

# Heat Transfer Simulation for Modeling Realistic Winter Sceneries

N. Maréchal, E. Guérin  
LIRIS - CNRS  
University Claude Bernard Lyon 1, France

E. Galin  
LIRIS - CNRS  
University Lumière Lyon 2, France

S. Mérillou, N. Mérillou  
XLIM - CNRS  
University of Limoges, France

---

## Abstract

*This paper presents a physically based method for simulating the heat transfers between the different environmental elements to synthesize realistic winter sceneries. We simulate the snow fall over the ground, as well as the conductive, convective and radiative thermal transfers using a finite volume method according to the variations of air and dew point temperatures, the amount of snow, cloud cover and day-night cycles. Our approach takes into account phase changes such as snow melting into water or water freezing into ice.*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—

**Keywords:** Snow, ice, winter landscapes, thermal transfers, natural phenomena.

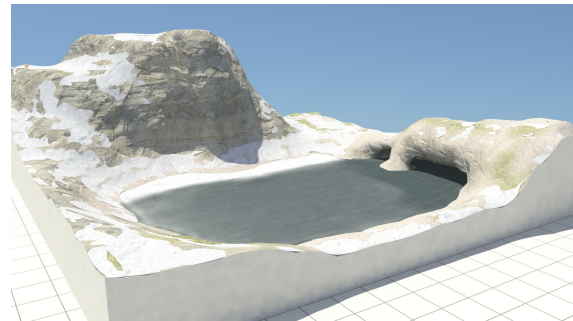
---

## 1. Introduction

Modeling natural and realistic sceneries is an important topic in computer graphics. Depending on their geographical location and the time in the year, landscapes may have completely different visual aspects. In particular, snow is a conspicuous natural phenomenon that plays an important part in the change of appearance of landscapes. During winter, several important phenomena occur, from the evolution of snow layers to the freezing of lakes, at varying speeds according to the weather conditions.

In this paper, we focus on generating and simulating the evolution of landscapes in winter according to the weather. A vast variety of procedural and physically based techniques have been proposed for modeling snow fall and its accumulation according to the geometry of objects [Fea00, HAH02, FB07] and the direction and intensity of the wind [SEN08, WWXP06]. Several methods were also proposed for simulating ice crystals formation [KL03, KHL04] and stalactites growth [KAL06].

While those approaches generate beautiful images, they do not take into account the thermal transfers between the different environmental elements in the scene. Therefore, they cannot capture many important natural phenomena such as snow melting in the sun, or lakes freezing during long low temperature periods.



**Figure 1:** A partially frozen high altitude lake in the mountains obtained with our method.

Simulating the change of appearance of a natural landscape covered by snow according to the weather and the characteristics of the environment is a very challenging problem. At the beginning of the snow fall, the temperature of the ground is often too warm to allow the accumulation of snow. Snow starts piling only when the temperature of the ground gets cold enough. Snow exposed to sun rays melts quicker than snow in the shade of trees or mountains. Therefore, there is a need to simulate thermal transfers to automatically generate realistic winter sceneries and simulate their

evolution throughout time. To the best of our knowledge, our method is the first to propose as many heat transfers to simulate realistic winter landscapes.

In this paper, we propose a framework for generating realistic winter sceneries by taking into account the thermal transfers between elements. Our approach combines a physically based thermal simulation driven by an environmental model with a procedural instantiation technique for creating the surface and texture details of ice and snow from the physical data. More precisely, our contributions are as follows:

- We define an environmental model consisting of a set of functions that describe the air and dew point temperatures, precipitations, day-night cycles and cloud cover.
- We describe a finite volume thermal simulation that takes into account conductive, convective and radiative heat transfers to simulate the variation of temperature of the different materials in the scene.
- We propose a simple technique for simulating phase changes between solid and liquid states and model freezing and melting processes.
- We present a simple and efficient technique to generate the geometric representation of the snow and ice layers in the scene.

The remainder of this paper is organized as follows. Section 2 presents an overview of related work on snow fall and ice growing simulations. Section 3 presents an overview of our method, recalls thermal fundamentals and gives notations. Section 4 describes the weather model controlling our simulation, the heat transfer simulation and presents the boundary limit condition, phase transition and natural convection. Section 5 presents procedural techniques for generating the textured snow and ice meshes from the previous data. Section 6 presents our results and discusses limitations. Finally, section 7 presents our conclusion and future work.

## 2. Related work

Most existing techniques for modeling snow accumulation can be classified in two categories: particle based and surface displacement snow generation techniques. Several techniques were also proposed for simulating ice crystal formation and stalactites growth.

**Particle based snow accumulation methods** Those methods aim at computing the distribution of fallen snow by evaluating the trajectory of snow flakes blown by the wind and piling onto the ground. A first model for creating fallen snow was presented in [NIDN97] based on metaballs and a user specified particle distribution. Fearing [Fea00] proposed to compute the distribution of fallen snow onto the ground by tracking stochastically generated snow particles shot upwards to the clouds performing recursive snow surface stability tests. Several improvements of this method were proposed by improving the transport of snow flakes by the wind.

Those simulations can capture some of the complex snow layer features obtained by the dynamics of the wind solving either the Navier-Stokes [FO02] or the Boltzmann equations [WWXP06]. Recently, parallel implementations to resolve Navier-Stokes equations were described in [SEN08].

To our knowledge, the only physically based simulation method involving thermal transfers was presented in [MC00]. In this approach, the snow distribution is computed by simulating the snow fall using vortex fields. Snow melting is simulated by resolving heat conduction. In contrast, in our method we take into account all thermal transfers and propose a weather model to drive the simulation.

**Surface displacement methods** Contrary to particle based snow accumulation methods, surface displacement approaches [PTS99] characterize the height of accumulated snow using local accessibility and occlusion evaluations. Several accelerated implementations have been proposed for large scale terrains. The snow accumulation regions are defined by calculating ambient occlusion and by dissipating it with illumination [FB07] or using the depth buffer to quantify the snow that a surface can receive [OS04]. Hybrid multi-mapping methods were proposed to generate winter sceneries [CSLW03] by using a displacement map to model snowy blocks on near objects and a volumetric texture map for distant objects.

