

# Physically based simulation of thin-shell objects' burning

Shiguang Liu · Qiguang Liu · Tai An · Jizhou Sun ·  
Qunsheng Peng

© Springer-Verlag 2009

**Abstract** We present a novel unified framework for simulating burning phenomena of thin-shell objects such as paper, cloth, etc. A gradient-based spreading model is proposed to simulate the movement of fire on thin-shell objects. To simulate the crumpling and deformation of thin-shell objects when burning, we present an FFD-based deformation model which takes the physical theory of burning phenomena into account. The fire is rendered by the method of particle systems. By inputting different parameters and initial conditions, our framework can simulate various burning phenomena of different thin-shell objects. Experiments validate that it is efficient and easy to implement.

**Keywords** Physically based modeling · Thin-shell objects · Burning · Spreading model · Deformation

---

S. Liu (✉) · T. An · J. Sun  
School of Computer Science and Technology, Tianjin University,  
Tianjin 300072, People's Republic of China  
e-mail: [lsg@tju.edu.cn](mailto:lsg@tju.edu.cn)

S. Liu  
State Key Lab of Computer Science, Institute of Software,  
Chinese Academy of Science, Beijing 100190,  
People's Republic of China

S. Liu · Q. Peng  
State Key Lab of CAD&CG, Zhejiang University,  
Hangzhou 310058, People's Republic of China

Q. Liu  
School of Computer Software, Tianjin University,  
Tianjin 300072, People's Republic of China

## 1 Introduction

Although much work has been done on simulation of fluid phenomena such as fire, water, smoke, explosion, etc., relatively little attention has been paid to modeling and rendering of the coupling of fluid with objects. One reason may be the complex physical mechanisms of these phenomena. Realistic simulation of objects' burning is now attracting more and more attention around the world, with wide applications, such as computer games, special effects, disaster prevention, etc.

Simulation of object burning encompasses many aspects such as modeling of fire, modeling of fire spreading, modeling of the crumpling and deformation of objects, etc. Among previous works focused on fire simulation, few have simulated fire spreading and deformation of objects which are indispensable parts of burning phenomena. Paper, cloth, etc. are all inflammable objects. They do not contain an interior region and we call them thin-shell objects. It is well known that thin-shell objects will crumple and deform largely when burning. On the other hand, the deformation will also influence the spreading of fire on thin-shell objects. So, how to model the above physical process realistically is a challenge for computer graphics researchers.

This paper proposes a unified simulation framework of burning phenomena for thin-shell objects. The main contributions of this work can be described as follows:

- A gradient-based spreading model is proposed to simulate the movement of fire on thin-shell objects.
- An FFD (Free-Form Deformation) based deformation method is designed to model the deformation of various thin-shell objects when they are burning.
- As far as we know, it is the first attempt to simulate burning phenomena of thin-shell objects under a unified framework.

- Our system is easy to implement. By inputting different physical parameters, different burning scenes can be produced automatically. This includes potential applications in computer games, special effects, and so on.

The paper is organized as follows. In Sect. 2 we give a brief survey of related work. Then we propose a gradient-based fire spreading model in Sect. 3. Section 4 describes the deformation model of burning thin-shell objects. The implementation and rendering results are discussed in Sect. 5. Conclusions and future works conclude the paper.

## 2 Related work

In recent years, there has been a large amount of research on simulating fluid phenomena such as fire, smoke, water, etc. [1–4]. Most of this work is devoted to simulating fluid itself. Relative little attention has been paid to simulation of the interactions between fluids and objects.

Two-way coupled methods simulate the interaction effects of fluid and solid objects, for example, an explosion's damage to a wall, floating objects in water, etc. Yngve et al. [5] proposed a method for simulation of explosion and its damage to environmental objects. The motion of blow wave was expressed by compressible Navier–Stokes (NS) equations. Carlson et al. [6] presented a method for simulation of two-way interaction between rigid objects and fluids. In their method, they first resolve the occupied area by fluid, and then the rigidity and rigid motion are enforced by applying deformation-free constraints. The results were wonderful. However, their method was too complex when handling thin objects. Guendelman et al. [7] improved the method. They addressed the two-way interaction between fluids and rigid thin shells. Liu et al. [8] proposed a Graphics Processing Unit (GPU) based method to simulate interaction between fluids and objects with complex boundaries.

