Heat Simulation on Meshless Crafted-Made Shapes

Auguste de Lambilly, Gabriel Benedetti, Nour Rizk, Chen Hanqi, Siyuan Huang, Junnan Qiu, David Louapre, Raphael Granier de Cassagnac, Damien Rohmer

Heat Simulation on Meshless Crafted-Made Shapes

Auguste De Lambilly
École Polytechnique, IP Paris

Gabriel Benedetti
École Polytechnique, IP Paris

Nour Rizk
École Polytechnique, IP Paris

Chen Hanqi
École Polytechnique, IP Paris

Siyuan Huang
École Polytechnique, IP Paris

Junnan Qiu
École Polytechnique, IP Paris

David Louapre
Ubisoft Entertainment

Raphael Granier de Cassagnac
LLR, CNRS/Ecole Polytechnique, IP Paris

Damien Rohmer
LIX, Ecole Polytechnique/CNRS, IP Paris

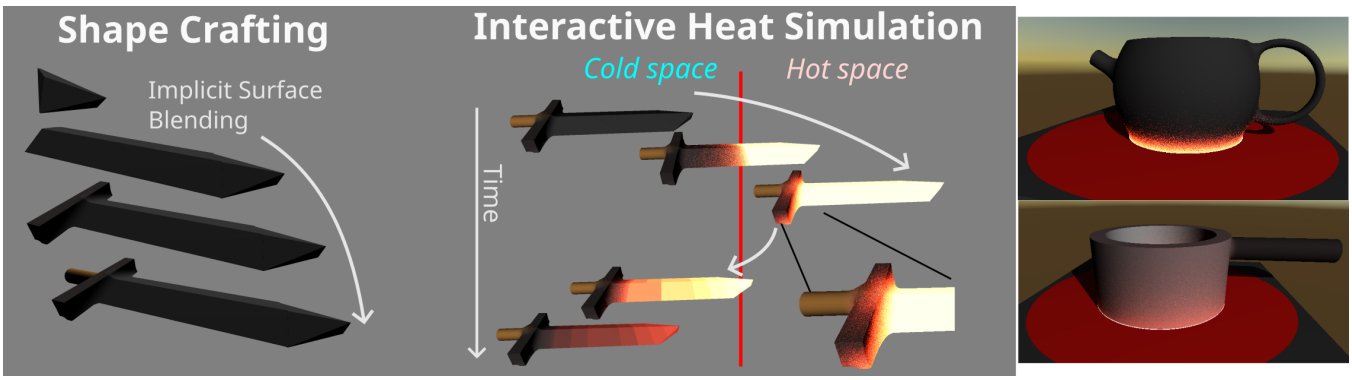

Figure 1: Our method computes the transient heat diffusion on arbitrary meshless shape defined as SDF-based Implicit Surface. Left: Crafting a sword using Implicit Blending Operators. Middle: Solving at interactive time the heat-up, and cool-down on the sword model. Note the slower heating of the cross guard associated with lower heat diffusivity. Right: Arbitrary shapes placed on a hot plate.

ABSTRACT

Interactive shape crafting is an increasingly popular feature in video games, offering players a sense of freedom and personalization. In this work, we propose to combine a stochastic simulation approach to solve the heat equation on Implicit Surface, enabling crafting-ready shapes. Our simulation relies on the "Walk on Sphere" (WoS) approach allowing to solve the asymptotic solution of the heat PDE at any point in space without the need for an explicit mesh structure. To enable interactivity when the shape is moved near a heat source, we propose the integration of time-evolving modifiers. Firstly, using the separation of variables over the PDE enables the approximation of the heating evolution using an additional exponential time variation. Then, we procedurally attach local secondary heat sources to the surface for smooth cool-down. We demonstrate the effectiveness our approach on blended-material shapes generated using CSG operations, combining spatially-varying thermal diffusivity. Overall, our method offers a promising avenue for incorporating often-neglected physical interactions, such as heat-related phenomena, into video games with complex and customizable shapes.

ACM MIG'23, November 15–17, 2023, Rennes, France

CCS CONCEPTS

• **Computer Graphics** → *Volumetric models*; • **Animation** → *Physical Simulation.*

KEYWORDS

Walk on Sphere, PDE, Heat equation, Implicit Surface, Shape Blending, Crafting

ACM Reference Format:
Auguste De Lambilly, Gabriel Benedetti, Nour Rizk, Chen Hanqi, Siyuan Huang, Junnan Qiu, David Louapre, Raphael Granier de Cassagnac, and Damien Rohmer. 2023. Heat Simulation on Meshless Crafted-Made Shapes. In *ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '23), November 15–17, 2023, Rennes, France.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3623264.3624457

1 INTRODUCTION

In the context of video games, shape crafting has gained considerable popularity and has been a fundamental component of some game success, such as *Minecraft* [Mojang Studios 2011] or the recent *The Legend of Zelda: Tears of the Kingdom* [Nintendo 2023]. Crafting means the process where the player can directly combine basic raw materials to generate more complex elements such as tools or weapons [Grow et al. 2020]. Crafting can be used to generate complex geometrical shapes using operations of melting, blending, or carving. This process becomes even more interesting when the

raw physical properties of different materials can combine themselves in the final assembly, thus giving the feeling to the player of being able to create new, fully customizable materials. While crafting mechanism can offer considerable potential to games, it is still hindered nowadays by technical challenges. Firstly, combining shapes using arbitrary operations, such as constructive solid geometry (CSG), is often incompatible with common mesh-based shape representations. Secondly, the ability to combine the physical properties of materials and simulate them is often restricted to predefined, scripted options.