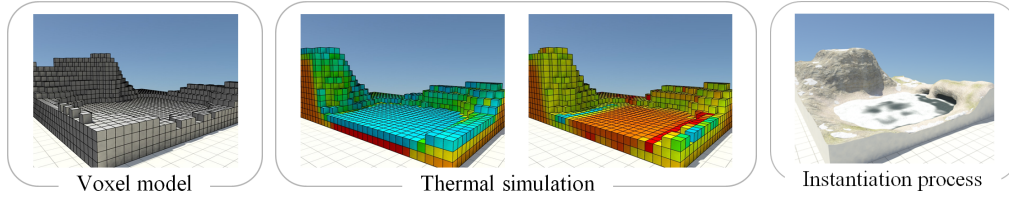
Height span maps, originally proposed for the interactive animation of granular materials [SOH99] have demonstrated to be an efficient model to model snow covering objects. A real-time snow accumulation method based on very simple height span map distribution scheme has been proposed in [HAH02]. Significant improvements were proposed in [FG09] by developing a phenomenologically inspired statistical model for snow accumulation, derived from real world observations.

**Ice growth** Ice stalactites and ice crystal growth is a natural phenomenon involving complex phase transition and solidification processes. [KL03] first proposed a two-dimensional technique using a phase field method for simulating ice crystals formation over objects. This method was improved in [KHL04] by using a hybrid approach combining a procedural diffuse limited aggregation, a phase field method and a fluid simulation. The same authors proposed a three-dimensional method for simulating the formation of ice stalactites by solving the Stefan problem [KAL06].

## 3. Overview and notations

In this section, we present our architecture for simulating thermal transfers and generating snow and ice.

**Overview** Modeling and simulating the evolution of snow and ice layers in natural sceneries is a very challenging problem. Capturing the small geometric details in a large scene

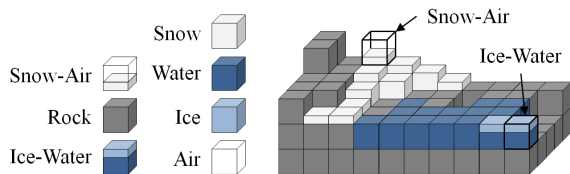


**Figure 2:** Synthetic overview of our winter scenery generation process.

using a fine voxel grid would require memory exhausting and computationally demanding simulations. Therefore, we use a coarse voxel grid for running our physically based thermal simulation driven by an environmental model, combined with a procedural instantiation technique for creating the surface and texture details of ice and snow from the computed physical data (Figure 2).

Starting from an initial input terrain model, we first generate a voxel representation of the terrain. Then we perform a thermal simulation to evaluate the evolution of the temperature of every voxel according to the weather which defines the variations of air temperature as well as the amount of snow fall throughout time. Our simulation method relies on a finite volume technique that computes the conduction, convection and radiation transfers between the elements in the scene.

The instantiation process consists in generating a textured mesh representation of snow and ice from the information carried in the voxel grid. The surface of the snow is generated as a heightfield from the snow heights stored in the voxels [HAH02, FG09], whereas ice and melted water-ice materials are generated using a specific surface displacement technique. Finally, the surface is perturbed using procedural noise to produce a more realistic and natural appearance.

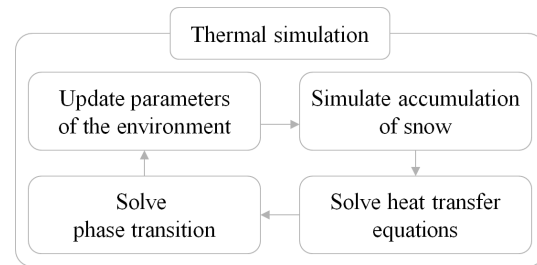


**Figure 3:** Voxel based representation of the scene with one and two-materials voxels.

**Data structure** The scene is represented by using a decomposition into a grid of voxels, denoted as  $\mathcal{V}_{ijk}$ . A voxel stores an identifier characterizing its material (Figure 3) and some information on the thermal properties of the voxel including the amount of energy exchange during a timestep, its temperature and a percentage of solid material for phase change. In our implementation, voxels can contain rock, snow, ice, water or air.

Some voxels can contain two different materials (Figure 3) and will be referred to as two-material voxels. Those particular voxels are needed to simulate phase transitions, such as ice-water voxels that are involved in the process of simulating ice melting into water. The quantity of each material is indicated by a percentage of the voxel volume.

In our implementation, only snow-air and water-ice voxels are concerned, which allows knowledge of the exact quantity of snow or ice during phase changes or during the snow fall.



**Figure 4:** Overview of the simulation loop

**Overview of the simulation** The overall simulation is performed in four steps (Figure 4):

1. Compute the parameters of the environment using the weather model.
2. Simulate snow fall and accumulation to the ground according to the characteristics of the environment which are driven by the weather conditions.
3. Compute the thermal transfers between the voxels  $\mathcal{V}_{ijk}$  in the scene by simulating the heat flows of different types through the faces of the voxels, and compute the variation of temperature of the voxels from the energy  $dQ$  and the characteristics of its material.
4. Compute the phase change for every voxel  $\mathcal{V}_{ijk}$  and update the temperature of water voxels by simulating natural convection.

The accumulation of snow onto the ground is performed as prescribed in [FB07]. Note that we although did not simulate the transport of snow by wind, our method is compatible with the technique described in [WWXP06].

|                                 |                   |  |
|---------------------------------|-------------------|--|
| Specific heat capacity          | $C_p$             | $\text{Jkg}^{-1}\text{K}^{-1}$             |
| Volumetric mass density         | $\rho$            | $\text{kgm}^{-3}$                          |
| Conductivity                    | $\lambda$         | $\text{Js}^{-1}\text{m}^{-1}\text{K}^{-1}$ |
| Convective transfer coefficient | $h_{\text{conv}}$ | $\text{Js}^{-1}\text{m}^{-2}\text{K}^{-1}$ |
| Melting temperature             | $T_g$             | K  |
| Latent heat fusion              | $L_f$             | $\text{Jkg}^{-1}$                          |

**Table 1:** Notations and units for some physical quantities

The next section presents the details of the evaluation of the heat transfers as well as the phase change computation.