The above methods mainly focus on gas–solid or liquid–solid systems. Unfortunately, no model was suitable for modeling burning phenomena of thin-shell objects such as paper, cloth, etc.

Modeling fire is also important in a burning scene. There are two categories for simulation of fire, heuristically based methods and physically based methods. Early researches mainly belong to the former. Reeves [9] adopted particle systems to simulated fire. By controlling the motion of the fire particles, they can get different shapes of fire realistically. Pakeshi et al. [10] proposed a fire model based on the theory of Cellular Automata. They supposed that the fire consists of some simple cellular automaton. By combining the automata differently, they can animate the motion of fire. For the latter, Nguyen et al. [11] presented a fire model based on Navier–Stokes equations. They defined the fire front as a moving boundary and used a level set method

to capture very complex motion of the fire. Just recently, Hong et al. [12] simulated interesting fire features such as the flame wrinkling and cellular patterns based DSD (Detonation Shock Dynamics) framework. There is also some work about explosion [13, 14], a special kind of fire burning. Physically based methods can simulate fire more realistically, but they will cost more in computation time.

However, only a few works in the field of computer graphics concern the simulation of burning phenomena. Losasso et al. [15] modeled burning objects using remeshing. This method cannot model fine scale decomposition structures. Melek et al. [16–18] implemented a simple model for simulation of the decomposition of burning objects using a level set method. Although these methods can simulate one kind of burning phenomenon, they cannot be applied to other burning objects easily. Also, none of these simulate the fire spreading on thin-shell objects, which will affect the temperature field of the fire and further largely affect the deformation of objects. In this paper, we propose a novel unified framework for simulating burning phenomena of thin-shell objects. We can simulate the spreading of fire on objects realistically. Our method will also capture the fine scale deformation of thin-shell objects when they are burning.

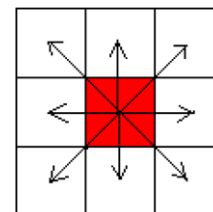
## 3 Fire spreading model

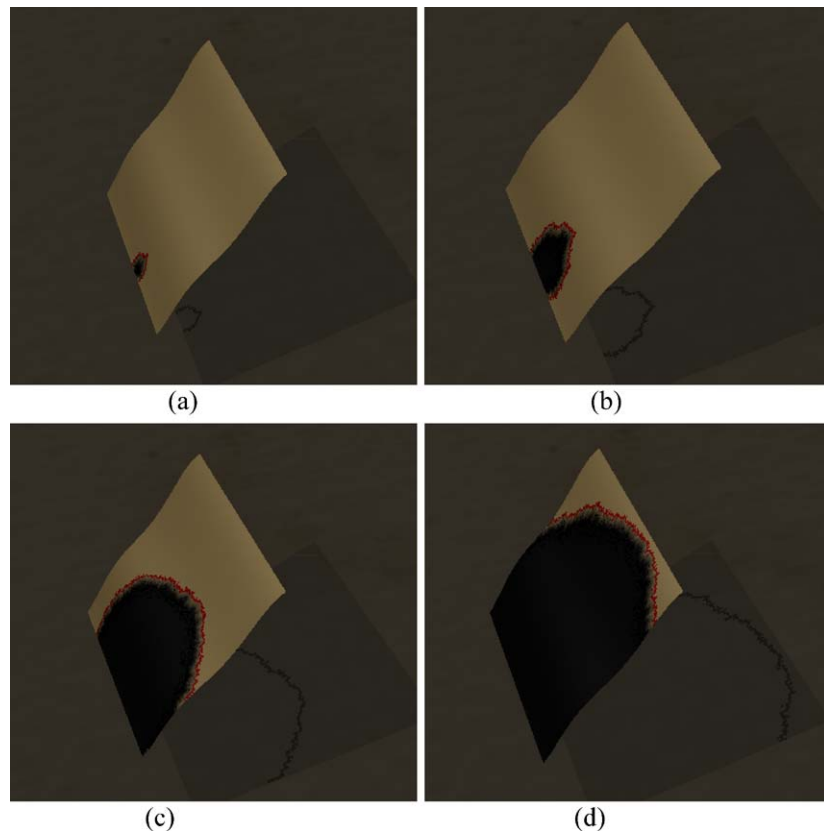
When thin-shell objects are burning, fire will spread through them according to the material properties of objects, wind force, etc. The spreading route and velocity of fire depend on the geometry and material properties of objects. On the other hand, the spreading of fire will affect the distribution of temperature and thus affect the deformation of objects. So, it is very important to model the fire spreading. Below we propose a gradient-based fire spreading model.