In our work, we consider the specific case of heat diffusion simulation, parameterized by heat diffusivity attached to materials. We present a proof of concept that demonstrates the ability to craft shapes freely using CSG operations while incorporating the blending of thermal diffusivity. Moreover, we enable the seamless simulation of heat evolution within the resulting shape in a game-like environment. Our method allows for the crafting of shapes composed of an assembly of primitive elements with different thermal diffusivity, enabling the interactive simulation of temperature variations when the shape is brought close to a heat source.

Our approach relies on the key ideas of leveraging fully-meshless representation for both shape geometry, and physical heat simulation, thus providing a seamless and flexible crafting process. Firstly, we rely on Implicit Based surface representations that are readily compatible with robust CSG operations. We use, in our case, Signed Distance Fields (SDF) for efficient computation and rendering. Secondly, we employ the Walk on Sphere (WoS) Monte Carlo-based approach to evaluate the solution of the heat equation at arbitrary points inside the volume bounded by the implicit surface. This approach does not require explicit mesh generation and can take advantage of SDF representation for efficient evaluation [Sawhney and Crane 2020]. Applying WoS in an interactive game-like environment raises the challenge that the Monte Carlo approach can only solve the elliptic partial differential equation (PDE) associated with static asymptotic heat diffusion and cannot provide the time-varying evolution of heat directly.

To address this problem, we propose two technical contributions that approximate the time-varying heat evolution within the shape when it is dynamically placed near a heat source. Firstly, we use a separation of variables approach to introduce an additional exponentially time-evolving term to the solution. This time-related term provides an approximation of the transient heat evolution inside the shape based on its characteristic length. Secondly, we ensure a smooth cool-down when the object is moved away from the heat source. We achieve this by considering a set of secondary heat sources represented as small shapes attached to the object surface, which store previous heat values over time.

We demonstrate the effectiveness of our method by presenting examples of blended shapes defined as approximated SDF with combined heat diffusivity. The temperature evolution over the assembled shape can be efficiently computed at interactive time using a compute shader integrated within the popular Game Engine Unity [Unity Technologies 2023]. Although our approach specifically targets heat diffusion, we believe that this approach relying on meshless representation offers a valuable tradeoff between flexibility and accuracy. This provides an opportunity for exploring and integrating a broader spectrum of physics properties into video games, thus enhancing their realism and interactive potential.

2 RELATED WORK

2.1 Shape crafting

The process of crafting object shapes in video games often involves fundamental Constructive Solid Geometry (CSG) operations such as boolean union, intersection, and blending, as well as carving. However, traditional mesh-based representations used in games face challenges when it comes to shape crafting, particularly in performing robust and efficient boolean operations, which remains an active research area requiring advanced numerical techniques [Zhou et al. 2016].

Implicit surface, on the other hand, offer a natural and versatile approach for shape blending. An implicit surface S can be defined as the zero-isovalue of a scalar field f, such as $S = \{p \in \mathbb{R}^3 \mid f(p) = 0\}$ [Bloomenthal et al. 1987]. Implicit surfaces inherently describe volumetric shapes, where the interior of the shape is associated, for instance, with $f(p) < 0$, and its exterior with $f(p) > 0$. Multiple shapes associated to different field values f_i can be combined using operators such that $f = O(f_i)$. Simple operator functions such as min, max, plus/minus, can lead to CSG and shape blending over the isosurface associated to the final combination f. The sequence of operators, called a *blob tree* [Wyvill et al. 1999] can capture the complete CSG tree for complex shape creation, and has already been demonstrated to support the interactive modeling of freeform shapes [Dewaele and Cani 2004; Génevaux et al. 2015].

The field functions used to define the implicit surface can be computed in combining specific functions primitives[Bloomenthal and Shoemake 1991], using global variational formulation [Turk et al. 2001], or stored in discrete voxels[Lorensen 1987]. In this work, we focus on the use of primitives associated to Signed Distance Functions (SDF), where $f(p)$ is the closed Euclidian distance from point p to the surface S, with negative and positive signs indicating points inside and outside the shape, respectively. SDFs have been extensively employed in level-set approaches for boundary tracking [Osher and Fedkiw 2001] and have re-gained recent popularily to learn and represent datasets using neural implicit representation [Yariv et al. 2021]. One notable advantage of SDF is their efficient rendering using the sphere tracing algorithm [Hart 1996; Seyb et al. 2019]. Modeling shape primitives with SDFs and applying blending operators can be achieved via combining primitive functions [Quilez 2008] or shape approximated from datasets [Takikawa et al. 2022].

2.2 Meshless Simulation

Numerical solution for simulating hyperbolic PDEs over a curved domain is typically computed using the Finite Element Method (FEM). FEM has been extensively studied and can provide high accuracy solutions. For arbitrary domains, FEM require well-defined (typically tetrahedral) meshes, which can be excessively expensive to compute and challenging to generate for domain that are interactively generated.

