The Journal of the Society for Art and Science Vol. 8, No. 1, pp. 35 – 42

Air Bubbles in Ice by Simulating Freezing Phenomenon

Carlos MADRAZO[1], Takeshi TSUCHIYA[1],
Hiroaki SAWANO[1,2], and Keiichi KOYANAGI[1]

[1]Graduate School of Information, Production and Systems, Waseda University, Japan

[2]JSPS Research Fellow

[1]carlos@akane.waseda.jp

Abstract: *Ice is a material that is rich in visual details. To achieve a photorealistic image that represents ice in computer graphics (CG) is a challenging issue. Ice in nature is usually not transparent because it contains bubbles. Bubbles are one of the characteristics that define the appearance of ice since they scatter light. To manually incorporate many bubbles in a CG representation would be a daunting task. In this paper we propose a physically based method for creating a geometrical representation of ice with bubbles. The phase change from liquid to solid is considered because it is during the ice formation process that bubbles nucleate in the ice-water interface and accrete in ice. The ice-water interface position is determined by a particle level set method driven by the Stefan conditions. An implicit representation is used to calculate bubble formation in the interface, considering dissolved air in water. Simulation results are compared with actual photographs to evaluate the accuracy of our method.*

Keywords: *Natural Phenomena, Physics-Based Modeling, Particle Level Set Method, Computer Graphics, Ice, Bubbles*

1 Introduction

Ice is a recurrent material in many feature films because of its visual appeal. Filming real ice is difficult because it tends to melt quickly under the lighting fixtures. Imitation ice, made of plastic or clay, can be filmed instead although it looks unrealistic. Computer graphics (CG) images of ice are desirable in order to make a scene in a practical yet realistic way. However, the use of CG-generated ice is usually avoided since it is a challenging task to reproduce a photorealistic image. To achieve photorealism, it is important to consider the detailed visual features of ice. You can notice that ice usually contains bubbles, or trapped air, as shown in Fig. 1. Bubbles radically change the appearance of ice from clear to opaque since they scatter light. To manually set a large number of bubbles in an ice CG representation would be a daunting task and result in implausible images. Therefore, in this paper we follow a physically based approach to solve the problem. In order to determine the bubble characteristics by a physically based approach, the bubble formation and the change of phase from liquid to solid (freezing) processes are simulated.

Fig. 1: Actual ice

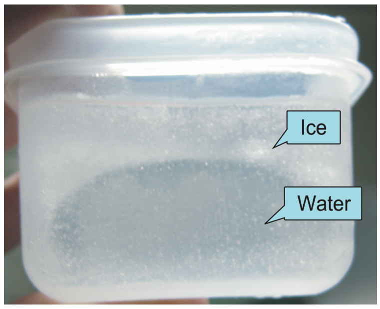


Fig. 2: Actual ice formation

Freezing and the underlying crystal growth are involved in various industrial processes. There are studies [1, 2] in the area of computational physics. In a macroscopic point of view, it is feasible to consider the surface of ice as smooth. The surface between ice and water is named ice-water interface. To predict its position over time is referred as “Stefan problem” [1], and it applies not only to solidification, but also to melting.

There have been studies that aim to visually simulate freezing in CG. Kim *et al.* have simulated two-dimensional ice formation[3]. Although this method aims to simulate the microscopic behavior of crystallization, it is not able to simulate a complex change of phase between water and ice in 3D. Formation of icicles is mentioned by Carlson *et al.* as a melting and flowing phenomenon[4]. However, this method is not suitable for distinguishing between the liquid and solid phases in a sharp interface, such as in actual ice at a macroscopic scale. Kim *et al.* simulated 3D icicle formation by considering water flowing in thin-films[5]. In case of ice growing in thin-films, dissolved air in water tends to escape resulting in ice with a small amount of bubbles. In this study, we consider the case where ice is grown from bulk water, as shown in Fig. 2.

