKI K., DOBASHI Y., KAKIMOTO M., KONDO K., NISHITA T.: Visual simulation of glazed frost using hybrid heat calculation. *IIEEJ Transactions on Image Electronics and Visual Computing 3*, 2 (2015). 10, 18, 20

[JCK*22] JOUVET G., CORDONNIER G., KIM B., LÜTHI M., VIELI A., ASCHWANDEN A.: Deep learning speeds up ice flow modelling by several orders of magnitude. *Journal of Glaciology 68*, 270 (2022), 651–664. 6

[JMR17] JANSSEN C. F., MIERKE D., RUNG T.: On the development of an efficient numerical ice tank for the simulation of fluid-ship-rigid-ice interactions on graphics processing units. *Computers & Fluids 155* (2017), 22–32. 20

[JP20] JUNKER A., PALAMAS G.: Real-time interactive snow simulation using compute shaders in digital environments. In *International Conference on the Foundations of Digital Games* (2020), pp. 1–4. 15, 16

[JST*16] JIANG C., SCHROEDER C., TERAN J., STOMAKHIN A., SELLE A.: The material point method for simulating continuum materials. In *ACM SIGGRAPH Courses*. 2016, pp. 1–52. 6, 7

[KAL06] KIM T., ADALSTEINSSON D., LIN M. C.: Modeling ice dynamics as a thin-film Stefan problem. In *Symposium on Computer Animation: Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation: Vienna, Austria* (2006), vol. 2, pp. 167–176. 18, 19

[Kap03] KAPLER A.: Avalanche! snowy FX for XXX. In *ACM SIGGRAPH 2003 Sketches & Applications*. 2003, pp. 1–1. 11

[KBST22] KOSCHIER D., BENDER J., SOLENTHALER B., TESCHNER M.: A survey on SPH methods in computer graphics. In *Computer graphics forum* (2022), vol. 41, Wiley Online Library, pp. 737–760. 3

[KG93] KHARITONSKY D., GONCZAROWSKI J.: A physically based model for icicle growth. *The Visual Computer 10*, 2 (1993), 88–100. 19

[KHL04] KIM T., HENSON M., LIN M. C.: A hybrid algorithm for modeling ice formation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004), pp. 305–314. 18, 19

[KIKL16] KIM J.-H., IM J., KIM C.-H., LEE J.: Subtle features of ice with cloudy effects and scratches from collision damage. *Computer Animation and Virtual Worlds 27*, 3-4 (2016), 271–279. 21

[Kin67] KINOSITA S.: Compression of snow at constant speed. *Physics of Snow and Ice: proceedings 1*, 2 (1967), 911–927. 2, 3

[KL03] KIM T., LIN M. C.: Visual simulation of ice crystal growth. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), Citeseer, pp. 86–97. 18

[KL21] KIM J.-H., LEE J.: Stable and anisotropic freezing framework with interaction between IISPH fluids and ice particles. *IEEE Access 9* (2021), 146097–146109. 19, 20

[KLTB19] KUGELSTADT T., LONGVA A., THUEREY N., BENDER J.: Implicit density projection for volume conserving liquids. *IEEE Transactions on Visualization and Computer Graphics 27*, 4 (2019), 2385–2395. 3

[KMT67] KUROIWA D., MIZUNO Y., TAKEUCHI M.: Micromeritical properties of snow. *Physics of Snow and Ice: proceedings 1*, 2 (1967), 751–772. 4

[Kob93] KOBAYASHI R.: Modeling and numerical simulations of dendritic crystal growth. *Physica D: Nonlinear Phenomena 63*, 3-4 (1993), 410–423. 18

[Kro10] KROG Ø. E.: *GPU-based real-time snow avalanche simulations*. Master's thesis, Institutt for datateknikk og informasjonsvitenskap, NTNU, 2010. 11

[KRP19] KANG K.-K., RYOO D., PARK C.-J.: Snow accumulation effects in screen space for real-time terrain rendering. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)* (2019), IEEE, pp. 1024–1026. 15