In contrast, meshless-based simulation approaches aim to avoid the need for discretization of the volumetric domain, enabling the robust handling of complex curved, concave, or thin shapes. Boundary

integral methods offer a solution by relying solely on the discretized boundary of the shape. Initially proposed in Graphics by James et Pai [1999], this approach has been further extended to dynamic elastic shape deformation using meshes [Sugimoto et al. 2022] and smooth NURBS [Trusty et al. 2021], or for fluid dynamics [Da et al. 2016]. Nethertheless, these approach still require a discretization of the shape bounday.

Instead, Monte Carlo based approaches offer an alternative by approximating numerical integration on arbitrary domain without requiring any discretization. While the use of Monte Carlo methods have been extensively studied in Graphics for rendering purpose [Jensen 2001; Kajiya 1986], their use in physically-based simulation has received less attention. Initially proposed by Muller [Muller 1956], the Walk on Sphere (WoS) approach is an efficient method to sample integral values along paths where the mean value property holds, i.e. a random walk that starts at a point p_0 can exit with equal probabiliy at any point on a sphere centered at p_0. It ivolves iterative jumping from samples to the surface of spheres centered at the current sample, until reaching proximity of the shape bourndary. This approach was recently exploited in Graphics community by Sawhney and Crane [2020] to approximate the solution for Laplace, Poisson, and Biharmonic PDEs with Dirichlet boundary conditions. WoS has the advantage of working on arbitrary domain without requiring mesh of grid discretization. Further it is a point-based approach allowing to evaluate the solution independently at each point, making it suitable for computation in shaders. Wos takes advantage of SDF representation for speeding-up its computation. Indeed, the iterative process of jumping from sphere to sphere benefits from the distance information to ensure that each new point lies on a maximal sphere within the shape, thus advancing at large distance while avoiding going outside it. Those properties makes the WoS algorithm particularly well-suited for computing heat diffusion over user-defined shapes.

Recent advancements have expanded the capabilities of the WoS approach to address more general PDE, such as handling varying spatial coefficient [Sawhney et al. 2022], and Neuman boundary conditions [Sawhney et al. 2023]. More efficient approach have also been developed via bidirectional formulation combining source to target and target to source paths [Qi et al. 2022], and efficient caching method to avoid recomputing similar values [Miller et al. 2023]. Nabizadeh et al. [2021] further proposed to couple Monte Carlo integration with Kelvin Transformations to Solve the PDEs on infinite exterior domains. Finally, Sugimoto et al. [2022] introduced the *Walk on Boundary* approach whose principle is more similar to ray tracing over the boundary and can handle more general boundary integral formulation for both interior and exterior domain.

WoS is however limited to static equations, and cannot be directly applied to solve time-varying evolution. Interestingly, Rioux-Lavoie et al. [2022] combined the WoS with Monte Carlo backtracing advection to evaluate the Navier Stokes equation at a given space and time position in taking advantage of finite velocity propagation. In the case of heat equation, however, the evolution of the temperature doesn't follow velocity and needs a dedicated method.

3 METHOD OVERVIEW

3.1 Shape crafting

In our approach, we initially consider a set of N primitives as input. Each primitive i is associated with a signed distance field (SDF) f_i and a corresponding thermal diffusivity α_i with

$$\begin{cases} \alpha_i > 0 \text{ for } f_i(p) \leq 0 \\ \alpha_i = 0 \text{ otherwise .} \end{cases} \tag{1}$$

During the crafting process, the user combines these primitives to generate a global SDF $f = O(f_1, \ldots, f_N)$, where O represents the operators applied to the N fields. We call $\partial\Omega$ the shape is obtained as the 0-isosurface of the field f, and Ω the interior domain defined as $f(p) \leq 0$. Similarily, the same operators are applied onto the thermal diffusivity such that

$$\forall p \in \Omega, \alpha(p) = O(\alpha_1(p), \ldots, \alpha_N(p)) \ . \tag{2}$$

We considered the following sets of operators. The clean union, subtraction, and intersection between primitives, obtained respectively with the operators O_u, O_s, O_i

$$\begin{aligned} O_u(f_1, f_2) &= \min(f_1, f_2) \\ O_s(f_1, f_2) &= \max(-f_1, f_2) \\ O_i(f_1, f_2) &= \max(f_1, f_2) \end{aligned} \tag{3}$$

as well as their smooth versions $O_{su/ss/si}$ [Quilez 2008], obtained as

$$\begin{aligned} O_{su}(f_1, f_2) &= a f_1 + (1-a)\, f_2 - K\, a\, (1-a), \\ a &= \lfloor 0.5\, (1 + (f_2 - f_1)/K) \rfloor \\ O_{ss}(f_1, f_2) &= -a f_1 + (1-a)\, f_2 + K\, a\, (1-a), \\ a &= \lfloor 0.5\, (1 + (f_2 + f_1)/K) \rfloor \\ O_{si}(f_1, f_2) &= a f_1 + (1-a)\, f_2 + K\, a\, (1-a), \\ a &= \lfloor 0.5\, (1 - (f_2 - f_1)/K) \rfloor \end{aligned} \tag{4}$$

where $\lfloor x \rfloor = \min(\max(x, 0), 1)$. Note that the function $f = O(f_1, f_2)$ is not any more an exact SDF function when O is a smooth blending operator. However, its gradient satisfies $\|\nabla f\| \leq 1$, thus the function can then still be used in sphere tracing and WoS approach, with slightly decreased efficiency.