There are various methods for representation of an ice surface. Fujisawa *et al.* have simulated ice melting[6] by using a volume of fluid (VOF) method. In the VOF method, the amount of solid, liquid, or air are tracked at each cell of a 3D Cartesian grid. It is easy to represent the exchange of mass by manipulating these values. However, extra effort has to be made in order to avoid numerical problems and to reconstruct the interface. Implicit surfaces[7], in contrast to VOF, have the advantage of being well-defined without needing extra computation. The calculation for the normal or curvature on any point, and the application of constructive solid geometry (CSG) operations are straightforward[7]. This method is able to simulate solidification since it naturally handles changes in topology. These benefits are exploited for our purpose of integrating bubbles. Implicit surfaces move efficiently by the particle level set method[8].

In this study we propose a physically based method to construct geometrical representations of ice with bubbles. Following, we simulate freezing [9] since the characteristics of bubbles in ice depend mainly on the ice formation rate and the concentration of air in water[10]. The ice-water interface is tracked by a particle level set method. Overall, our method is capable of improving the look and feel of ice and the animation of freezing.

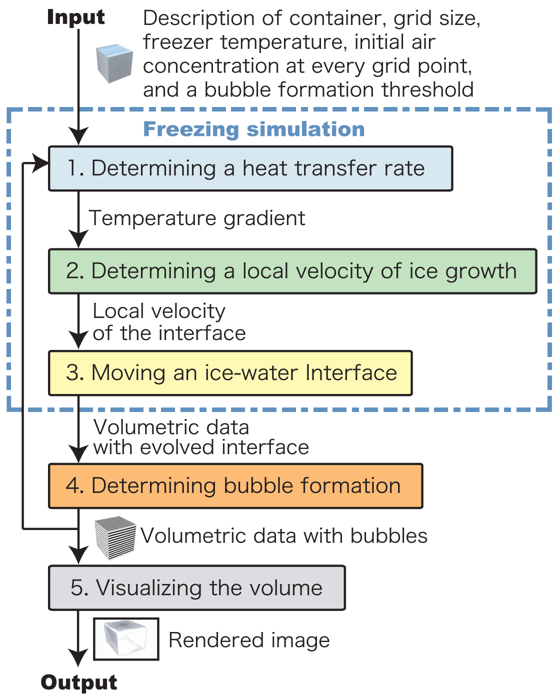


Fig. 3: Overview of the proposed method

2 Proposed method

2.1 Outline

A method is presented for describing the ice growth and determining bubble characteristics. It consists of five steps as indicated in Fig. 3. These steps are:

1. Determining a heat transfer rate,
2. Determining a local velocity of ice growth,
3. Moving an ice-water interface,
4. Determining bubble formation,
5. Visualizing the volume.

In the first three steps, a freezing simulation takes place. In the step 4, the bubble formation characteristics are determined. In the last step, the geometric representation is visualized. These steps are repeated until all of the water becomes ice. A fixed 3D Cartesian grid is used for the calculations.

2.2 Freezing simulation

2.2.1 Determining a heat transfer rate

Heat can be transported by conduction, convection, and radiation. In this research, a model of heat transfer in water that only considers conduction is utilized as we assume it static. Heat conduction is caused by the interactions between micro particles in matter. In our model, an analytical solution considering one-dimensional heat flow is proposed. In this step a temperature gradient ∇T is calculated on each point $\boldsymbol{p}_{i,j,k} = (x_i, y_j, z_k)$ that will be used later to calculate the local velocity of the interface.

The heat flux vector $\boldsymbol{q}$ [W/m^2] in a homogeneous material by conduction is defined by the Fourier's law as follows:

$$\boldsymbol{q} = -\lambda \nabla T \tag{1}$$

where $\boldsymbol{q}$ is proportional to the constant thermal conductivity λ in [W/mK] ($\lambda_{\text{water}} = 0.6$ [W/mK]), and the gradient of temperature T in [K][12].

For a one-dimensional heat transfer approximation as in Fig. 4, it can be integrated to

$$\frac{\partial \boldsymbol{q}}{\partial t} = \lambda A \frac{(T_2 - T_1)}{L} \tag{2}$$

where $\frac{\partial \boldsymbol{q}}{\partial t}$ is the time rate of heat conduction in