#### 4. Physical simulation

In order to have a uniform representation of all types of heat transfers, we model transfers as heat flows between the contact surface of our voxels using a finite volume method. The heat flow  $\phi$  ( $\text{Js}^{-1}$ ) is equal to  $\phi = S\varphi$  where  $S$  ( $\text{m}^2$ ) denotes the area of the transfer surface and  $\varphi$  refers to the density of the heat flow ( $\text{Js}^{-1}\text{m}^{-2}$ ). After each iteration, the heat variation  $dQ$  (J) is obtained by  $dQ = \phi dt$  where  $dt$  is the time step.

Let  $V$  denote an elementary volume element,  $\rho$  the volumetric mass density of the material and  $C_p$  the specific heat capacity at constant pressure. The conversion between heat variation  $dQ$  and the temperature variation  $dT$  is obtained by the equation  $dQ = \rho C_p V dT$ . To solve the inhomogeneous heat equation, we compute the different heat flows between the elements in the scene, using the corresponding material coefficients. The variation of temperature of a given voxel is defined as:

$$dT = \sum_i \frac{S\varphi_i}{\rho C_p V} dt$$

##### 4.1. Environmental model

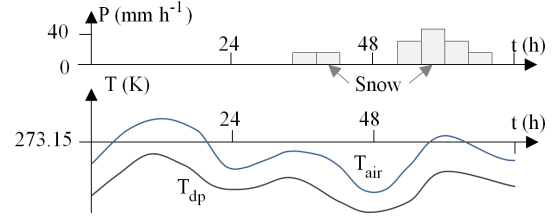
Our environmental model defines the evolution of the weather throughout time. The weather is characterized by the air temperature  $T_{\text{air}}(t)$ , the dew point temperature  $T_{\text{dp}}(t)$ , the amount of snow fall  $P(t)$  ( $\text{mmh}^{-1}$ ), the cloud cover  $C(t)$  and the day and night cycles  $D(t)$ .

The functions  $T_{\text{air}}(t)$ ,  $T_{\text{dp}}(t)$ ,  $P(t)$  and  $C(t)$  are either obtained from weather data or controlled by the user. The cloud cover  $C(t)$  ranges within the unit interval  $[0, 1]$ , with 0 denoting a clear sky and 1 a dense cloud cover. The day and night function  $D(t)$  is procedurally defined as a function of the geographical location of the scene (characterized by the latitude and longitude and the time in the year).

Figure 5 illustrates the functions representing the air and the dew point temperature  $T_{\text{air}}$  and  $T_{\text{dp}}$  respectively as well

| Material | $C_p$ | $\rho$ | $\lambda$ | $\epsilon$ | $a$  |
|----------|-------|--------|-----------|------------|------|
| Rock     | 1600  | 2300   | 2.1       | 0.45       | 0.3  |
| Water    | 4186  | 999    | 0.602     | 0.95       | 0.05 |
| Snow     | 2090  | 110    | 0.05      | 0.82       | 0.9  |
| Ice      | 2050  | 915    | 2.22      | 0.97       | 0.8  |
| Air      | 1006  | 1.188  | 0.026     | 1          | 0    |

**Table 2:** Physical coefficients for different materials



**Figure 5:** Synthetic representation of the parameters of our environmental model

as the amount of snow fall  $P$ . Snow fall occurs only when the air temperature is less than 274K.

##### 4.2. Thermal transfer simulation

**Physical properties of materials** Every material in the scene is characterized by a set of constants defining its physical properties: the specific heat capacity  $C_p$ , the volumetric mass density  $\rho$ , the thermal conductivity  $\lambda$ , the emissivity  $\epsilon$ , the melting temperature  $T_g$ , the fusion latent heat  $L_f$  and the albedo  $a$ . Fluids like air and water are also characterized by their convective transfer coefficient denoted as  $h_{\text{conv}}$ . Table 1 recalls some of the physical properties of materials, the corresponding notations and units. Note that the albedo  $a$  and the emissivity  $\epsilon$  are dimensionless coefficients.

Table 2 shows the values of physical properties we used for our thermal simulation. The melting temperature of water, ice and snow is defined as  $T_g = 273.15$  K and the fusion latent heat as  $L_f = 333.10^3 \text{Jkg}^{-1}$ .

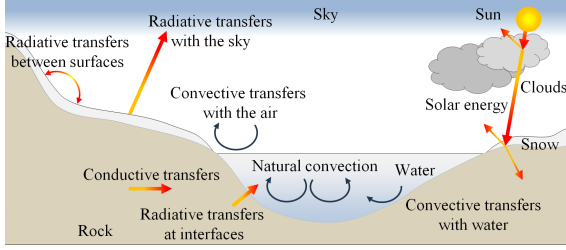
**Thermal transfers** A thermal transfer appears in a medium and at the interface between two media when there is a temperature gradient  $\nabla T \neq 0$ . This transfer comes from the warmest region to the coolest one. There are three types of thermal transfers:

- heat transfers by **conduction** that take place inside a medium or between two media in contact. There is no material displacement composing the medium, the heat is diffused gradually by molecular agitation;
- heat transfers by **convection** that take place in fluid materials and at the interface of a fluid medium and a solid

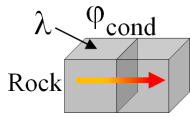
medium. Unlike conduction, convection is produced by material displacement. Heat is carried by the fluid move;

- heat transfers by **radiation** that are emitted as electromagnetic rays from the surface of any material whose temperature is greater than 0 K.

Figure 6 presents an overview of the thermal transfers that are taken into account in our simulation. The following paragraphs detail the computation of the different thermal transfers.

