Physically Based Simulation of Solid Objects' Burning

Shiguang Liu[1,2,*], Tai An[1], Zheng Gong[3], and Ichiro Hagiwara[3]

[1] School of Computer Science and Technology, Tianjin University, Tianjin 300072, China
[2] State Key Lab of Virtual Reality Technology and System,
Beihang University, Beijing 100191, China
[3] Department of Mechanical Science and Engineering,
Tokyo Institute of Technology, Tokyo 152-8550, Japan
lsg@tju.edu.cn

Abstract. This paper presents a novel method for realistic simulation of solid objects' burning. The temperature field is first constructed based on combustion theories. Then, a temperature field motivated interaction model (TFMI) is proposed to simulate the interactions between the fire and the objects during burning. In TFMI, the decomposition of the objects is modeled by improving the level set method and the spreading of fire is calculated using the updated temperature field at each time step. Our method can deal with varied topologies of different objects during burning. The fire is simulated by adopting stable fluid method and integrated into the whole burning scenes. Finally, various solid objects' burning scenes are rendered automatically using the above model. The experiment results show the validity of our method.

1 Introduction

Realistic simulation of nature scenes automatically has become one of the research hotspots in computer graphics in recent years. Among them, burning simulation draws more and more attentions which can be found wide applications such as computer games, special effects, fire disaster prevention and rescue, virtual reality, etc. However, most previous works focus on the simulation of fire in burning scenes. How to simulate the interactions between fire and burning objects realistically remains an open problem for computer graphics researchers.

It is well known that the combustible object will begin to burn when the temperature reaches its ignition temperature. During burning, the object will decompose due to the propagation of the temperature in the environment. On the other hand, the process of objects' decomposition will also lead to heat release and thus influence the temperature field. The above physics principles should be considered to simulate a burning scene realistically.

Based on the combustion theories, we present a novel method to simulate solid objects' burning. A temperature field motivated interaction model which is called TFMI in this paper is proposed to model the interactions between the fire and the burning objects. TFMI can be used to simulate the objects' decomposition and the fire spreading for complex burning scene. It can also deal with complex varied topologies. The dynamics of the fire is modeled by the stable fluid method. The simulation of the

* Corresponding author.

Z. Pan et al. (Eds.): Transactions on Edutainment VII, LNCS 7145, pp. 110–120, 2012.

fire and the objects' deformation is modeled in the same 3D space and integrated as a whole framework. The main contributions of our paper can be summarized as follows:

- A novel physically based simulation framework for solid objects' burning is proposed which can deal with various dynamic burning scenes.
- TFMI is specially designed to model the interactions between the fire and the objects during burning.
- Our method can deal with complex varied topologies during burning which is also easy to implement.

The rest of the paper is organized as follows. In Section 2 we give a brief survey of relative works. Then we propose the method of modelling the temperature field in Section 3. Section 4 describes TFMI which is adopted to simulate the interactions between the fire and the object during burning. The implementation and rendering results are discussed in Section 5. Conclusions and future works are at last.

2 Related Work

Recently, more and more attention has been paid to realistic simulation of burning scenes. However, extensive studies are conducted on the simulation of fire phenomena without considering its interactions with the burning objects.

For fire simulation, the existed methods belong to two categories: heuristic methods and physics based methods. Heuristic methods are fast and easy to implement, including particle systems, cellular automata, etc. Physically based methods can simulate more realistic fire scenes but need more computational times. In 1983, Reeves [1] adopted particle systems to simulate fire which was the first fire model in computer graphics. This method is simple and easy to implement. A large amount of particles are needed to generate a realistic scene and the motion of fire is also hard to control by this method. Supposing that fire consisted of some simple cellular, Pakeshi et al. [2] used the theory of Cellular Automata to model fire. By combining the cellars into different patterns, this method can control the motion of fire easily. Wang et al. [3] proposed a unified framework for modeling and animating fire using geometric constrains such as curves, surfaces, etc. To achieve more accurate fire scenes, many physics based fire models are proposed. Rushemeier et al. [4] described the visualization method for fire by using data from pool fires. In 1995, Stam et al. [5] simulated the dynamics of fire by a finite difference scheme. This method can generate realistic motion of fire. However, the solution is not stable. Nguyen et al. [6] introduced a Navier-Stokes equations based fire model. By adopting level set method to track the blue core of the fire, this method can model complex fire with realistic shapes at the cost of a large amount of computations. Considering the physical and chemical theories, Ishikawa et al. [7] presented a novel method to simulate the fire spread. Hong et al. [8] tried to simulate fire with more detailed features like flame wrinkling and cellular patterns using the DSD (Detonation Shock Dynamics) framework. Explosion is the results of instant fire in a limited space, which was also simulated [9, 10] in computer graphics in recent years. Though the above methods can simulate the dynamics and the shapes of the fire realistically, they can not be used to simulate the interactions between fire and the burning objects. Until now, there are only a few works for the simulation of realistic burning scenes considering the interactions between fire and the objects.