We finally propose an interactive substraction effect inspired from games featuring digging material with a shovel. When sufficiently close to the camera, the user is able to apply a local substraction of a given primitive to the shape in front of it.

3.2 Heat diffusion

Once the object is defined, the player can interactively manipulate and bring it toward a heat source, thus seeing dynamically the change of temperature inside Ω as a modification of color. In our case, we consider the general cold air to be at temperature T_c, while the hot heat source is defined as a closed domain $\mathbb{H}$ at temperature T_h given by the user.

These temperatures are used to set the boundary conditions $T_{\partial\Omega}$ at the surface of the shape such that

$$\forall p \in \partial\Omega \left| \begin{array}{l} p \in \mathbb{H} \Rightarrow T_{\partial\Omega}(p) = T_h \\ p \notin \mathbb{H} \Rightarrow T_{\partial\Omega}(p) = T_c \end{array} \right. \tag{5}$$

We then aim at providing an approximate solution to the heat equation inside Ω

$$\begin{cases} \dfrac{\partial T}{\partial t}(p,t) = \alpha(p)\ \Delta T(p,t) & \text{for } p \in \Omega \\ T(p,t) = T_{\partial\Omega}(p) & \text{for } p \in \partial\Omega \\ T(p,0) = T_0(p)\ , \end{cases} \tag{6}$$

where $T_0(p)$ is a given initial condition. Note that the full heat diffusion should consider $\nabla\cdot(\alpha\,\nabla T)$ instead of $\alpha\,\Delta T$. In the following, we will neglect the influence of the term related to the spatial variation of α in the PDE, and discuss possible improvements in future work.

Eq. 6 is associated to a time-varying solution T which cannot be solved directly on arbitrary domain. In a first step, we consider the static, or asymptotic temperature solution in time T^+, that satisfies a Laplace equation with Dirichlet boundary condition

$$\begin{cases} \Delta T^+(p) = 0 & \text{for } p \in \Omega \\ T^+(p) = T_{\partial\Omega}(p) & \text{for } p \in \partial\Omega\ , \end{cases} \tag{7}$$

This second, static equation, can be solved using a WoS approach at any point $p \in \Omega$. For each point p, we compute $T^+(p)$ using N_{walk} random paths inside Ω, until reaching its border $\partial\Omega$ within a threshold ϵ. The detailed algorithm is summarized in Alg. 1, and can be computed in a shader for each visible point of Ω for interactive results. Note that the algorithm converges even if f is not a signed distance function, as long as $\|\nabla f\| \leq 1$. The optimal efficiency is however reached for unit gradient magnitude.

Algorithm 1 WoS to compute $T^+(p)$

```
T+(p) ← 0                                   ▷ Initialize temperature
k_walk ← 0
while k_walk < N_walk do                    ▷ Perform N_walk
    p^k ← p
    k ← 0
    while −f(p^k) > ϵ do                    ▷ Convergence criteria for a walk
        d⃗ ← random direction
        p^{k+1} = p^k + d⃗ f(p^k)            ▷ Walk on Sphere iteration
        k=k+1
    end while
    T+(p)  +=  T_∂Ω(p^k)/N_walk             ▷ Temperature update
end while
```

Applying Alg. 1 at each frame results in quasi-static states, that can lead to sudden variations when the boundary conditions change during the user interaction moving the shape. We therefore propose to improve this framework along two modification described in the following parts. First, in approximating a time-dependant solution for Eq. (6), and second, in ensuring a time-continuous change of the boundary condition $T_{\partial\Omega}$ using secondary sources.

4 TEMPORAL-EVOLVING HEAT APPROXIMATION

Our objective is to approximate the solution of Eq. (6), while preserving the possibility to evaluate it at arbitrary point in space and time without requiring any meshing structure. A standard approach relies on the *separation of variables* in space and time.

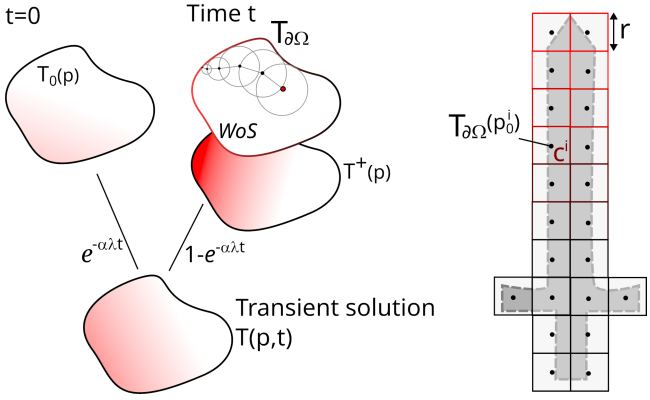

Figure 2: Our temporal adaptation for the heat diffusion. Left: Transient evolution of the solution in blending the initial solution T_0 to the asymptotic solution T^+ computed using the WoS. Right: Grid based approach to store and model smooth time-varying temperature of the boundary $\partial\Omega$.