### 3.1 Spreading of the ignition point

In our method, the thin-shell object is divided into  $N \times N$  cells. Figure 1 shows a part of the cells. Each cell has three states: burning, burnt and unburnt. Once a cell catches fire, it becomes a burning cell which is called an ignition point. Each ignition point has several attribute values, for example, initial time of catching fire, time interval for burning. After that definite time interval, the cell will become a burnt cell.

**Fig. 1** Sketch map of the spreading of an ignition point



**Fig. 2** Spreading of fire on a piece of paper

Provided that a cell has become a burning cell, fire will spread from it to the adjacent cells at a certain spreading speed. In Fig. 1, the red cell represents a burning cell and others are the so-called adjacent cells. We can get the initial time of catching fire of the adjacent cells by calculating the sum of the initial time of the ignition point and the spreading time. If the initial time of catching fire is earlier than the current time, this adjacent cell will become a burning cell.

A fringe cell is a burning cell where at least one of its adjacent cells is not a burning or a burnt cell. In each calculation time step, we will find all the fringe cells and use them to update the spreading of fire. Figure 2 shows the spreading of fire on a piece of paper. The black part and the red part denote the burnt cells and the fringe cells, respectively.

### 3.2 Calculation of the spreading speed

The spreading speed of a fire from an ignition point to its adjacent cells depends on many factors, e.g. the gradient of the geometry, the material of different thin-shell objects. We calculate it by the following equation:

$$V = V_0 \cdot (1 - C_v \cdot \text{grad}^2), \quad (1)$$

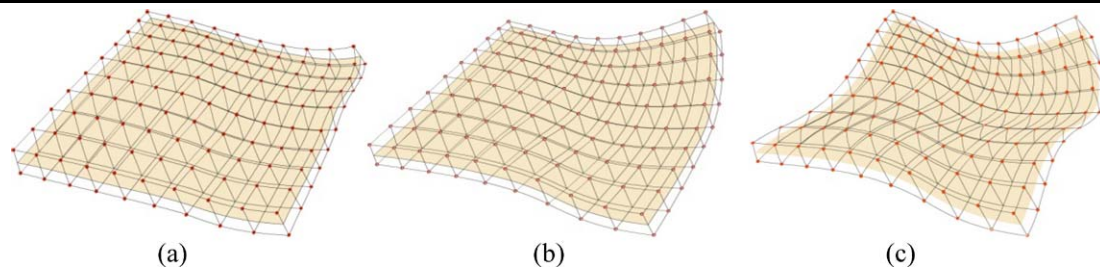
where  $V_0$  is a constant which is related to the material of a different object,  $C_v$  is a coefficient of spread and grad is the gradient of the burning area.

## 4 Deformation model of burning thin-shell objects

During the burning of thin-shell objects, gas will be released, followed by loss of water, decomposition and additional structural changes of the objects. Although these changes are minor, their accumulation results in a considerable change in ways of bending or crumpling of thin-shell objects [16]. The difference of the amount of heat between two faces of thin-shell objects causes an imbalance which leads to the deformation. We propose a novel deformation model of thin-shell objects based on the above physical theory.

### 4.1 Method of controlling deformation

FFD is an important tool in geometric design and animation. In FFD, a geometric model is enclosed within a lattice of control vertexes. Any point on the model with Cartesian coordinates in world space has corresponding coordinates in lattice space. When the control vertexes are changed, the Cartesian coordinates of the model are re-computed [19]. Besides the FFD, there are also many other methods for deformation simulation. Terzopoulos et al. [20] developed deformable models based on the mechanical laws of continuous bodies. Baraff et al. [21] proposed a cloth animation system which can stably take large time steps.



**Fig. 3** The FFD control mesh of thin-shell objects

Rather than modeling the actual chemical process, we propose a simplified model based on the FFD method to mimic similar behavior. Figure 3 shows the control mesh of thin-shell objects. It also deforms when thin-shell objects are burning. The red points are the control vertices which divide the thin-shell objects into different areas. Each control vertex is in charge of one of the above areas.