[KS21] KOUTSIMANIS D., SOBIERAJ L.: *Prediction and Modelling of Snow Accumulation on Commercial Vehicles using CFD Simulations and Experimental Methods*. Master's thesis, Department of Mechanics and Maritime Sciences, Chalmers University of Technology, 2021. 4

[LCZ*21] LIU X., CHEN Y., ZHANG H., ZOU Y., WANG Z., PENG Q.: Physically based modeling and rendering of avalanches. *The Visual Computer 37*, 9 (2021), 2619–2629. 10, 11

[Lib16] LIBBRECHT K.: *A Field Guide to Snowflakes*. 2016. 16

[Lin16] LIND A.: *DEM Modeling of Snow-Wall Adhesion*. Master's thesis, Chemical Engineering, Chalmers University of Technology, 2016. 4

[LJHH18] LIU Y.-F., JAW D.-W., HUANG S.-C., HWANG J.-N.: Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing 27*, 6 (2018), 3064–3073. 17

[LLRR08] LEHNING M., LÖWE H., RYSER M., RADERSCHALL N.: Inhomogeneous precipitation distribution and snow transport in steep terrain. *Water Resources Research 44*, 7 (2008). 5

[LSJG21] LI X., SOVILLA B., JIANG C., GAUME J.: Three-dimensional and real-scale modeling of flow regimes in dense snow avalanches. *Landslides 18*, 10 (2021), 3393–3406. 5

[LW14] LII S.-Y., WONG S.-K.: Ice melting simulation with water flow handling. *The Visual Computer 30*, 5 (2014), 531–538. 10, 20

[LZ03] LANGER M., ZHANG Q.: Rendering falling snow using an inverse fourier transform. *ACM SIGGRAPH Technical Sketches program* (2003). 17

[LZK*04] LANGER M. S., ZHANG L., KLEIN A. W., BHATIA A., PEREIRA J., REKHI D.: A spectral-particle hybrid method for rendering falling snow. *Rendering Techniques 4* (2004), 217–226. 17

[MAMMH14] MAST C. M., ARDUINO P., MILLER G. R., MACKENZIE-HELNWEIN P.: Avalanche and landslide simulation using the material point method: flow dynamics and force interaction with structures. *Computational Geosciences 18*, 5 (2014), 817–830. 5

[MC95] MASSELOT A., CHOPARD B.: Cellular automata modeling of snow transport by wind. In *International Workshop on Applied Parallel Computing* (1995), Springer, pp. 429–435. 12

[MC00] MURAOKA K., CHIBA N.: Visual simulation of snowfall, snow cover and snowmelt. In *Proceedings Seventh International Conference on Parallel and Distributed Systems: Workshops* (2000), IEEE, pp. 187–194. 15

[McL12] MCLEAN D.: Continuum fluid mechanics and the Navier-Stokes equations. *Understanding Aerodynamics: Arguing from the Real Physics* (2012), 13–78. 2

[MEC17] MUKAI N., ETO Y., CHANG Y.: Representation method of snow splitting and sliding on a roof. In *5th International Conference on Advances in Engineering and Technology* (2017), pp. 100–103. 4

[Mel74] MELLOR M.: *A review of basic snow mechanics*. US Army Cold Regions Research and Engineering Laboratory Hanover, NH, 1974. 2

[Mel77] MELLOR M.: Engineering properties of snow. *Journal of Glaciology 19*, 81 (1977), 15–66. 2

[MGG*10] MARÉCHAL N., GUÉRIN E., GALIN E., MÉRILLOU S., MÉRILLOU N.: Heat transfer simulation for modeling realistic winter sceneries. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 449–458. 12, 13, 16

[Mic14] MICHAEL M.: *A Discrete Approach to Describe the Kinematics between Snow and a Tire Tread*. PhD thesis, University of Luxembourg, 2014. 4

[MMAL05] MOESLUND T. B., MADSEN C. B., AAGAARD M., LERCHE D.: Modeling falling and accumulating snow. In *VVG* (2005), Citeseer, pp. 61–68. 12, 16