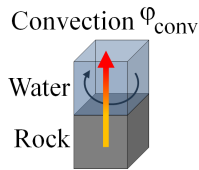


**Figure 6:** Thermal transfers that are taken into account in our simulation.



**Conductive heat transfer** The conduction law has been introduced by Joseph Fourier and states that the density of heat flow  $\phi_{\text{cond}}$  ( $\text{Js}^{-1}\text{m}^{-2}$ ) is proportional to the gradient of temperature  $\nabla T$ . Let  $\lambda$  denote the thermal conductivity of the material, we have:

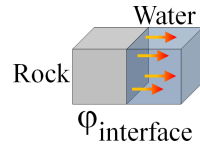
$$\phi_{\text{cond}} = -\lambda \nabla T$$



**Convective transfers at fluid interfaces** Let  $h_{\text{conv}}$  denote the convective exchange coefficient,  $T_{\text{water}}$  the temperature of the fluid and  $T_{\text{rock}}$  the temperature of the surface of the material. At the interface between a fluid and another material, the heat flow follows the Newton law:

$$\phi_{\text{conv}} = h_{\text{conv}}(T_{\text{water}} - T_{\text{rock}})$$

This law models the heat flow between the fluid and the surface of the material according to the fluid agitation. The degree of fluid agitation is expressed by the convective exchange coefficient  $h_{\text{conv}}$ . The higher this coefficient is, the more agitated the fluid is. The coefficient for natural air convection is generally comprised within  $[5, 10] \text{Js}^{-1}\text{m}^{-2}\text{K}^{-1}$ . The corresponding natural water convection coefficient is much larger and can be set within  $[100, 900] \text{Js}^{-1}\text{m}^{-2}\text{K}^{-1}$ . In our simulations, those convective exchange coefficients have been set to 5 and 100 respectively, which represents still air and water.



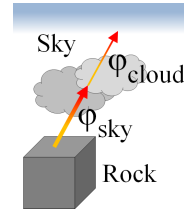
**Radiative heat transfer at interfaces**

Most of the materials present in nature are considered as grey bodies. Their electro-magnetic emissivity is isotropic. This hypothesis allows the direct computation of the heat flow emitted from the surface of a material whose temperature is known. Let  $\epsilon$  denote the emissivity of the material,  $\sigma$  the Stefan-Boltzmann constant ( $\text{Js}^{-1}\text{m}^{-2}\text{K}^{-4}$ ) and  $T_{\text{rock}}$  the temperature of the material at the surface respectively. The Stefan-Boltzmann relation may be written as follows:

$$\phi_{\text{rad}} = \epsilon \sigma T_{\text{rock}}^4$$

At the interface of two materials perfectly in contact, the radiative flow is calculated by taking into account both radiative transfers. Let  $T_{\text{water}}$  denote the temperature of a water neighboring voxel:

$$\phi_{\text{interface}} = \epsilon \sigma (T_{\text{rock}}^4 - T_{\text{water}}^4)$$



**Radiative transfer with the sky**

Let  $\epsilon_{\text{sky}}$  denote the emissivity of the sky,  $\sigma$  the previously defined Stefan-Boltzmann constant,  $T_{\text{sky}}$  the temperature of the sky and  $T_{\text{rock}}$  the temperature of the rock voxel. The heat flow between a surface and the sky is defined as:

$$\phi_{\text{sky}} = \epsilon_{\text{sky}} \sigma (T_{\text{sky}}^4 - T_{\text{rock}}^4)$$

Recall that the emissivity of a material is the relative power of its surface to emit heat by radiation. It is the ratio of energy radiated by a particular material to energy radiated by a black body at the same temperature. It is a measure of a material's ability to radiate absorbed energy. A commonly used coarse approximation consists in considering the sky as a black body, thus we can simplify the previous equation by setting its emissivity to  $\epsilon_{\text{sky}} = 1$ .

The temperature of the sky  $T_{\text{sky}}(t)$  is needed to calculate the radiative heat transfer between the surface and the sky. The sky temperature depends on the temperature of the air  $T_{\text{air}}(t)$  and the dew point temperature  $T_{\text{dp}}(t)$  for a cloudless sky [LILV08]. Let  $h(t) \in [0, 24]$  denote the time of the day beginning at midnight, we have:

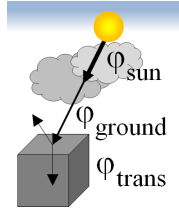
$$T_{\text{sky}}(t) = T_{\text{air}}(t) (0.711 + 0.0056 T_{\text{dp}}(t) + 7.3 \times 10^{-5} T_{\text{dp}}(t)^2 + 0.013 \cos(2\pi h(t)/24))^{1/4}$$

This equation simulates the decrease of temperature of the sky  $T_{\text{sky}}$  as the amount of watervapor in the air decreases with the dew point temperature  $T_{\text{dp}}$ .

When clouds are present in atmosphere, the value of  $\phi_{\text{sky}}$  decreases. To take this phenomenon into account, we weight the heat flow by the clearness index denoted as  $I_c(t)$  and whose range lies between 0.3 and 0.8 [DB91]. Recall that

in our weather model,  $C(t) \in [0, 1]$  denotes the cloud cover. Thus, we define  $I_c(t) = 0.8 - 0.5C(t)$ . The flow between the surface and the cloudy sky is finally defined as:

$$\Phi_{\text{cloud}} = I_c \Phi_{\text{sky}}$$



**Solar energy** The solar energy represents the direct contribution of sun rays to the surface of the objects. To evaluate the contribution of the solar energy onto the surface of objects, we need to calculate the extraterrestrial radiance denoted as  $\phi_{\text{sun}}$  ( $\text{J s}^{-1} \text{m}^{-2}$ ).

Let  $n$  denote the day of year and  $\theta$  the incidence angle of solar rays on the surface. The solar irradiance before atmospheric filtering is given by [DB91]:

$$\phi_{\text{sun}} = 1367 (1 + 0.033 \cos(2\pi n/365)) \cos(\theta)$$

$\cos(\theta)$  is calculated from the latitude, declination, azimuth, tilt of the surface and hour angle of the sun.