We map the property parameters such as the amount of pyrolysis, the amount of heat, etc., of each cell onto adjacent couples of the FFD control vertices. Here, pyrolysis is a chemical process in which thin-shell objects will be converted by heat to one or more other products. A couple of the FFD control vertices has a meaning of the two control vertices on the top face and bottom face of the control mesh, which are in charge of the same area of a thin-shell object. The difference of pyrolysis and heat mapped onto one couple of control points determines the deformation amount of the area represented by that couple of control points. Then we can animate the position of each cell of a thin-shell object by changing the position of its corresponding control points.

#### 4.2 Influence factors of deformation

The main factors that determine the deformation of thin-shell objects during burning are: the influence of fire particles' heat on the control vertex and the influence of ignition point on the control vertices.

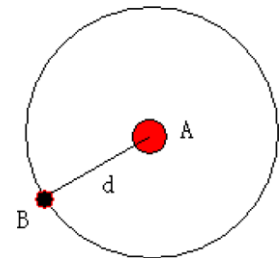
##### (1) Influence of fire particles' heat

The fire is modeled by classical particle systems. The heat will be given off from each fire particle when burning. The amount of heat depends on the temperature of each particle. The heat from each fire particle results in a local imbalance of the temperature, and thus leads to the deformation of the thin-shell object.

As shown in Fig. 4, suppose that the distance from a particle A to a control vertex B is  $d$ , the amount of heat received by control point at B in each time step can be calculated by the following equation:

$$T = C_t \cdot T_{\text{current}} \cdot 1 / (4 \cdot P_i \cdot (d + \text{dmin})^2) \cdot dt, \quad (2)$$

**Fig. 4** Sketch map of a particle's emitting heat



where  $C_t$  is a constant coefficient,  $T_{\text{current}}$  denotes the current temperature of the fire particle.  $\text{dmin}$  is a minimum distance. The total amount of heat received by a control vertex is the sum of heat amount received from each particle.

##### (2) Influence of ignition points

The ignition points will affect the pyrolysis amount of the areas mapped onto control vertices. The pyrolysis amount  $P$  mapped onto a control point can be obtained by the following equation:

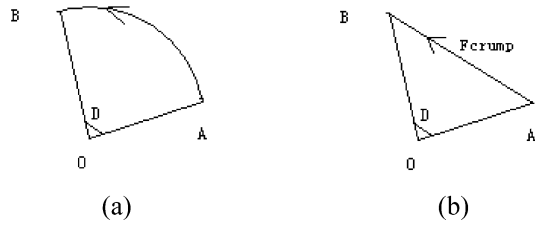
$$P = C_p \cdot 1 / \text{dis}, \quad (3)$$

where  $C_p$  is the pyrolysis coefficient,  $\text{dis}$  denotes the distance from the ignition point to the control vertex. We can see that the amount of pyrolysis is in inverse ratio to  $\text{dis}$ .  $T$  and  $P$  will determine the deformation  $D$  of thin-shell objects together. We discuss this below.

#### 4.3 Calculation of the deformation

Because of an imbalance caused by the differences of  $P$  and  $T$  between control vertices of a couple, the area of thin-shell objects which is operated on by these control vertices will bend and crumple. We first map  $P$  and  $T$  of the top face and bottom face of thin-shell objects onto a couple of control vertices. Then we can use differences of  $P$  and  $T$  between control vertices of this couple to compute the amount of deformation.

We express the bend and crumple of thin-shell objects by letting the adjacent areas rotate around the area where deformation occurs. For each control vertex where deformation occurs, we first select a line crossing this control vertex and



**Fig. 5** Convert the amount of rotation  $D$  to a force

parallel to  $X$  or  $Y$  axis. Then we let the control adjacent vertexes perpendicular to this line rotate by it. The rotation angle is used to represent the amount of deformation  $D$ . Suppose that  $i$  and  $j$  are control vertexes of one couple, then the deformation  $D^{i,j}$  of the relevant area operated on by them can be computed as follows:

$$DT^{i,j} = \Delta T_i - \Delta T_j, \quad (4)$$

$$DP^{i,j} = \Delta P_i - \Delta P_j, \quad (5)$$

$$D^{i,j} = \alpha DT^{i,j} + \beta DP^{i,j}, \quad (6)$$

where  $T_i$  and  $P_i$  denote the amount of heat and pyrolysis mapped onto the control vertex  $i$ , respectively.  $T_j$  and  $P_j$  denote the amount of heat and pyrolysis mapped onto the control vertex  $j$ , respectively.  $\alpha$  and  $\beta$  are deformation coefficients.