We first convert the inhomogeneous boundary condition in space into the homogeneous equivalent assuming that the solution can be written as

$$T(p,t) = Z(p,t) + T^+(p)\ , \tag{8}$$

where Z is a transient solution that vanishes when T converges toward its static solution T^+. Z now satisfies the homogeneous equation

$$\begin{cases} \dfrac{\partial Z}{\partial t} = \alpha\,\Delta Z \ \text{ on } \Omega \\ Z = 0 \ \text{ on } \partial\Omega\ . \end{cases} \tag{9}$$

We further assume that Z can be written as product between a time-dependant function $A(t)$, and a space-dependant function $B(p)$.

$$Z(p,t) = A(t)\,B(p)\ . \tag{10}$$

Plugin this relation in Eq. (6) leads to

$$\begin{aligned} & A'(t)\,B(p) = \alpha\,A(t)\,\Delta B(p) \\ \Rightarrow\quad & \frac{A'(t)}{A(t)} = \alpha\,\frac{\Delta B(p)}{B(p)}\ . \end{aligned} \tag{11}$$

As both sides of the equality depends on different variables, their must be equal to a constant $-\lambda$ (negative sign for convenience). If we further assume that we can consider α to be constant in space, we can write the following system that must satisfy A and B

$$\begin{aligned} A'(t) &= -\lambda\ \alpha\ A(t) \\ \Delta B(p) &= -\lambda\ B(p) \end{aligned} \tag{12}$$

A has an explicit solution $A(t) = K\ \exp(-\lambda\,\alpha\,t)$, with K a constant depending on the initial condition. A has a stable solution for any $\lambda \geq 0$. B, on the other hand, is a solution of the Helmholtz equation with Dirichlet boundary condition. The general solution for B can be written as a linear combination of the eigenfunctions with eigenvalues λ corresponding to the spectral decomposition of the interior domain defined by the boundary conditions on $\partial\Omega$. Therefore the valid λ values are restricted to be selected among these eigenvalues. A full decomposition of the spectral modes over a generic domain would be computationally expensive and require spatial meshing of Ω. Instead, we propose the heuristic considering

the first mode of the greatest length L of the bounding box of Ω, such as

$$\lambda = \left(\frac{\pi}{L}\right)^2 . \tag{13}$$

This lambda corresponds to the slowest transient mode over the domain Ω in the case where $B(p)$ would be constant.

Our approximate time-dependant solution approximating the initial condition over T_0 for $t = 0$ and converging toward T^+ can finally be written as (see Fig. 2-left)

$$T(p,t) = e^{\left(-\frac{\alpha(p)\,\pi^2}{L^2}t\right)}\,\left(T_0(p) - T^+(p)\right) + T^+(p) \ . \tag{14}$$

5 SECONDARY HEAT SOURCES FOR SMOOTH COOL-DOWN

The time-dependant solution proposed in Eq. (14) provides a continuous time-varying solution when the shape is placed near a heat source. However, the solution may still exhibit sudden variations if the initial boundary conditions change over time. Such rapid change occurs whenever the shape is moved relative to the heat source (see Eq. (5)). For instance, removing an object away from the heat source then leads to a sudden drop of temperature on its boundary, which in turn affects the temperature throughout the entire object.

In the case of a triangulated mesh, the previous heat value can be saved from one frame to the other, however our meshless approach doe not store the memory of past values. We propose to approximate this memory effect via a lightweight grid-based approach as illustrated in Fig. 2-right. This grid serves as a storage mechanism to represent, on each cell c^i of length r, the smooth time-varying temperature on the shape boundary $\partial\Omega$. Furthermore, the grid follows rigidly the interactive motion of the shape, ensuring that each cell remains locally static with respect to the surface boundary.

In our application, we considered the special case where the object is moved away from the heat source. Let us call t_{out} the time since the mid-cell sample point, at position p_0^i is removed from the heat source. We then consider an exponential temperature variation such that

$$T^{\text{saved}}_{\partial\Omega}(p_0^i, t_{out}) = e^{\left(-\frac{\alpha(p_0^i)\,\pi^2}{L^2}t_{out}\right)}\,(T_h - T_c) + T_c \tag{15}$$

And finally, apply this temperature to the boundary points inside the cell c^i

$$\begin{aligned} T_{\partial\Omega}(p,t) &\leftarrow T^{\text{saved}}_{\partial\Omega}(p_0^i, t_{out}) \\ \text{if } p \in c^i&, \text{ and } t_{out} \geq 0 \end{aligned} \tag{16}$$

Note that this relation only models a slow decrease of the surface temperature when removed from the heat source, while its heating when plugged in is still considered as being instantaneous.

6 RESULTS

Our heat meshless diffusion algorithm is described in Alg. 2. We implemented the prototype using Unity, where all the computations related to the shape rendering and heat simulation are performed in a shader. The implicitly defined object is rendered using a sphere tracing [Lague 2019]. The WoS and its time-varying solution are then locally computed for each visible position of the object found by the sphere tracing. We measured the timing of our approach on a laptop equipped with an Intel i9-11950H 2.6GhZ CPU and an Nvidia GeForce RTX 3080 GPU. The most time-consuming part relates to the WoS algorithm, a tradeoff between quality and computation time is obtained in adapting the number of walks N_{walk} per position. In the presented results, we considered $N_{walk} \in [8, 40]$, and a distance threshold $\epsilon = 0.01$ for objects of unit size. The framerate also strongly depends on the size of the shape in the windows as only visible rendered fragments are associated with the temperature computation. We obtained in average 25 fps (or more) for $N_{walk} = 8$, and about 10 fps for $N_{walk} = 32$. Please see the accompanying video for more details. In order to visualize the temperature inside the shape Ω, and not directly the boundary condition at the surface $\partial\Omega$, the shapes are displayed using a small negative isovalue $\mu = -0.1$, therefore representing a slice slightly under their external surface. The temperature is illustrated by a color coding ranging from dark-red for low heat, to yellow and white for high temperature. The main diffusivity values used for our illustrations are indicated in Table 1.