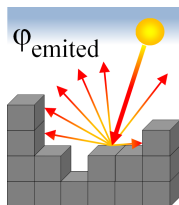
The solar energy is filtered by both the atmosphere and clouds however. Therefore, we weight the heat flow density  $\phi_{\text{sun}}$  by the clearness index  $I_c$  to compute the energy received by the surface of the ground:

$$\Phi_{\text{ground}} = I_c \Phi_{\text{sun}}$$

A part of this energy is also reflected by the surface. The energy transmitted to the ground  $\Phi_{\text{trans}}$  is evaluated from the albedo  $a \in [0, 1]$ :

$$\Phi_{\text{trans}} = (1 - a) \Phi_{\text{ground}}$$

The albedo is defined as the ratio between solar energy reflected and incident solar energy (Table 2 recalls the albedo values used in our system for the different materials). In our implementation, the computation of this energy is done only during daytime, when  $D(t) = 1$ . For each voxel, we trace a ray in the direction of the sun. If this ray does not intersect the terrain, then the voxel receives solar energy.



**Radiative transfers between surfaces** This is a visible phenomenon that plays an important role in the mountains: the sun is reflected by the sun-facing slopes of the valley, melting snow on the bottom of the opposite-facing slopes.

Radiative transfers between surfaces are considered as follows. Instead of simulating the overall complete radiosity in the scene, we only consider the reflection of heat on the surfaces of the scene, thus giving the first bounce of heat. The other successive bounces are considered as negligible.

Recall that  $\Phi_{\text{ground}}$  denotes the heat flow density from the sun reaching the ground. The amount of energy reflected by the ground is denoted as:

$$\Phi_{\text{reflected}} = a \Phi_{\text{ground}}$$

Moreover, every voxel at the contact with air also emits some radiative energy, denoted as  $\Phi_{\text{interface}}$ . Therefore, the total emitted energy is:

$$\Phi_{\text{emitted}} = \Phi_{\text{reflected}} + \Phi_{\text{interface}}$$

In our implementation, we simulate radiosity by emitting rays from surface of the free side patches of the voxels in contact with air.

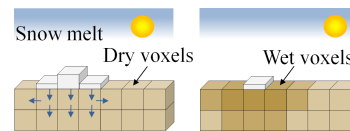
### 4.3. Phase changes

A material can have three different states in our system: solid, liquid and gas. Phase change corresponds to the transition of a material from one state to another. Transition from solid to liquid or from liquid to solid requires the material to be at fusion temperature  $T_g$ . Let us consider melting from solid to liquid state. When the fusion temperature is reached, energy injected to the material does not increase the temperature any more. This energy is absorbed into a thermal hole  $Q_{\text{hole}}$ . When this energy has reached the fusion latent heat  $L_f$  for the considered material volume, the material becomes completely liquid.  $L_f$  also depends on the nature of the material.

When the material has reached the fusion temperature and energy absorbed by the thermal hole  $Q_{\text{hole}}$  increases, the material is considered as having two states simultaneously. The percentage of solid material can be evaluated as follows:

$$r = \frac{Q_{\text{hole}}}{L_f \rho V}$$

In our system, water, snow and ice are the three materials that can enter in phase change. The melting of snow or ice into water is a complex phenomenon involving thermal phenomena and fluid dynamics. When ice or snow melts, it is converted into water which can be partially or totally absorbed by the ground (thus changing the characteristics of the ground). Water may also flow at the surface of the ground and generate water puddles or accumulate in rivers or lakes. Flowing water may even freeze, depending on its flowing speed and the temperature of the ground and the air.



**Figure 7:** Snow melts into water and is absorbed by the ground; the thermal characteristics of the ground voxels are changed.

In our implementation, we make the following approximations and do not simulate the dynamics of flowing water. Instead, snow or ice melting into water is absorbed by the ground or existing large water bodies such as lakes or rivers. When water is absorbed by large water bodies, we modify

the temperature of the corresponding water voxels. We simulate water absorbed by ground voxels as follows.

Ground voxels are characterized by a wetness coefficient denoted as  $w$  that characterizes the amount of water diffusing inside the pores and cracks of the ground. We approximate the porosity of the ground and the capillary action by diffusing absorbed water in the neighboring ground voxels (Figure 7). In contrast, we do not simulate the evaporation of water from the ground which is a very complex phenomenon.

#### 4.4. Boundary conditions

In transient regime, the amplitude of temperature oscillations is damped as the depth into the ground increases. Therefore, we need to evaluate at which depth the temperature of the ground is almost constant.

A good approximation consists in analysing the variations of the temperature in the ground as a function of an input periodic variation of the air temperature [CJ59]. Let us consider an infinite plane separating the ground and the air. We assume that the variation of temperature of the air is defined as  $\delta T_{\text{air}}(t) = T_0 \cos(\omega t)$ . Let  $\beta = \lambda/\rho C_p$  denote the diffusivity of the ground. The variation of the temperature of the ground can be written as:

$$\delta T_{\text{ground}}(z, t) = T_0 e^{-\sqrt{\frac{\omega}{2\beta}} z} \cos\left(\omega t - \sqrt{\frac{\omega}{2\beta}} z\right)$$

Given an input maximum variation of temperature  $T_e$ , the depth from which the temperature of the ground can be considered as a constant and can be written as:

$$z = -\ln\left(\frac{T_e}{T_0}\right) \sqrt{\frac{2\beta}{\omega}}$$

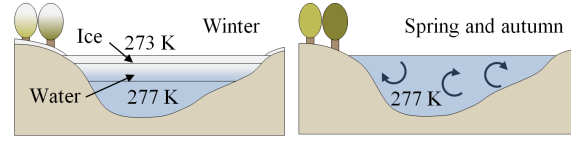
Numerical results show that the temperature is almost constant at  $z = 12\text{m}$ . Thus, in our implementation, we do not simulate thermal transfers down to this depth. Consequently, we chose Neumann conditions at boundary of our domain by setting the flow to zero. We also apply Neumann conditions on the four vertical boundaries of the domain. The prescribed temperature of the air at the top boundary corresponds to a Dirichlet condition.