To calculate the deformation efficiently, we convert the rotated amount to a force which is called the crumple force. Then we use it to change the control vertex's position.

Figure 5(a) shows a control vertex  $A$  rotate by the crumple area  $O$  to a new position  $B$ . This process is as if the control vertex moves from  $A$  to  $B$  along line  $AB$  under a constant force  $\vec{F}_{\text{crump}}$ . By the physical theory, we can get the following equations:

$$\vec{F}_{\text{crump}} = \text{mass} \cdot \vec{a}, \quad (7)$$

$$|AB| = \frac{1}{2} \vec{a} \cdot dt^2, \quad (8)$$

where  $\text{mass}$  is the mass of the area of thin-shell objects mapped onto the control vertex,  $\vec{a}$  is the acceleration and  $dt$  is the time step. By the rotation amount  $D$ , we can compute  $|AB|$ . Then  $\vec{F}_{\text{crump}}$  can be obtained by (7) and (8).

Besides the crumple force, the other forces mapped onto the control vertexes also include gravity, buoyant force and inner forces which are caused by the deformation. We express them by the following equations:

$$\vec{F}(i, j) = \vec{F}_{\text{inner}}(i, j) + \vec{F}_{\text{outer}}(i, j), \quad (9)$$

$$\vec{F}_{\text{inner}}(i, j) = \vec{F}_{\text{crump}}(i, j) + \vec{F}_{\text{spring}}(i, j), \quad (10)$$

$$\vec{F}_{\text{outer}}(i, j) = \vec{F}_{\text{buoyant}}(i, j) + \vec{F}_{\text{gravity}} + \vec{F}_{\text{damping}}(i, j), \quad (11)$$

where  $\vec{F}_{\text{outer}}(i, j)$  denotes the outer force,  $\vec{F}_{\text{buoyant}}(i, j)$  is the buoyant force,  $\vec{F}_{\text{gravity}}$  is the gravity,  $R$  is the set of all

couples linking to vertex  $(i, j)$ .  $\vec{F}_{\text{spring}}(i, j)$  denotes the tension linking  $(i, j)$  and  $(k, l)$ .  $\vec{F}_{\text{damping}}(i, j)$  is the damping force which can be expressed as the following:

$$\vec{F}_{\text{damping}}(i, j) = -C_{\text{damping}} \cdot \vec{V}(i, j), \quad (12)$$

where  $C_{\text{damping}}$  is a damping coefficient, and  $\vec{V}(i, j)$  is the velocity of  $(i, j)$ .

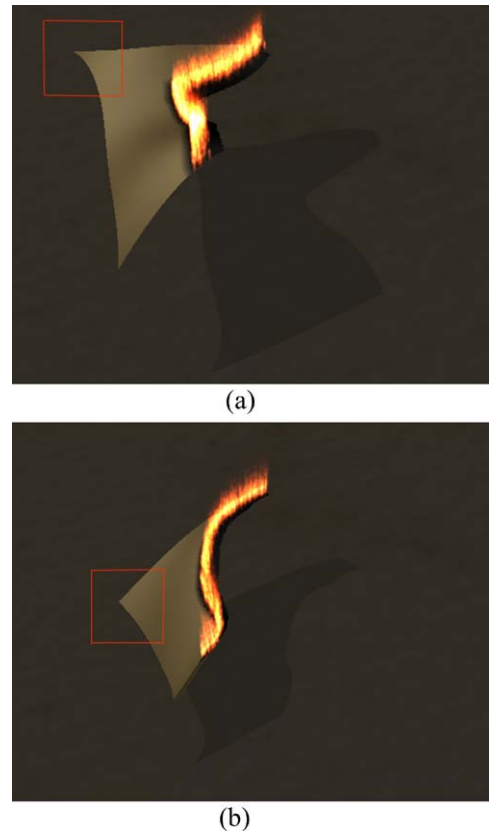
$\vec{F}_{\text{spring}}(i, j)$  is composed of structure force, shear force and flexible force which are related to the structure coefficient  $K_{\text{structure}}$ , the shear coefficient  $K_{\text{shear}}$  and the flexible coefficient  $K_{\text{flex}}$ , respectively [22].