Burning solids into gases was simulated by Losasso et al. [11] using a remeshing method. This model doesn't consider the burning object's influence on the motion of the fire and can not generate fine-scale decomposition structures. Melek et al. [12, 13] proposed an interactive method to simulate the objects' decomposition and deformation during burning. However, their method did not consider the object's property such as the calorific value and the specific heat capacity, etc. Also, they suppose the deformation direction of the object as the opposition to the normal. The similar method [14] was also adopted to simulate the bending and crumpling effects of simple objects such as matches and paper. Just recently, Liu et al. [15] proposed a novel unified framework for simulating burning phenomena of thin-shell objects such as paper, cloth, etc. However, this method is difficult to be extended for the simulation of solid objects' burning. In this paper, we propose a novel method for simulation of solid objects' burning, which can simulate the interactions between fire and burning objects realistically. Our method can also capture the fine scales of decomposition for solid objects when they are burning.

3 Temperature Field

During objects' burning, the temperature in the space will vary due to the motion of fire. Meanwhile, the updated temperature field will influence the propagation of the fire. For example, the diffusion of the temperature can make more vertexes' temperature reach their ignition temperature and thus begin to burn. So, calculation of the temperature field is the basis to model the interactions between fire and the object. Figure 1 shows the framework of our temperature field motivated burning simulation.

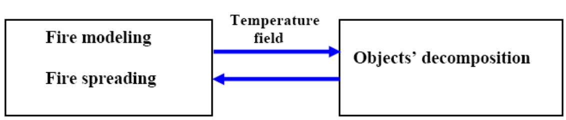


Fig. 1. Temperature field motivated burning simulation

The propagation of temperature will be different for different vertexes in the space. We calculate the temperature for the outer vertexes (including the boundary vertexes) and the interior vertexes, respectively. For the outer vertexes, the temperature will be advected by the velocity field and spread to its six neighboring vertexes. As the object will release heat during burning, new thermal energy will add to the temperature field. We model the temperature of the boundary vertexes as follows:

$$T_t = -(\vec{u} \cdot \nabla)T + K_T \nabla^2 T + T_{add} \,, \tag{1}$$

where T is the temperature to be calculated, T_t is the time differentiation of the temperature with respect to time, $\vec{u}$ represents the velocity of the fire, K_T is the diffusion coefficient which is varied with the position, T_{add} denotes the added temperature due to the heat release of the objects. T_{add} is a function of the calorific value and the specific heat capacity of the object, which can be expressed as follows:

$$T_{add}(\vec{x}) = \begin{cases} \dfrac{q}{c \cdot f_{total}} \cdot \dfrac{\partial}{\partial t} S(\vec{x}), T(\vec{x}) > T_{thres}, f_{left} > 0 \\ 0, T(\vec{x}) < T_{thres} \end{cases}, \tag{2}$$

where $S(\vec{x})$ is the amount of the released fuel of vertex $\vec{x}$ which will be detailed in Section 4, $T(\vec{x})$ is the temperature of vertex $\vec{x}$, f_{total} is the total amount of the fuel, T_{thres} represents the self-pyrolysis temperature of the object, and f_{left} is the amount of the left fuel. In our method, $S(\vec{x})/f_{total}$ is used to control the intensity of the temperature's variation, q and c are the calorific value and the specific heat capacity of the object, respectively.