#### 4.5. Natural convection

All direct solar radiation enters water bodies through the air-water interface. A significant amount of light is reflected from the water surface, especially where the sun's rays strike the surface at a narrow angle. Of the light that enters water bodies, some is scattered upwards. Absorption of solar energy results in the generation of heat.

When water is warmed by absorption of solar radiation, there is a pronounced change in temperature at the bottom of the photic zone. The density of water is varying according to the temperature. Above 277 K, warm water is lighter

than cold one. In summer, the change in density of the water ensures that warm water remains at the surface where it becomes further warmed, effectively dividing the water body horizontally which becomes divided by a thermocline. Stratification is maintained unless water is circulated by winds or currents.



**Figure 8:** Natural convection in spring and summer, and water stratification in winter.

Below 277 K however, the density curve of water is inverted: cold water becomes lighter than warm one. Therefore, in autumn and spring, warm and cold water enter into a convective motion and homogenise. In winter, when the air temperature is lower than water temperature, there is no convection (Figure 8).

In our implementation, we approximate this phenomenon by averaging the temperature of all the connected water voxels with a temperature above 277 K. Below 277 K, we stop this homogenisation and calculate conductive heat transfers, which makes appear a thermal gradient into water.

Air is also subject to convection. In our implementation, we consider that the temperature of air is constant and do not consider the influence of the wind.

#### 4.6. Implementation details

For every voxel  $\mathcal{V}_{ijk}$  in the scene, we store an identifier corresponding to the material type (which enables us to retrieve the thermal constants of the material), the temperature  $T$  and the total amount of energy  $\delta Q^n$  representing the sum of all the heat flows at every iteration  $n$  of the simulation. Two material voxels, such as water-ice or snow-air voxels, also store the relative amount of material within the voxel which are needed in the computation of phase changes.

The spatial gradient of temperature between two neighboring voxels is simply computed as  $\delta T/\delta x$  where  $\delta x$  denotes the spatial resolution of the voxel and  $\delta T$  the difference of temperature. The overall simulation is performed using a forward Euler integration scheme. Let  $\delta t$  denote the integration time step. At every step  $n$  and for every voxel  $\mathcal{V}_{ijk}$ , we compute the sum of all the heat flows  $\delta Q^n$  and compute the new temperature as:

$$T^{n+1} = T^n + \frac{\delta Q^n}{\rho C_p V} \delta t$$

The landscape simulations presented in this paper were performed using a spatial resolution of 1 m, whereas the technical examples demonstrating the impact of the shade of

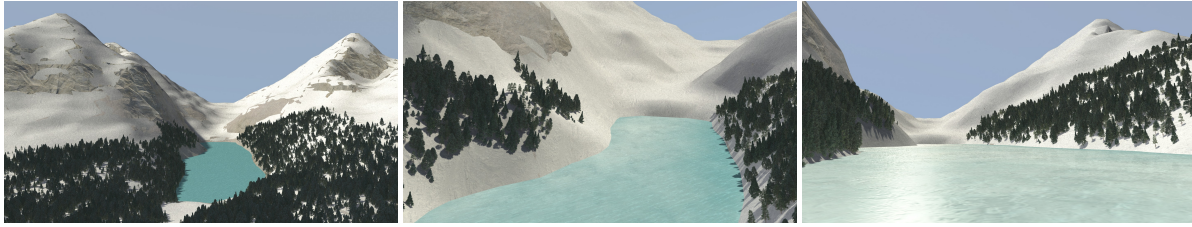


Figure 9: The lake Louise in winter.

the sun and the thermal inertia of water were conducted on smaller scenes with a spatial resolution of 10cm. Note that all the images showing the temperature of the voxels show the voxel structure at a coarser resolution.

The time step for the simulations was computed by evaluating the stiffness of the equation. The stiffness of the conduction equation of heat is very low and would allow the choice of a very high time step. The overall simulation loop includes many other heat flows that have a greater impact however. We experimented that the solar energy as well as radiative transfers with the sky are the major contributions and give the highest heat flows. Numerical experiments have shown that 600s is an upper bound for spatial resolution of 1m. Moreover, this time step enables us to capture the weather events occurring in the weather scenario with a sufficient accuracy.

## 5. Instantiation

The rendering of snow and ice is a very complex research topic beyond the scope of this paper. This section presents our method for generating the textured mesh representation of water, snow and ice. Our approach consists in generating three different meshes from the voxel model which will be denoted as  $\mathcal{M}_{\text{Water}}$ ,  $\mathcal{M}_{\text{Snow}}$  and  $\mathcal{M}_{\text{Ice}}$  respectively. Smooth transitions between different materials will be obtained by blending the corresponding textures using the relative amount of water, snow and ice as a weighting function.

**Snow** The mesh  $\mathcal{M}_{\text{Snow}}$  representing the snow cover is generated as a textured heightfield covering the ground [HAH02,FG09]. The heightfield is obtained by elevating the vertices  $\mathbf{v}$  of surface mesh of the terrain by a value corresponding to the amount of snow and ice in the voxel  $\mathcal{V}_{ijk}$  containing  $\mathbf{v}$ .

**Water and ice** The generation of water and ice meshes is more complex. The challenge stems from the different water, ice and water-ice voxels that are involved in their characterization. Our method proceeds in two steps. First, we generate a generic mesh  $\mathcal{M}$  by computing the intersection of the surface corresponding to water, ice and water-ice voxels with the terrain. This mesh is then decomposed into a water

mesh  $\mathcal{M}_{\text{Water}}$  and an ice mesh  $\mathcal{M}_{\text{Ice}}$  by analysing the material characteristics at its vertices.

The ice mesh is defined as a combination of a top and a bottom height field corresponding to the top and bottom surface of the ice layer.