Figure 1 provides an overview of our approach. The left side illustrates the creation of a sword-like shape using blending operators that combine sharp union and soft blending at the cross guard. The middle-part shows the temporal evolution of the temperature while the sword is placed in a hot environment (right side), and then removed to cool down (left side). The blade, characterized by a high diffusivity value α similar to cooper, heats up quickly, while the cross guard, associated with a lower value similar to steel, remains relatively cooler. A close-up view highlights this temperature difference after a few seconds in the hot environment. A few artifacts can be noted. The presence of noisy dots in the heated surface is a typical result of Monte Carlo sampling, which highly depends on the limited number of WoS paths N_{walk} used for the interactive purpose. Additionally, during the cool-down phase, we can notice sharp transitions between values over the surface. These

Algorithm 2 Time varying heat equation

```
while Game Loop over time t do
    Interaction with the object in space
    Update secondary source samples and t_0          ▷ Eq.(15)
    Perform rendering using Sphere-Tracing           ▷ In Shader
    while Position p ∈ Ω is visible do               ▷ In Shader
        Compute asymptotic solution T+(p) with WoS   ▷ Alg.1
        Compute time-varying temperature T(p,t)      ▷ Eq.(14)
          with Boundary Conditions                   ▷ Eq. (15) and Eq.(5)
    end while
end while
```

Examples	Diffusivity ($10^{-6}m^2/s$)	Material
Fig.4 - top	99	Aluminium
Fig.4 - bottom	23	Iron
Sword Blade	15	Steel
Sword cross guard	117	Cooper

Table 1: Diffusivity values used in our examples.

correspond to the piecewise values of the grid cells used as secondary sources to store the previous temperature outside of the hot space (Fig. 2-right).

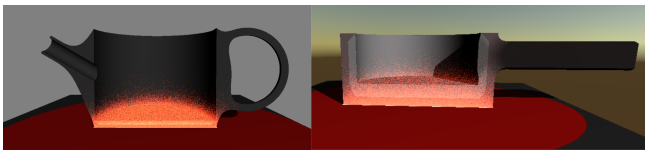

Figure 3: Cut view of the tea-pot and sauce pan placed on a hot bottom surface.

Thanks to the use of the meshless structure, our method can be seamlessly applied to arbitrary shapes, including those with sharp corners, curved boundaries, or hollow interiors. This adaptation is shown at the right side of Fig. 1 where a tea-pot and a sauce-pan are heated from their bottom surface. We show additional cut view of the cut-view of the interior of these two shapes in Fig. 3. Note that the sauce pan on the right is made of three layers with different diffusivity. The external layer of the pan core is the most diffusive, followed by the interior one assumed to have lower value. The handle is itself associated with an even lower diffusivity.

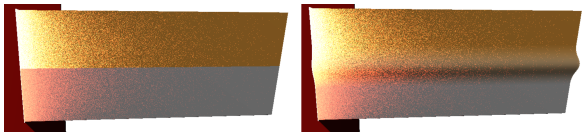

Figure 4: Heating rectangular bars with different diffusivity (higher α value for the top bar). The bars are merged using a simple union (Left), and smooth blending (Right).

We further showcase the flexibility of the method to handle various diffusivity values in combination with the implicit surface modeling in Fig. 4. In this example, two rectangular primitives are heated from their left side. The left image shows a crafting where the two primitives are merged as a sharp union, while the right example shows a smooth blend. The distribution of heat, captured at the same time, shows that the distribution of the heat is similar and can be handled independently of the complexity of the blending operation.

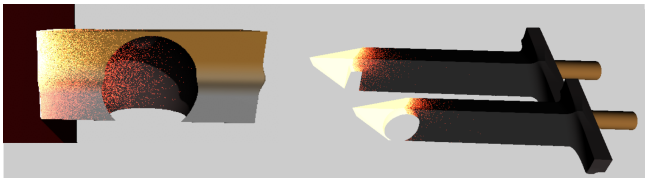

Figure 5: Crafting shapes with local material substraction.

Figure 5 shows the robustness of the method to complex shapes. In this case, the user applied local substraction to the material during the crafting (see Sec. 3.1) as spherical holes in the rectangular model, and as two different notch shapes in the sword blade. Our simulation can readily be applied to the deformed models, while the change of geometry affects the heat propagation.

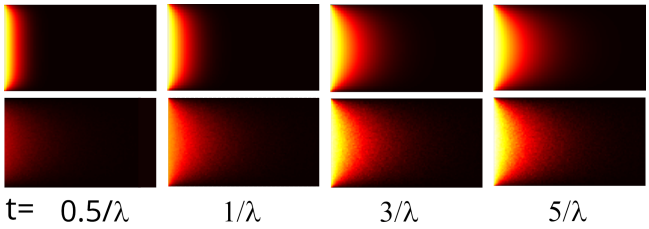

Figure 6: Comparison of the temporal evolution of the heat on a rectangle. Top: Solution of Eq. (6) on a grid using finite differences; Bottom: Our meshless approach.