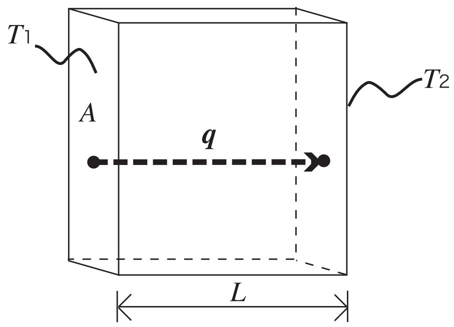


Fig. 4: One dimensional heat transfer

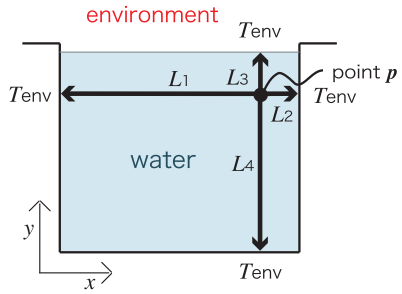


Fig. 5: Determination of local heat

time t in [W], A is the area of the object perpendicular to the heat conduction in [m^2], T_1 and T_2 are the temperatures on each side separated by a distance L in [m].

In our method ∇T is constant in time. It is calculated on each point of a fixed grid by considering its distance to each wall where heat escapes, as shown in Fig. 5. Therefore, T_1 becomes the ambient temperature T_{env} and T_2 a temperature on the interface considered to be equal to the freezing temperature $T_{\mathrm{m}} = 273$ [K]. Component of ∇T in positive and negative x directions are added together as follows:

$$\frac{\partial T}{\partial x} = \left(\frac{T_{\mathrm{m}} - T_{\mathrm{env}}}{L_1}\right) - \left(\frac{T_{\mathrm{m}} - T_{\mathrm{env}}}{L_2}\right) \tag{3}$$

where L_1 and L_2 are the respective distances to each wall. The calculations are done similarly in the case of y and z directions.

2.2.2 Determining local velocity of ice growth

The conditions that determine the velocity of the interface are referred in physics as the Stefan problem[1]. Neglecting surface tension, the process is referred as the “classical” Stefan problem. The problem is solved by finding the temperatures near the ice-water interface. Only the heat field of one of the phases is taken into account in a one-sided Stefan problem. We consider the temperature on the ice-water interface and in the ice region to be equal to the melting point. In this case freezing is driven by the supercooled water (*i.e.* at a temperature below its melting point). At constant pressure, the Stefan problem states that the velocity of the interface is proportional to the heat that passes through it in its normal direction as shown in Fig. 6. The local velocity of the interface on its normal direction $\boldsymbol{v}$ in [m/s] is defined as:

$$\boldsymbol{v} = \mathrm{D}\nabla T \cdot \boldsymbol{n} \tag{4}$$

where D is a thermal diffusion constant in [m^2/s] ($\mathrm{D} = 1.3 \times 10^7$ [m^2/s]), and $\boldsymbol{n}$ the normal vector on the interface [1, 5].

2.2.3 Moving the ice-water interface

The interface is initially defined as an input. Its geometrical description is provided as a polygonal mesh to be converted to an implicit surface by calculating its signed distance field[11], or directly as an implicit surface. Initial ice has to be in a place in contact with water in order to be able to grow.

The interface is moved by a particle level set method, using the velocity field calculated in the previous step. This method needs less computational power to correct errors caused by numerical approximations than the original level set method. This is because auxiliary particles are moved by the same velocity field as the interface instead of solving with a higher order accuracy. The interface is moved to its correct position in the case that a particle escapes from one side of it

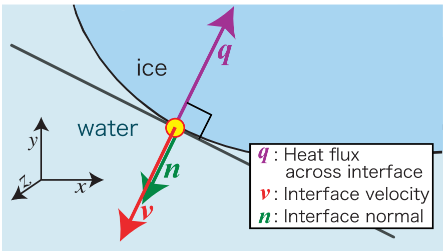


Fig. 6: Heat flux across the ice-water interface

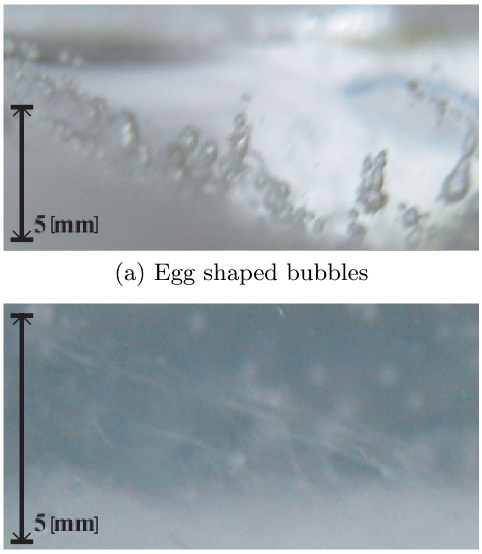
to the other, after a time step. Later, the signed distance field of the interface is recalculated, and particles are seeded again in order to have new particles where needed.