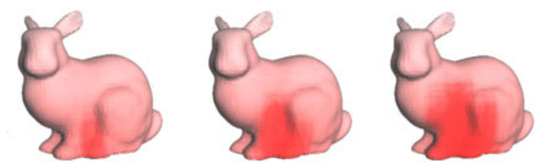


Fig. 2. The varied temperature field of a bunny model at different time

For the interior vertexes, the spread of the temperature is only caused by the diffusion process. We calculate it by $T_t = K_T \nabla^2 T$. By introducing the signed distance field which will be expressed in the following section, we can divide the vertexes into the boundary vertexes, the outer vertexes and the interior vertexes easily. Then, the temperature for different kind of vertexes will be calculated using the above model, respectively. Figure 2 shows the result of the varied temperature field of a bunny model at different time. The ignition point is at the bottom of the model. The color shows the magnitude of the temperature, that is, red represents the highest and white represents the lowest.

4 Temperature Field Motivated Interaction Model (TFMI)

4.1 The Overview of TFMI

Figure 3 shows the sketch map of TFMI. When the object burns, its temperature will increase. The object will decompose when the temperature reaches its self-pyrolysis temperature. At the same time, the object will release heat and influence the temperature field. On the other hand, the object will also be converted into hot gaseous product during its burning, and thus increase the source fuel of the fire. This converted source value can be calculated as follows:

$$\frac{\partial}{\partial t}S(\vec{x})=\begin{cases} S(\vec{x},t)\cdot 2^{\frac{1}{10}\cdot\frac{\partial T}{\partial t}}, \vec{x}\in\partial\Omega, T(\vec{x})>T_{thres}, f_{left}>0, \\ 0, \textit{otherwise} \end{cases} \tag{3}$$

where a is an adjustment parameter, $|\vec{u}|$ is the modulus of the velocity, $\partial\Omega$ represents the boundary of the object, ρ and μ are the density and the viscosity, respectively. As the object converts into hot gaseous product during burning, the left fuel will decrease gradually. The time differentiation of the amount of the fuel of vertex $\vec{x}$ can be expressed as $\frac{\partial}{\partial t}f_{left}(\vec{x})=-\frac{\partial}{\partial t}S(\vec{x})$.

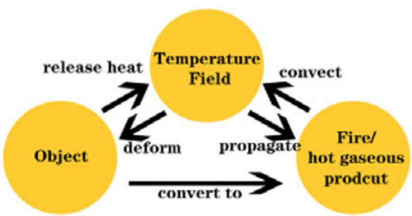


Fig. 3. The overview of TFMI

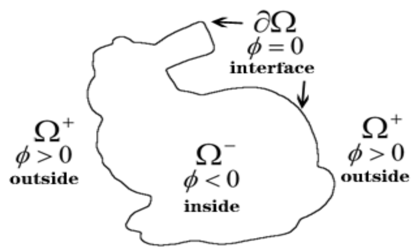


Fig. 4. The isocontour surface in our method

4.2 Level Set Based Object Decomposition

When the temperature reaches the object's self-pyrolysis temperature, it will decompose. In most cases, the object will be burned into pieces. We introduce the physics of burning into the level set method to model the complex process of the object's decomposition.

Level set method is widely used to trace the dynamic interfaces in computer graphics and image processing [16-18]. In our method, the interface between the fire and the object is simulated as one side of an isocontour of a signed distance field, ϕ. The interface $\partial\Omega$ is defined by $\phi=0$ isocontour with $\phi<0$ representing the inside of the object Ω^- and $\phi>0$ representing the outside of the object Ω^+, respectively (see Fig. 4).

The signed distance $d(\vec{x})$ of a vertex $\vec{x}$ is defined as the minimum length from $\vec{x}$ to the interface, that is, $d(\vec{x})=\min(|\vec{x}-\vec{x}_I|)$, where $\vec{x}_I\in\partial\Omega$. The sign of $d(\vec{x})$ is positive, zero and negative when $\vec{x}\in\Omega^+$, $\vec{x}\in\partial\Omega$ and $\vec{x}\in\Omega^-$, respectively. Until now, there are some methods to calculate the signed distance field in 3D space [19-24]. Here, we introduce the method of pseudo normal [25], which can design the signed distance field from the triangle meshes. The interface is dynamically evolved in space and time according to the velocity field $\vec{u}$. This process is expressed by the following equation:

$$\phi_t + \vec{u} \cdot \nabla \phi + n |\nabla \phi| = b\kappa |\nabla \phi|, \tag{4}$$

where ϕ_t is the partial derivative of ϕ with respect to time, ∇ is the gradient operator, κ is the curvature, n denotes the velocity on the direction of the normal, b is a coefficient. During burning, the maximum deformation direction of the object is opposite to the normal. So, Equation (4) can be converted as follows:

$$\phi_t - D(\vec{x}) |\nabla \phi| = 0, \tag{5}$$

where $D(\vec{x})$ is the magnitude of the decomposition of vertex $\vec{x}$ which is a function of the position, the temperature, the property of the object such as the self-pyrolysis temperature, density, etc. We calculate $D(\vec{x})$ as the following:

$$D(\vec{x}) = \begin{cases} (a(T(\vec{x}) - T_{\mathrm{air}}) / \rho |\vec{u}|^2), \vec{x} \in \partial\Omega, T(\vec{x}) > T_{thres}, f_{left} > 0, \\ 0, \; otherwise \end{cases} \tag{6}$$

where a denotes the intensity of the combustion of the object, T_{air} is the average temperature of the air. As Equation (5) satisfies the condition of the signed distance field, it can be discreted as $\phi^{n+1} = \phi^n + D(\vec{x})\Delta t$, here Δt is the time step. This method can deal with complex varied topology during burning. Figure 5 shows the topology variation of a dog model constructed by our method.

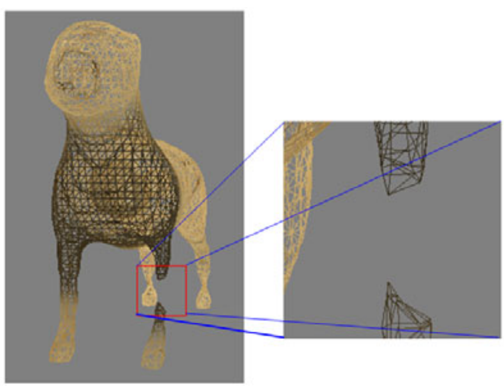


Fig. 5. Topology construction for a dog model during burning

4.3 Fire Spreading

To simulate the spreading of fire, we will first initialize the position of the ignition points. Then, according to the updated temperature field at each time step, we determine which vertexes will reach the combustion conditions. We divide the vertexes into three categories: burning vertex, unburned vertex and burned vertex. Figure 6 shows the results of the spreading of the burning areas. In Figure 6, the black part represents the burned area.

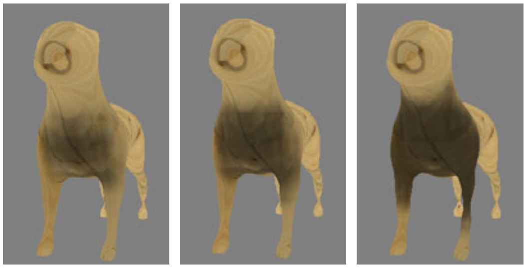
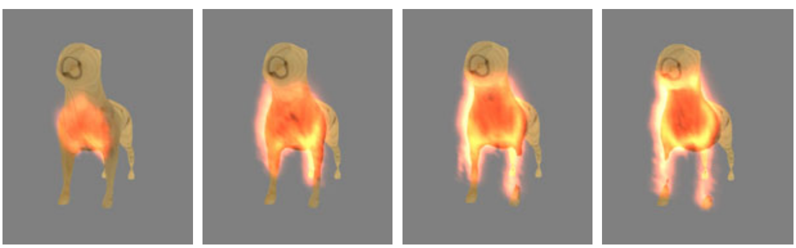


Fig. 6. Results of the spreading of the burning areas

5 Implementation and Rendering Results

Based on the above model, we have rendered various burning scenes of solid objects. All simulations were run on a Pentium IV with 3.0G RAM, nVidia 7800 display card. In our experiments, the time step dt is 0.25, and the values of T_{air} and $T_{\max}$ are $25.0^{\circ}C$ and $600.0^{\circ}C$, respectively. The rendering rates are about 1 second per frame. More meaningful parameters of the different examples are showed in Table 1. RES is the resolution of the space to calculate level set equations.