We finally provide a comparison in Fig. 6 for the accuracy of our method compared to a reference simulation in the case of a rectangular cold bar $T_c = 0$, with its left edge set to be $T_h = 100C^\circ$. The bar is considered to have constant diffusivity $\alpha = 1$. The reference simulation at the top is obtained using finite differences over a discretization of the domain over a fine grid. The bottom row show the results of our meshless approach captured at the same specific time corresponding respectively to 0.5, 1, 3, and 3 times the characteristic time $1/\lambda = L^2/\pi^2$, where $L = 1$m is the horizontal length of the rectangle. Our transient solution obtained from Eq. (14) is only an approximation, and the difference with the reference can mainly be noticed at early time. Indeed, the real solution simulates a propagation of heat, while our approximation models an homogeneous heating toward the final state. Still, general time scale of the heating evolution is well captured, and the local temperature difference becomes less perceptible when the time passes as the Monte Carlo solution converges toward the correct static solution.

7 CONCLUSION AND FUTURE WORK

We have presented an interactive method for computing the temporal evolution of the heat equation inside shapes defined by implicit surfaces, allowing the computation of temperature at arbitrary positions in space using the Walk on Sphere approach. This method is compatible with arbitrary shape crafting in video games, and introduces a new transient time-evolution of heat for plausible heat-up and cool-down effects. Although this Monte-Carlo method provides a noisy approximation to the real solution, its strength relies on the fact that it can be applied only on selected points where knowing the temperature matters. In a video game context for instance, it could efficiently track the evolution of temperature in specific points that are important for triggering certain gameplay actions, like the melting of a part of an object, a fire, or an explosion. When gameplay systems and logic require tracking the temperature only at specific points in space, using such a Monte-Carlo transient method becomes much more efficient than simulating the whole temperature field evolution. We thus believe that our method paves the way for exploring new types of simulations in complex player-modeled shapes, offering possible unexplored gameplay possibilities in the future.

Several limitations still remain to address. First, the computational time required for the simulation remains high for real game production, and the results still exhibit noticeable noise artifacts due to the limited number of path sampling in the WoS. To overcome these issues, dedicated caching structures and filtering techniques can be employed to speed up computation and improve visual quality [Miller et al. 2023]. Second, our current method relies on a series of approximations on the heat PDE. At the spatial level, the variation of thermal diffusivity α is treated as a constant in the PDE. Extending this approach to handling accurately spatial varying coefficient could be handled using the approach proposed by Sawhney et al. [2022] in the static case. In the temporal PDE, an additional advection term depending on $\text{div}(\alpha)$ should be considered and could be handled in a meshless environment using backtracing at the price of higher computational cost [Rioux-Lavoie et al. 2022]. Finally, the proposed time-dependent solution to Eq. (12) is only a rough approximation that does not account for the precise geometry of the shape. In future work, we aim to refine this approximation by incorporating the domain spectral modes. In particular, our preliminary investigation in this direction highlighted that the first mode may not always exert the dominant influence on the final result. We, therefore, believe that considering such additional spectral modes should be a promising avenue for enhancing accuracy.

ACKNOWLEDGMENTS

This action benefited from the support of the "Science and Video Game" chair led by l'X - Ecole polytechnique and the Fondation de l'Ecole polytechnique, sponsored by Ubisoft.

REFERENCES

Jules Bloomenthal, Chandrajit Bajaj, Jim Blinn, Marie-Paule Cani, Allyn Rockwood, Brian Wyvill, and Geoff Wyvill. 1987. Introduction to implicit surfaces. *Morgan Kaufmann* (1987).

J. Bloomenthal and K. Shoemake. 1991. Convolution surfaces. *ACM SIGGRAPH* 25, 4 (1991). https://doi.org/10.1145/127719.122757

Fang Da, David Hahn, Christopher Batty, Chris Wojtan, and Eitan Grinspun. 2016. Surface-Only Liquids. *ACM Transactions on Graphics, Proc. SIGGRAPH* 35, 4 (2016). https://doi.org/10.1145/2897824.2925899

Guillaume Dewaele and Marie-Paule Cani. 2004. Interactive global and local deformations for virtual clay. *Graphical Models, Proc. Pacific Graphics* 66, 6 (2004). https://doi.org/10.1016/j.gmod.2004.06.008

Jean-David Génevaux, Éric Galin, Adrien Peytavie, Éric Guérin, Cyril Briquet, François Grosbellet, and Bedrich Benes. 2015. Terrain Modeling from Feature Primitives. *Computer Graphics Forum* 34, 6 (2015). https://doi.org/10.1111/cgf.12530

April Grow, Melanie Dickinson, Johnathan Pagnutti, Noah Wardrip-Fruin, and Michael Mateas. 2020. Crafting in Games. *Digital Humanities Quarterly* 39, 4 (2020).

John Hart. 1996. Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer* 12 (1996).

Doug L. James and Dinesh K. Pai. 1999. ArtDefo: Accurate Real Time Deformable Objects. *ACM SIGGRAPH* (1999). https://doi.org/10.1145/311535.311542

Henrik W. Jensen. 2001. State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis. *SIGGRAPH Course* (2001).

James T. Kajiya. 1986. The rendering equation. *ACM Transactions on Graphics. Proc. SIGGRAPH* 20, 4 (1986). https://doi.org/10.1145/15886.15902

Sebastien Lague. 2019. Ray Marching. https://github.com/SebLague/Ray-Marching.