In this step we also account for collision detection. This is achieved by simply setting the local velocity to zero if the interface is not inside the water region. In other words, ice formation cannot occur without surrounding bulk water.

2.3 Bubble Determination

The concentration of bubbles depends mainly on the freezing rate (ice-water velocity)[10]. In a low rate, less and bigger bubbles are formed that can be easily visible, usually with an egg shape as shown in Fig. 7(a). In a high rate, more and smaller bubbles are formed that may have the shape of a cylinder, or a thread, that form in parallel direction of ice growth as shown in Fig. 7(b). The size distribution of bubbles depends on the amount of air dissolved in water.

In this study, we focus on the low freezing rate case, and assume spherical bubbles with a high surface tension. Egg shapes are achieved by joining two or more bubbles together. Initially, an air saturation level $S_{i,j,k} = S(\boldsymbol{p}_{i,j,k})$ is assigned to each point $\boldsymbol{p}_{i,j,k}$ of the fixed grid of the water region. This value is a relative amount of air contained in a cell volume dissolved in water. For example, the amount of air concentrated in boiled water is lower than water that was shaken. When the ice-water interface moves as shown in Fig. 2, the dissolved air is pushed into the water and creates a supersaturated air zone near the ice-water interface. The force that air produces on both ice and water is neglected since its weight is much lower. In the proposed method, dissolved air in points of a region that became ice is pushed into the nearest point in water. An air saturation threshold value $S_{\max}$ is initially defined in order to decide if a bubble will be created. This value can be higher than 100 [%] as water can remain supersaturated before bubble nucleation. The size and number of bubbles depend on the velocity of the interface and on the saturation level.



(a) Egg shaped bubbles

(b) Thin threads

Fig. 7: Bubble shapes in actual ice

a) Rise concentration level on points in water: As mentioned above, the dissolved air is pushed into the water by the moving interface. This process is simulated by adding the air saturation $S_{i,j,k}$ of each point $\boldsymbol{p}_{i,j,k}$ of the grid that became part of the ice and moving it to its nearest point $\boldsymbol{p}_{s,t,u}$ that is on water ($S_{s,t,u} \leftarrow S_{s,t,u} + S_{i,j,k}$), as shown in Fig. 8. We diferenciate points that are in ice or water directly because of the signed distance representation. $S_{i,j,k}$ in points that become ice are cleared since the concentration of air in ice is too small

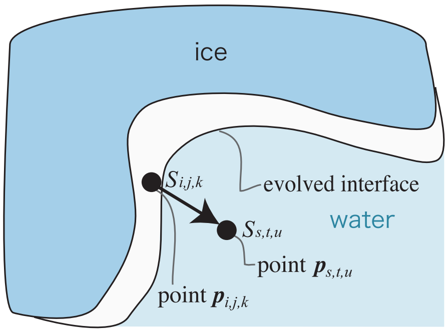


Fig. 8: Air concentration displacement from $\boldsymbol{p}_{i,j,k}$ to $\boldsymbol{p}_{s,t,u}$

to be taken into account.

b) Determine when a bubble is created: Air saturation level $S_{s,t,u}$ in every point $\boldsymbol{p}_{s,t,u}$ in the water region is compared to the established bubble formation threshold value $S_{\max}$. A bubble will form in this point if the saturation level value is higher than the threshold ($S_{s,t,u} > S_{\max}$). The velocity of the interface is not considered directly. However, the interface determines in which points air concentration raises more quickly.

c) Bubble Creation: When a bubble is created, its center position is assigned to the point $\boldsymbol{p}_{s,t,u}$ and its radius assigned proportionally to its air saturation level $S_{s,t,u}$. $S_{s,t,u}$ is cleared, and air concentration of some neighbors is moved to this point. This is done in order to simulate the fact that bubbles form aligned, sometimes forming egg shape bubbles or cylinders, and it is accomplished by combining two or more spherical bubbles. They are stored in an independent implicit surface by a CSG union operation.