[MNP15] MICHAEL M., NICOT F., PETERS B.: Advanced micromechanical description of snow behaviour—part i—mechanical and numerical modelling. *Int. J. Sol. Struct* (2015). 4

[MO08] MADRAZO C., OKADA M.: Physically based modeling of ice with bubbles. In *Eurographics (Short Papers)* (2008), pp. 13–16. 20

[Mon12] MONAGHAN J. J.: Smoothed particle hydrodynamics and its diverse applications. *Annual Review of Fluid Mechanics 44* (2012), 323–346. 3

[MPN15] MICHAEL M., PETERS B., NICOT F.: Advanced micromechanical description of snow behaviour part ii model validation. *International Journal of Solids and Structures, submitted* (2015). 4

[MS06] MCCLUNG D., SCHAERER P. A.: *The avalanche handbook*. The Mountaineers Books, 2006. 4

[MT05] MATSUMURA M., TSURUNO R.: Visual simulation of melting ice considering the natural convection. In *ACM SIGGRAPH Sketches*. 2005, pp. 61–es. 20

[MT10] MORIYA T., TAKAHASHI T.: A real time computer model for wind-driven fallen snow. In *ACM SIGGRAPH ASIA Sketches*. 2010, pp. 1–2. 14

[MTSK09] MADRAZO C., TSUCHIYA T., SAWANO H., KOYANAGI K.: Air bubbles in ice by simulating freezing phenomenon. *The Journal of the Society for Art and Science 8*, 1 (2009), 35–42. 20

[MVP15] MICHAEL M., VOGEL F., PETERS B.: DEM–FEM coupling simulations of the interactions between a tire tread and granular terrain. *Computer Methods in Applied Mechanics and Engineering 289* (2015), 227–248. 4

[MX15] MIAO Y., XIAO S.: Particle-based ice freezing simulation. In *Proceedings of the ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry* (2015), pp. 17–22. 10, 19

[NAS18] NEUKOM B., ARISONA S. M., SCHUBIGER S.: Real-time GIS-based snow cover approximation and rendering for large terrains. *Computers & Graphics 71* (2018), 14–22. 14, 16

[NIDN12] NISHINO T., IWASAKI K., DOBASHI Y., NISHITA T.: Visual simulation of freezing ice with air bubbles. In *SIGGRAPH Asia Technical Briefs*. 2012, pp. 1–4. 20, 21

[NVI04] NVIDIA: Snow accumulation. *Technical Report* (2004). 15

[OS04] OHLSSON P., SEIPEL S.: Real-time rendering of accumulated snow. In *The Annual SIGRAD Conference. Special Theme-Environmental Visualization* (2004), no. 013, Citeseer, pp. 25–31. 14

[Pet03] PETROVIC J.: Review mechanical properties of ice and snow. *Journal of Materials Science 38*, 1 (2003), 1–6. 2

[PGBT18] PEER A., GISSLER C., BAND S., TESCHNER M.: An implicit sph formulation for incompressible linearly elastic solids. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 135–148. 8

[PM76] PERLA R. I., MARTINELLI M.: *Avalanche handbook*. No. 489. US Department of Agriculture, Forest Service, 1976. 4

[PTL14] PEI S.-C., TSAI Y.-T., LEE C.-Y.: Removing rain and snow in a single image using saturation and visibility features. In *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)* (2014), IEEE, pp. 1–6. 17

[PTS99] PREMOŽE S., THOMPSON W. B., SHIRLEY P.: Geospecific rendering of alpine terrain. In *Eurographics Workshop on Rendering Techniques* (1999), Springer, pp. 107–118. 14

[Ras05] RASMUS S.: *Snow pack structure characteristics in Finland: Measurements and modelling*. Helsingin yliopisto, 2005. 2, 3

[RLD15] REYNOLDS D. T., LAYCOCK S. D., DAY A.: Real-time accumulation of occlusion-based snow. *The Visual Computer 31*, 5 (2015), 689–700. 15, 16