To characterize the relative amount of water and ice for every vertex of the mesh, we first compute an ice field function and a water field function, denoted as  $f_{\text{Water}}$  and  $f_{\text{Ice}}$  respectively. Those field functions are defined as a convolution functions as presented in [PGMG09] so as to smooth the discrete ice and water values stored in the voxel representation. For every vertex of the mesh  $\mathcal{M}$ , we compute the thickness of the ice layer  $\delta(\mathbf{v})$  by computing the intersection between the implicit surface  $f_{\text{Ice}}(\mathbf{x}) = 0$  and the vertical line passing at vertex  $\mathbf{v}$ .

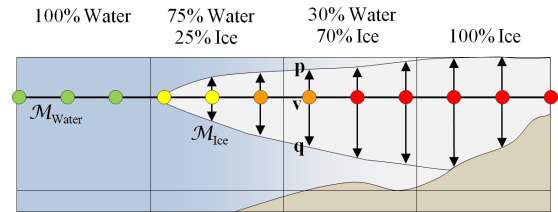


Figure 10: Water and ice instantiation

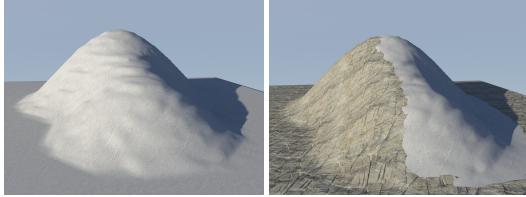
If  $\delta(\mathbf{v}) = 0$ , the vertex  $\mathbf{v}$  belongs to the water mesh and is left unchanged. Otherwise we create the top and bottom vertices  $\mathbf{p}$  and  $\mathbf{q}$  of the ice layer by displacing  $\mathbf{v}$  in the vertical direction. Recall that the volumetric mass of ice and water are  $915$  and  $999 \text{ kg m}^{-3}$  respectively. Thus, 8.5% of ice emerges out of the water surface and 91.5% is immersed. We approximate this phenomenon by defining the top and bottom vertices as  $\mathbf{p} = \mathbf{v} + 0.085 \delta(\mathbf{v})$  and  $\mathbf{q} = \mathbf{v} - 0.915 \delta(\mathbf{v})$  respectively.

**Texture** The water mesh is shaded with a water texture. In contrast, the texture of the ice mesh is obtained by blending two different water and ice textures. The blending coefficient is defined as a function of the thickness of the ice layer.



## 6. Results and discussion

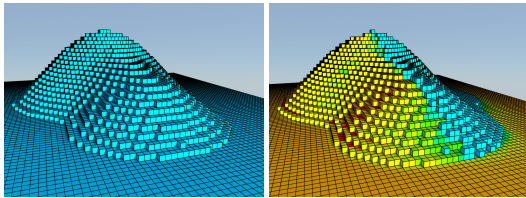
Our method has been implemented in C++ on a Dual-Core Intel®P4 3GHz hardware with 3GB RAM. Results demonstrate that our method can capture complex natural phenomena and generate very realistic winter sceneries.



**Figure 11:** Influence of the sun radiation and the shade over the melting of the snow cover

**Influence of the sun and the shade** Figure 11 shows a hill covered by snow and the evolution of the snow layer when exposed to the sun. In this simulation, the temperature of air was set to 273.15K in order to limit the effect of air on the snow. As expected, the snow melts on the south oriented parts of the hills that are the most exposed to solar rays.

Figure 13 shows the temperature of the voxels at the beginning and the end of the simulation bringing out cooler shaded areas and warmer illuminated areas.



**Figure 13:** A thermal view that shows the warmer side due to solar energy.

**Thermal inertia** Figure 14 illustrates the impact of thermal inertia in the heat transfer simulation. We modeled two small lakes with different water volumes ( $11\text{m}^3$  and  $26\text{m}^3$  respectively) and simulated the evolution of temperature and the water freezing process with the same weather parameters. More precisely, the simulation settings were set as follows:

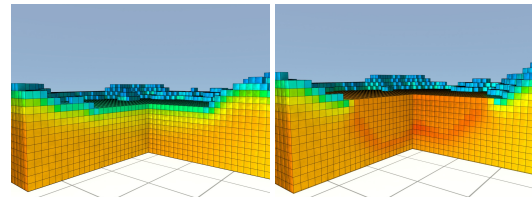
- The initial temperature of the ground and water were 279K and 281K respectively.
- Solar radiation simulation was turned off so that the big lake with a bigger surface would not get more solar energy than the small one.
- The temperature of the air  $T_{\text{air}}$  decreases linearly from 277K down to 269K over a three days period of time.

After two days and a half, the surface of the small lake is completely frozen, whereas the big lake has not yet started



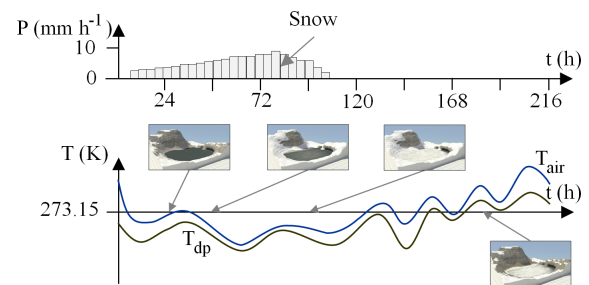
**Figure 14:** After two days and a half, the surface of the small lake is completely frozen (left) whereas the bigger lake (right) still hasn't frozen due to thermal inertia.

to freeze. The corresponding thermal view for the small lake (Figure 15) shows a linear thermal gradient, which means that the temperature of water in the lake is lower than 277K. In contrast, the temperature of the big lake did not reach the 277K threshold.



**Figure 15:** A thermal gradient appears in the small lake (left) as the temperature gets lower than 277K, whereas the deep lake (right) is still under natural water convection.

**Landscape simulation** In the following simulation, we first defined a weather model on a nine days period. The first half of the period is cold and snowy, whereas the second one is warmer. The simulation took 5 hours to compute for a 3.2 million voxels scene. The simplified weather parameters are given in Figure 16.



**Figure 16:** Outline of the weather parameters.

Two hours after the beginning of the snow fall, it begins to hold on the ground. Figure 12 shows the freezing growth onto the lake beginning with the most shaded areas (left and center) and ending with the complete freeze of the lake except in the sheltered area under the rocks (right).