2.4 Visualizing the volume

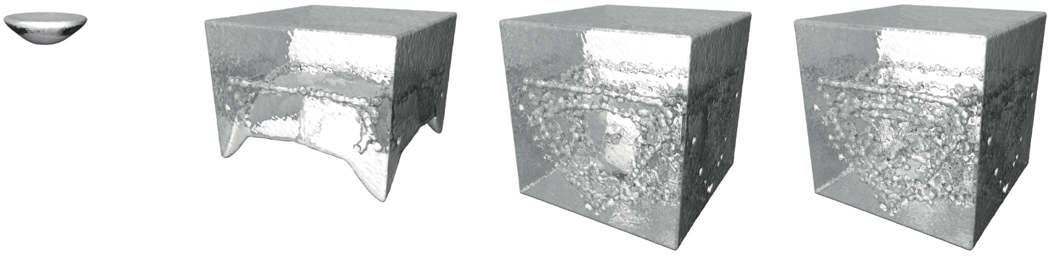
The implicit surfaces that represent bubbles and ice are merged together by a CSG difference operation without modifying the originals. In order to visualize this volumetric data, it is first polygonized using the marching cubes technique[14] and it is rendered by ray-tracing using an ice material with High Dynamic Range (HDR) illumination[15].

3 Experimental Results

A $100 \times 100 \times 100$ grid was used to represent a 4.5^3 [cm^3] ice cube. A time step of $\Delta t = 1/24$ [s] was used. T_{m} was set to the freezing point of the water 273 [K], and the T_{env} was set to 258 [K]. The initial $S_{i,j,k}$ was set to 0.15 (15 [%]) on every point $\boldsymbol{p}_{i,j,k}$. Bubble formation threshold $S_{\max}$ was set to 1.30 (130 [%]) air saturation. In the step 3, a library by Mokbery *et al.*[13] was used as a base code to implement the evolution of the ice-water interface. The initial interface was set in the top surface as a half sphere. For the experiments, the computational environment is an iMac G5, 1.8 [GHz] CPU, Mac OSX ver. 10.4.11, and an NVIDIA GeForce FX5200 Ultra GPU.

Images from the resulting sequence of the experiment are shown in Fig. 9. The average calculation time of the volumetric data was 40 seconds per frame. The average rendering time was 20 minutes per frame at a 640×480 resolution by Mental ray renderer. The resulting total physical time is 37.5 [s]. Freezing occurs quickly because we considered supercooled water. CG ice formation in the simulation resembles the shape of actual ice as in Fig. 2. They first form in the top, follow growing in the walls and enclose water. A second experiment is presented in an annexed video which varies initial interface to a four quarter-spheres placed on the top corners of the cube. In these experiments ideal conditions were considered. In order to obtain more realistic results, the physical model should be extended to include variable heat flows and convection.

A comparison between a model with and without bubbles is shown in Fig. 10. It is observed that photorealism is improved by the inclusion of bubbles. The cube with bubbles is more recognizable as ice than the one without bubbles. We

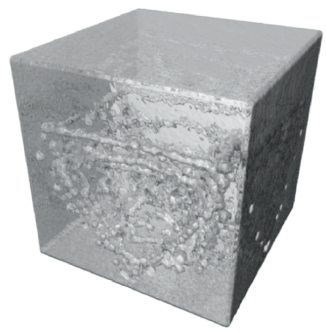


(a) $t = 2.5$ (b) $t = 11.25$ (c) $t = 18.75$ (e) $t = 37.5$

Fig. 9: CG ice cube formation sequence with bubbles (t indicates physical time [s])

also compare CG ice in Fig. 10(b) with actual ice in Fig. 1. In actual ice, micro-bubbles look more like a cloud to the human eye. In the current method, the number and size of bubbles is limited to the size of the grid. Although the current method represents micro bubbles with finer grids, it will be more practical to use a different scheme that does not limit one bubble per grid point. When polygonizing the volume by marching cubes technique, information of smaller bubbles is lost, and a direct visualization scheme will be more suitable.