[RM13] RAJDERKAR D., MOHOD P.: Removing snow from an image via image decomposition. In *2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)* (2013), IEEE, pp. 576–579. 17

[RSDH09] REIWEGER I., SCHWEIZER J., DUAL J., HERRMANN H. J.: Modelling snow failure with a fibre bundle model. *Journal of Glaciology 55*, 194 (2009), 997–1002. 5

[SBJS03] SCHWEIZER J., BRUCE JAMIESON J., SCHNEEBELI M.: Snow avalanche formation. *Reviews of Geophysics 41*, 4 (2003). 5

[Sch16] SCHMID T. M.: *Real-Time Snow Simulation-Integrating Weather Data and Cloud Rendering*. Master's thesis, NTNU, 2016. 7

[SEN06] SALTVIK I., ELSTER A. C., NAGEL H. R.: Parallel methods for real-time visualization of snow. In *International Workshop on Applied Parallel Computing* (2006), Springer, pp. 218–227. 16

[Sig06] SIGRIST C.: *Measurement of fracture mechanical properties of snow and application to dry snow slab avalanche release*. PhD thesis, ETH Zurich, 2006. 4

[SOH99] SUMNER R. W., O'BRIEN J. F., HODGINS J. K.: Animating sand, mud, and snow. In *Computer Graphics Forum* (1999), vol. 18, Wiley Online Library, pp. 17–26. 9

[SSC*13] STOMAKHIN A., SCHROEDER C., CHAI L., TERAN J., SELLE A.: A material point method for snow simulation. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 1–10. 6, 7, 8, 9, 10, 11

[SSL14] STEINKOGLER W., SOVILLA B., LEHNING M.: Influence of snow cover properties on avalanche dynamics. *Cold Regions Science and Technology 97* (2014), 121–131. 5

[SSP*07] SULSKY D., SCHREYER H., PETERSON K., KWOK R., COON M.: Using the material-point method to model sea ice dynamics. *Journal of Geophysical Research: Oceans 112*, C2 (2007). 5

[ST20] SAI TANNERU Y.: *A DEM Study to investigate the influence of ice particle adhesion on the angle of repose*. Master's thesis, Department of Chemistry and Chemical Engineering, Chalmers University of Technology, 2020. 4

[Sta99] STAM J.: Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., p. 121–128. URL: `https://doi.org/10.1145/311535.311548`, `doi:10.1145/311535.311548`. 3

[Sta23] STAM J.: Stable fluids. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 779–786. 3

[Sto06] STOFFEL M.: *Numerical modelling of snow using finite elements*. No. 291. vdf Hochschulverlag AG, 2006. 4, 6

[Sve19] SVENSSON J.: *Real-time rendering of deformable snow covers*. Master's thesis, Department of Computer Science, Umeå University, 2019. 15

[TF11] TAN J., FAN X.: Particle system based snow simulating in real time. *Procedia Environmental Sciences 10* (2011), 1244–1249. 16

[TF12] TAKAHASHI T., FUJISHIRO I.: Particle-based simulation of snow trampling taking sintering effect into account. In *ACM SIGGRAPH Posters*. 2012, pp. 1–1. 8

[TFN14] TAKAHASHI T., FUJISHIRO I., NISHITA T.: Visual simulation of compressible snow with friction and cohesion. In *Proceedings of NICOGRAPH International* (2014), pp. 35–42. 7, 9, 10

[TL15] TIAN Z., LI B.: A simulation system of snow based on particle system. In *International Conference on Intelligent Systems Research and Mechatronics Engineering* (2015), Atlantis Press, pp. 740–743. 16

[Tok06] TOKOI K.: A shadow buffer technique for simulating snow-covered shapes. In *International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)* (2006), IEEE, pp. 310–316. 14

[TPS11] TECHEL F., PIELMEIER C., SCHNEEBELI M.: Microstructural resistance of snow following first wetting. *Cold Regions Science and Technology 65*, 3 (2011), 382–391. 5