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

Fig. 7. Burning results of a dog model

Figure 7 shows the simulation results of a burning bunny model using our method. The ignition point is in the front part of the model. We can see that the burning process and the decomposition of the model from Figure 7(a) through 7(d). Our method can deal with the varied topology of the burning model as shown in Figure 7(c) and 7(d). Figure 8 is the burning results of a monster model. Our method can also simulate the burning phenomena generated by multiple ignition points. Figure 9 (a) through 9(f) shows the burning process of a bunny model generated by only one ignition point which lies on the lower part. We can see that the fire spreads upward along the surface of the model accurately. Figure 10 shows the burning results generated by two ignition points. In Figure 10, one ignition point lies on the lower part and the other lies on the upper part. We can see that the two groups of fire spread according to their own paths first and then convergence into one when they meet. At the same time, the decomposition of the bunny model gets faster. In addition, our method can be used to simulate the burning phenomena of very complex models such as a tree, a house, etc. Figure 11 shows the simulation results of a burning tree. In this experiment, we improve the resolution of the simulation space to $200 \times 200 \times 200$ to get accurate results. The fine details of the burning of little branches are showed in Figure 11(a) through 11(d). The varied topologies of the tree model during burning are also simulated accurately.

6 Conclusions and Future Works

In this paper, we proposed a novel method to simulate the burning phenomena of solid objects automatically. This method adopted TFMI to model the interactions between the fire and the objects. The decomposition of the burning object is simulated by introducing the combustion theory into level set method. The decomposition of the object will be converted into hot gaseous product and add influence the fire on the contrary. Our method can also model the above process by calculating the temperature field accurately. This new method can deal with complex objects and varied topologies during the burning process.

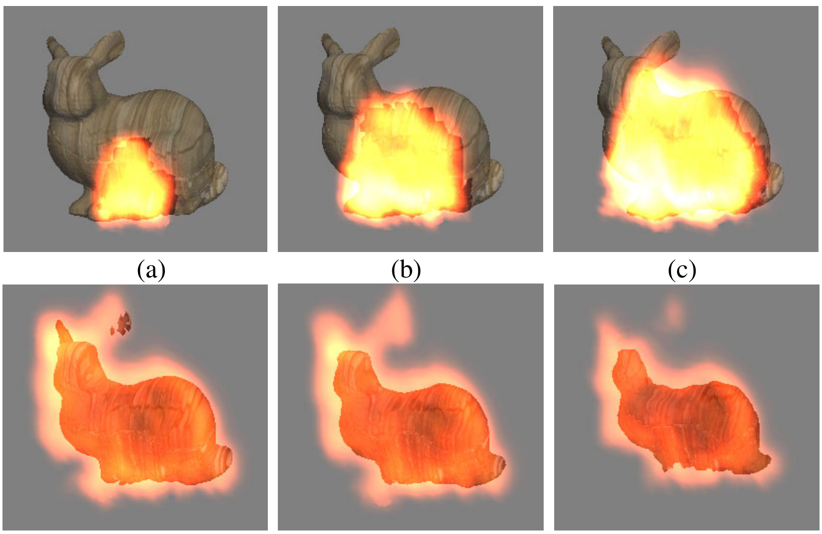
Our method can be integrated into current popular simulation software to add more details of various burning scenes. However, since we don't consider the animation of the burning model, the dynamics of the burning is less realistic. We will improve it in future work by introducing the advanced animation techniques into our simulation framework. By considering the properties of different objects, our method can be extended to simulate the phenomena of melt, explosion, etc. Our future work will also include acceleration and optimization of the algorithms. For example, the adaptive method to calculate the level set equations and the optimization of the re-construction of signed distance field may accelerate the rendering. Modern graphics hardware will also be used to achieve real time simulation in our future work.

Table 1. The parameters and their values in our experiment

Figures	RES	$c\ (J/(Kg\cdot {}^{\circ}C))$	$q\ (J/Kg)$	K_{mat}	$T_{thres}\ ({}^{\circ}C)$
Fig.7	80×80×80	2.1×10^3	0.8×10^7	0.1520	185.0
Fig.8	80×80×80	2.6×10^3	1.0×10^7	0.1250	205.0
Fig.9/10	100×100×100	1.7×10^3	1.2×10^7	0.3750	196.4
Fig.11	200×200×200	3.0×10^3	0.4×10^7	0.0625	260.0

(a) (b) (c)

Fig. 8. Burning results of a monster model



(a) (b) (c)

(d) (e) (f)

Fig. 9. Burning results of a bunny model with one ignition point

(a) (b) (c)

Fig. 10. Burning results of a bunny model with two ignition points

(a) (b) (c)

Fig. 11. Burning results of a tree mode

Acknowledgments. This work was supported by Natural Science Foundation of China under Grant No. 60803047 and 61170118, the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 200800561045, the Open Project Program of the State Key Laboratory of Virtual Reality Technology and System, Beihang University under Grant No. BUAA-VR-10KF-1, the Open Project Program of the State Key Lab of CAD&CG, Zhejiang University under Grant No. A1210. The authors would also like to thank the reviewers for their insightful comments which greatly helped improving the manuscript.