(a) CG ice without bubbles



(b) Resuling CG ice with bubbles

Fig. 10: Comparison between CG ice without bubbles and our results (4.5^3 [cm^3])

4 Conclusion

In this paper we proposed a physically based method for determining characteristics of bubbles by simulating ice formation. A sequence of images indicated the evolution of our algorithm. CG images of ice with and without bubbles were compared. The result sequence showed that the method is able to reproduce freezing animations. Currently, there are no other studies on integrating bubbles into ice in CG literature. Therefore, our method has great contibution in this field.

Ice is a complex material and further considerations have to be taken into account in order to match the actual ice and bubble formation processes. We considered water as static. A future work is to couple the proposed method with a fluid solver to simulate water movement. Movement of air should be considered. This feature is less visible although it contributes to the determination of heat transfer. The heat transfer between a solid and a fluid is called convection. It is an important characteristic that determines the shape of an ice during formation since heat transfer by this mean is higher than by conduction.

Our experiments were done on a cubic shape. To extend the algorithm to be used in arbitrary shapes, collisions and temperature interpolations, should be handled properly to avoid stair-case artifacts. Defining parameters that couple both ice and bubble formations will be necessary to achieve a more friendly control to the user. Crystal orientation, temperatures, and densities, can be considered for visualization in a future work. The freezing of lakes or waterfalls will be a desirable phenomena to be simulated. Furthermore, simulations of ice sculpturing or winter sports, by physically based animation are among the potential research directions.

References

[1] S. Chen, B. Merriman, S. Osher, and P. Smereka: “A Simple Level Set Method for Solving Stefan Problems”, *J. of Computational Physics*, pp. 8–29, 1997.

[2] F. Gibou, R. Fedkiw, L. Cheng, and M. Kang: “A second order accurate symmetric discretization of the Poisson equation on irregular domains”, *J. Comput. Phys.*, No. 176, pp. 205–227, 2002.

[3] T. Kim, M. Henson, and M. C. Lin: “A Hybrid Algorithm for Modeling Ice Formation”, *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pp. 305–314, 2004.

[4] M. Carlson, P. Mucha, R. Brooks, and G. Turk: “Melting and Flowing”, *Proc. of the 2002 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pp. 167–174, 2002.

[5] T. Kim, D. Adalsteinsson, and M. C. Lin: “Modeling Ice Dynamics as a Thin-Film Stefan Problem”, *Proc. of the 2006 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pp. 167–176, 2006.

[6] M. Fujisawa and K. Miura: “Animation of Ice Melting Phenomenon Based on Thermodynamics with Thermal Radiation”, *Proc. of the 5th Int’l Conf. on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pp. 249–256, 2007.

[7] J. Bloomenthal, C. Bajaj, J. Blinn, M. Cani-Gascuel, A. Rockwood, B. Wyvill, and G. Wyvill: “Introduction to Implicit Surfaces”, *Morgan Kaufmann Publishers, Inc.*, 1997.

[8] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell: “Animation and Rendering of Complex Water Surfaces”, *ACM Trans. Graph. (SIGGRAPH Proc.)*, Vol. 21, No 3, pp. 736–744, 2002.

[9] C. Madrazo and M. Okada: “Physically Based Modeling of Ice with Bubbles”, *Eurographics 2008 Short papers*, Greece, April 2008.

[10] A. E. Carte: “Air Bubbles in Ice”, *Proc. of the Phys. Soc.*, Vol. 77, pp. 757–768, 1961.

[11] M. W. Jones, J. A. Baerentzen, and M. Sramek, “3D Distance Fields: A Survey of Techniques and Applications”, *IEEE Trans. on Visualization and Computer Graphics*, Vol. 12, No. 4, pp. 581-599, 2006.

[12] J. P. Holman: “Heat Transfer”, *eighth ed.*, *McGraw-Hill*, 1996.

[13] E. Mokberi and P. Faloutsos: “Particle Level Set Library”, *Tech. Rep.*, 2006.

[14] W. E. Lorensen and H. E. Cline: “Marching Cubes: a High Resolution 3D Surface Construction Algorithm”, *Computer Graphics*, Vol. 21, No. 4, 1987.

[15] P. E. Debevec: “Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography”, *Proc. of SIGGRAPH 98, Computer Graphics Proceedings, Annual Conference Series*, pp. 189–198, 1998.