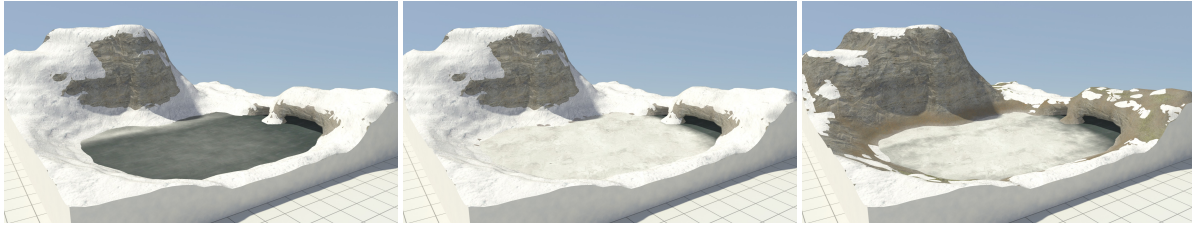


Figure 12: The lake starts freezing as the temperature of the air drops below 273.15K.

**Terrains from digital elevation models** We have modeled the surrounding of Lake Louise (Alberta, Canada) from digital elevation data and performed a simulation reproducing the average weather condition of the region. Figure 9 shows that the lake starts freezing in the region which is mostly in the shade of the surrounding mountains.

## 7. Conclusion

In this paper, we have presented a framework based on a heat transfer simulation for generating realistic winter sceneries and simulating their evolution throughout time. Our results demonstrate that heat transfers play a very important part in the distribution of the snow cover over the ground. Our approach enables us to capture complex natural phenomena including the melting of snow, freezing and melting of water.

Our method can be improved in several ways. An interesting area of research would be to simulate the flow of water over the ground, as well as evaporation. Another interesting topic worth investigating would be to extend our framework and include wind simulation. The wind plays an important role in winter scenery: it moves snow, changes its properties (transforming powder into packed snow, creating solid crust) and can even sublimate it. Finally, the generation of snowdrifts and ice cracks would improve the overall realism.

## References

- [CJ59] CARSLAW H. S., JAEGER J. C.: *Conduction of heat in solids, 2<sup>nd</sup> Edition*. Oxford University Press, 1959. 7
- [CSLW03] CHEN Y., SUN H., LIN H., WU E.: Modelling and rendering of snowy natural scenery using multi-mapping techniques. *Journal of Visualization and Computer Animation* 14, 1 (2003), 21–30. 2
- [DB91] DUFFIE J. A., BECKMAN W. A.: *Solar Engineering of Thermal Processes, 2<sup>nd</sup> Edition*. John Wiley & Sons, 1991. 5, 6
- [FB07] FOLDES D., BENEŠ B.: Occlusion-based snow accumulation simulation. In *The Fourth Workshop on Virtual Reality Interactions and Physical Simulation* (2007), pp. 35–41. 1, 2, 3
- [Fea00] FEARING P.: Computer modeling of fallen snow. In *Proceedings of ACM SIGGRAPH* (2000), pp. 37–46. 1, 2
- [FG09] FESTENBERG N., GUMHOLD S.: A geometric algorithm for snow distribution in virtual scenes. In *Eurographics Workshop on Natural Phenomena* (2009), pp. 15–25. 2, 3, 8
- [FO02] FELDMAN B. E., O'BRIEN J. F.: Modeling the accumulation of wind-driven snow. In *Proceedings of ACM SIGGRAPH* (2002), pp. 218–218. 2
- [HAH02] HAGLUND H., ANDERSSONAND M., HAST A.: Snow accumulation in real-time. In *Proceedings of SIGRAD* (2002). 1, 2, 3, 8
- [KAL06] KIM T., ADALSTEINSSON D., LIN M.: Modeling ice dynamics as a thin-film stefan problem. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2006), pp. 167–176. 1, 2
- [KHL04] KIM T., HENSON M., LIN M.: A hybrid algorithm for modeling ice formation. In *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), pp. 305–314. 1, 2
- [KL03] KIM T., LIN M.: Visual simulation of ice crystal growth. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 86–97. 1, 2
- [LILV08] LIENHARD IV J. H., LIENHARD V J. H.: *A Heat Transfer Textbook, 3<sup>rd</sup> Edition*. Phlogiston Press, Cambridge, Ma, 2008. 5
- [MC00] MURAOKA K., CHIBA N.: Visual simulation of snowfall, snow cover and snowmelt. In *Proceedings of the Seventh International Conference on Parallel and Distributed Systems: Workshops* (2000), pp. 187–194. 2
- [NIDN97] NISHITA T., IWASAKI H., DOBASHI Y., NAKAMAE E.: A modeling and rendering method for snow by using metaballs. *Computer Graphics Forum* 16, 3 (1997), 357–364. 2
- [OS04] OHLSSON H., SIEPEL S.: Real-time rendering of accumulated snow. In *Proceedings of SIGRAD* (2004). 2
- [PGMG09] PEYTAVIE A., GALIN E., MERILLOU S., GROSJEAN J.: Arches: a framework for modeling complex terrains. *Computer Graphics Forum* 28, 2 (2009), 457–467. 8
- [PTS99] PREMOZE S., THOMPSON W., SHIRLEY P.: Geospecific rendering of alpine terrain. In *Eurographics Workshop on Rendering* (1999), pp. 107–118. 2
- [SEN08] SALTVIK I., ELSTER A. C., NAGEL H. R.: Parallel methods for real-time visualization of snow. In *Applied Parallel Computing. State of the Art in Scientific Computing* (2008), vol. 4699, Springer Berlin / Heidelberg, pp. 218–227. 1, 2
- [SOH99] SUMNER R., O'BRIEN J., HODGINS J.: Animating sand, mud, and snow. *Computer Graphics Forum* 18, 1 (1999), 17–26. 2
- [WXXP06] WANG C., WANG Z., XIA T., PENG Q.: Real-time snowing simulation. *The Visual Computer* 22, 5 (2006), 315–323. 1, 2, 3