William E. Lorensen. 1987. marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics, Proc. SIGGRAPH* 21, 4 (1987). https://doi.org/10.1145/37402.37422

Bailey Miller, Rohan Sawhney, Keenan Crane, and Ioannis Gkioulekas. 2023. Boundary Value Caching for Walk on Spheres. *ACM Transactions on Graphics. Proc. SIGGRAPH* 42, 4 (2023). https://doi.org/10.1145/3592400

Mojang Studios. 2011. Minecraft. https://www.minecraft.net/.

Mervin E. Muller. 1956. Some continuous monte carlo methods for theDirichlet problem. *The Annals of Mathematical Statistics* 27, 3 (1956). https://doi.org/10.1214/aoms/1177728169

Mohammad Sina Nabizadeh, Ravi Ramamoorthi, and Albert Chern. 2021. Kelvin Transformations for Simulations on Infinite Domains. *ACM Transactions on Graphics, Proc. SIGGRAPH* 40, 4 (2021). https://doi.org/10.1145/3450626.3459809

Nintendo. 2023. The Legend of Zelda: Tears of the Kingdom. https://www.zelda.com/tears-of-the-kingdom.

Stanley Osher and Ronald Fedkiw. 2001. Level Set Methods: An Overview and Some Recent Results. *Journal of COmputational Physics* 169, 2 (2001). https://doi.org/10.1006/jcph.2000.6636

Yang Qi, Dario Seyb, Benedikt Bitterli, and Wojciech Jarosz. 2022. A bidirectional formulation for Walk on Spheres. *Eurographics Symposium on Rendering* 41, 4 (2022). https://doi.org/10.1111/cgf.14586

Inigo Quilez. 2008. Modeling with Distance Functions. http://iquilezles.org/www/articles/distfunctions/distfunctions.htm.

Damien Rioux-Lavoie, Ryusuke Sugimoto, Tümay Özdemir, Naoharu H. Shimada, Christopher Batty, Derek Nowrouzezahrai, and Toshiya Hachisuka. 2022. A Monte Carlo Method for Fluid Simulation. *ACM Transactions on Graphics, Proc. SIGGRAPH Asia* 41, 6 (2022). https://doi.org/10.1145/3550454.3555450

Rohan Sawhney and Keenan Crane. 2020. Monte Carlo Geometry Processing: A Grid-Free Approach to PDE-Based Methods on Volumetric Domains. *ACM Transactions on Graphics. Proc. SIGGRAPH* 39, 4 (2020).

Rohan Sawhney, Bailey Miller, Ioannis Gkioulekas, and Keenan Crane. 2023. Walk on Stars: A Grid-Free Monte Carlo Method for PDEs with Neumann Boundary Conditions. *ACM Transactions on Graphics. Proc. SIGGRAPH* 42, 4 (2023).

Rohan Sawhney, Dario Seyb, Wojciech Jarosz, and Keenan Crane. 2022. Grid-free Monte Carlo for PDEs with spatially varying coefficients. *ACM Transactions on Graphics. Proc. SIGGRAPH* 41, 4 (2022). https://doi.org/10.1145/3528223.3530134

Dario Seyb, Alec Jacobson, Derek Nowrouzezahrai, and Wojciech Jarosz. 2019. Nonlinear sphere tracing for rendering deformed signed distance fields. *ACM Transactions on Graphics, Proc. ACM SIGGRAPH* 38, 6 (2019).

Ryusuke Sugimoto, Christopher Batty, and Toshiya Hachisuka. 2022. Surface-Only Dynamic Deformables using a Boundary Element Method. *Computer Graphics Forum, Proc. SCA* (2022). https://doi.org/10.1111/cgf.14625

Towaki Takikawa, Andrew Glassner, and Morgan McGuire. 2022. A Dataset and Explorer for 3D Signed Distance Functions. *Journal of Computer Graphics Techniques* 11, 2 (2022).

Ty Trusty, Honglin Chen, and David I.W. Levin. 2021. The Shape Matching Element Method: Direct Animation of Curved Surface Models. *ACM Transactions on Graphics, Proc. SIGGRAPH* (2021). https://doi.org/10.1145/3450626.3459772

Greg Turk, Huong Quynh Dinh, James F. O'Brien, and Gary Yngve. 2001. Implicit Surfaces that Interpolate. *IEEE Shape Modeling and Applications, Proc. Shape Mondeling International* (2001). https://doi.org/10.1109/SMA.2001.923376

Unity Technologies. 2023. Unity3D. https://unity.com/.

Brian Wyvill, Andrew Guy, and Eric Galin. 1999. Extending the CSG Tree: Warping, Blending and Boolean Operations in an implicit Surface Modeling System. *Computer Graphics Forum. Proc. Eurographics* 18, 2 (1999). https://doi.org/10.1111/1467-8659.00365

Lior Yariv, Jiatao Gu, and Yaron Lipman Yoni Kasten. 2021. Volume Rendering of Neural Implicit Surfaces. *NeurIPS* (2021).

Qingnam Zhou, Eitan Grinspun, Denis Zorin, and Alec Jacobson. 2016. Mesh Arrangements for Solid Geometry. *ACM Transactions on Graphics. Proc. SIGGRAPH* 35, 39 (2016). https://doi.org/10.1145/2897824.2925901