[TYDN10] TSUDA Y., YUE Y., DOBASHI Y., NISHITA T.: Visual simulation of mixed-motion avalanches with interactions between snow layers. *The Visual Computer 26*, 6 (2010), 883–891. 10

[TZWZ09] TAN Y., ZHANG X., WANG C., ZHAO Q.: Real-time snowing simulation based on particle systems. In *First International Workshop on Education Technology and Computer Science* (2009), vol. 3, IEEE, pp. 7–11. 16

[VBVB19] VON BERNUTH A., VOLK G., BRINGMANN O.: Simulating photo-realistic snow and fog on existing images for enhanced CNN training and evaluation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (2019), IEEE, pp. 41–46. 17

[WF15] WONG S.-K., FU I.-T.: Hybrid-based snow simulation and snow rendering with shell textures. *Computer Animation and Virtual Worlds 26*, 3-4 (2015), 413–421. 9, 10

[WLCZ17] WANG Y., LIU S., CHEN C., ZENG B.: A hierarchical approach for rain or snow removing in a single color image. *IEEE Transactions on Image Processing 26*, 8 (2017), 3936–3950. 17

[WQS*20] WANG X., QIU Y., SLATTERY S. R., FANG Y., LI M., ZHU S.-C., ZHU Y., TANG M., MANOCHA D., JIANG C.: A massively parallel and scalable multi-gpu material point method. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 30–1. 7

[WW04] WANG N., WADE B.: Rendering falling rain and snow. In *ACM SIGGRAPH Sketches*. 2004, p. 14. 15, 17

[WWX*23] WANG D., WANG H., XU Y., ZHOU J., SUI X.: Numerical simulations of the driving process of a wheeled machine tire on a snow-covered road. *Machines 11*, 6 (2023), 657. 8

[WWXP06] WANG C., WANG Z., XIA T., PENG Q.: Real-time snowing simulation. *The Visual Computer 22*, 5 (2006), 315–323. 16

[XZLT12a] XU J., ZHAO W., LIU P., TANG X.: An improved guidance image based method to remove rain and snow in a single image. *Computer and Information Science 5*, 3 (2012), 49. 17

[XZLT12b] XU J., ZHAO W., LIU P., TANG X.: Removing rain and snow in a single image using guided filter. In *IEEE International Conference on Computer Science and Automation Engineering (CSAE)* (2012), vol. 2, IEEE, pp. 304–307. 17

[YSHW03] YANYUN C., SUN H., HUI L., WU E.: Modelling and rendering of snowy natural scenery using multi-mapping techniques. *The Journal of Visualization and Computer Animation 14*, 1 (2003), 21–30. 14

[YZMC08] YANG Y., ZHU X., MEI J., CHEN D.: Design and real-time simulation of rain and snow based on lod and fuzzy motion. In *Third International Conference on Pervasive Computing and Applications* (2008), vol. 1, IEEE, pp. 510–513. 16

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Transactions on Graphics (TOG) 24*, 3 (2005), 965–972. 7, 18

[ZCL10] ZHANG J., CAI X., LI J.: Rendering snowing scene on GPU. In *IEEE International Conference on Intelligent Computing and Intelligent Systems* (2010), vol. 3, IEEE, pp. 199–202. 16

[ZLG*13] ZHENG X., LIAO Y., GUO W., FU X., DING X.: Single-image-based rain and snow removal using multi-guided filter. In *International Conference on Neural Information Processing* (2013), Springer, pp. 258–265. 17

[ZXZ10] ZOU C., XIE X., ZHAO G.: Algorithm for generating snow based on GPU. In *Proceedings of the Second International Conference on Internet Multimedia Computing and Service* (2010), pp. 199–202. 16

[ZZL11] ZHANG Y., ZOU L., LIU J.: Simulation of snow effects in visual simulation of virtual campus based on OSG. In *International Conference on Multimedia Technology* (2011), IEEE, pp. 3658–3662. 16