References

1. Reeves, W.T.: Particle Systems-a Technique for Modeling a Class of Fuzzy Objects. In: Proceedings of SIGGRAPH, pp. 359–376 (1983)
2. Pakeshi, A., Yuz, K., Nakajima, M.: Generating Two Dimensional Flame Images in Computer Graphics. IEICE Transactions 74(2), 457–462 (1991)
3. Hong, Y., Zhu, D., Qiu, X., Wang, Z.: Geometry-Based Control of Fire Simulation. The Visual Computer 26 (2010)
4. Rushmeier, H.E., Hamins, A., Choi, M.Y.: Volume rendering of pool fire data. In: Proceedings of IEEE Conference on Visualization, pp. 382–385 (1994)
5. Stam, J., Fiume, E.: Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes. In: Proceedings of SIGGRAPH, pp. 129–136 (1995)
6. Nguyen, D., Fedkiw, R., Jensen, H.W.: Physically based modeling and animation of fire. In: Proceedings of SIGGRAPH, pp. 721–728 (2002)
7. Ishikawa, T., Miyazaki, R., Dobashi, Y., Nishita, T.: Visual simulation of spreading fire. In: Proceedings of NICOGRAPH International, pp. 43–48 (2005)
8. Hong, J., Shinar, T., Fedkiw. R.: Wrinkled flames and cellular patterns. In: Proceedings of SIGGRAPH, pp. 47.1–47.6 (2007)
9. Yngve, G.D., O'Brien, J.F., Hodgins, J.K.: Animating explosions. In: Proceedings of SIGGRAPH, pp. 29–36 (2000)
10. Feldman, B.E., O'Brien, J.F., Arikan, O.: Animating suspended particle explosions. ACM Transactions on Graphics 22(3), 708–715 (2003)
11. Losasso, F., Irving, G., Guendelman, E., et al.: Melting and burning solids into liquids and gases. IEEE Transactions on Visualization and Computer Graphics 12(3), 343–352 (2006)

12. Melek, Z., Keyser, J.: Interactive simulation of burning objects. In: Proceedings of Pacific Graphics, pp. 462–466 (2003)
13. Melek, Z., Keyser, J.: Driving object deformations from internal physical processes. In: Proceedings of the ACM Symposium on Solid and Physical Modeling, pp. 51–59 (2007)
14. Melek, Z., Keyser, J.: Bending burning matches and crumpling burning paper. In: Proceedings of SIGGRAPH Poster (2006)
15. Liu, S., Liu, Q., An, T., Sun, J., Peng, Q.: Physically based simulation of thin-shell objects' burning. The Visual Computer 25(5-7), 687–696 (2009)
16. Osher, S., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Springer, New York (2002)
17. Fedkiw, R.: Geometric Level Set Methods in Imaging, Vision and Graphics. Springer, New York (2003)
18. Mitchell, I.M.: The flexible, extensible and efficient toolbox of level set methods. Journal of Scientific Computing 35(2-3), 300–329 (2007)
19. Sethian, J.A.: Level Set Methods and Fast Marching Methods. Cambridge University Press, New York (1999)
20. Strain, J.: A fast modular semi-lagrangian method for moving interfaces. Journal of Computational Physics 161(2), 512–536 (2000)
21. Sethian, J.: A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences 93, 1591–1595 (1996)
22. Hugues, H., Tony, D., Tom, D., John, M., Werner, S.: Surface reconstruction from unorganized points. Computer Graphics 26(2), 71–78 (1992)
23. Desbrun, M., Meyer, M., Schroder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceeding of SIGGRAPH, pp. 317–324 (1999)
24. Ken, M., David, E.B., Ross, T.W., Alan, H.B.: Level set surface editing operators. In: Proceedings of SIGGRAPH, pp. 330–338 (2002)
25. Aanaes, H., Baerentzen, J.A.: Pseudo-normal for signed distance computation. In: Proceedings of Vision, Modeling and Visualization, pp. 407–413 (2003)