Now we exert the above forces on the control vertexes to change their positions. Let  $(i, j)$  represent a control vertex,  $\vec{V}_0(i, j)$ , and  $\vec{\text{Pos}}_0(i, j)$  denote the initial velocity and position of this control vertex, respectively. After a time step, the new velocity  $\vec{V}(i, j)$  and position  $\vec{\text{Pos}}(i, j)$  can be updated by the following equations:

$$\vec{F}(i, j) = \text{mass}(i, j) \cdot \vec{a}(i, j), \quad (13)$$

$$\vec{V}(i, j) = \vec{V}_0(i, j) + \vec{a}(i, j) \cdot dt, \quad (14)$$

$$\vec{\text{Pos}}(i, j) = \vec{\text{Pos}}_0(i, j) + \vec{V}(i, j) \cdot dt + \frac{1}{2} \cdot \vec{a}(i, j) \cdot dt^2, \quad (15)$$



**Fig. 6** A contrast of the deformation effect between our method and that of [17]

**Table 1** The parameters and their values in our experiment

Figure	mass	$V_0$	$C_p$	$C_{\text{damping}}$	$K_{\text{structure}}$	$K_{\text{shear}}$	$K_{\text{flex}}$
Fig. 8	0.45	3.5	1.0	1.7	0.05	0.05	0.05
Fig. 9	1.45	4.0	0.8	2.5	0.3	0.01	0.04
Fig. 10	10.45	3.5	1.0	1.7	0.5	0.01	0.001
Fig. 11	6.45	4.0	0.8	2.5	0.01	0.01	0.01
Fig. 12	1.45	4.0	0.8	2.5	0.3	0.01	0.04

where  $\vec{F}(i, j)$  is the force exerted on the control vertex  $(i, j)$ ,  $\vec{a}(i, j)$  denotes the acceleration of the control vertex  $(i, j)$ .

Figure 6 shows the contrast of the deformation effect of a piece of paper when burning, between our method and that of [17]. Melek et al. used the amount of deformation  $D$  of each control vertex to estimate the position of all the other control vertexes affected by it. In fact, the deformation occurring in one area only affects its neighboring areas, and the deformation of other areas is mainly caused by the inner forces. In our method, we exert  $D$  of each control vertex on a couple of control vertexes which is adjacent to it and perpendicular to the chosen rotation axis. In Fig. 6(a) we can see that Melek's method may lead to overstretching of the corner areas, like the part in the red rectangle. Our method can solve the overstretching problem very well, as is shown in Fig. 6(b).

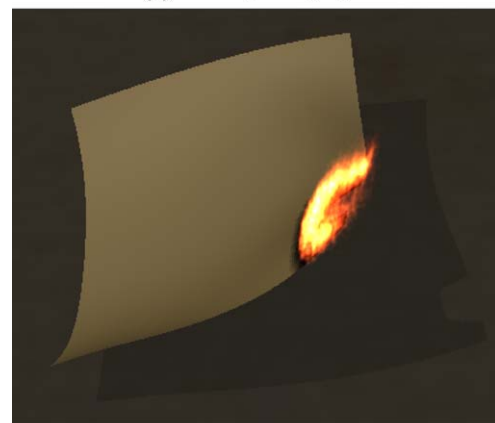
## 5 Implementation and rendering results

Based on the above model, we have generated various burning scenes of thin-shell objects for different kinds of paper and cloth. All simulations were run on a Pentium IV with 3.0 G RAM, nVidia 7800 display card. The thin-shell object is divided into  $128 \times 128$  cells. The rendering rates are about 10 frames per second. The meaningful parameters of the different examples are gathered in Table 1.

Figure 7 shows the contrast between our simulation results and a real photograph. Figure 8 illustrates burning scenes of a piece of paper. We can see from Fig. 8 that a paper is crumpling and bending while the fire is spreading. On the other hand, the deformation of a paper also affects the spreading route of the fire. Figure 9 shows the simulation sequence of cloth burning. As the density of cloth is larger than that of paper, we can see from Fig. 9 that the cloth is sinking when burning. The deformation of cloth is also different from that of a paper, which is determined by their structure and physical properties. By changing some parameters, we can also generate burning scenes of different thin-shell objects. Figures 10 and 11 show burning scenes of thin-shell objects made of different materials. Figure 10



(a) Real photograph

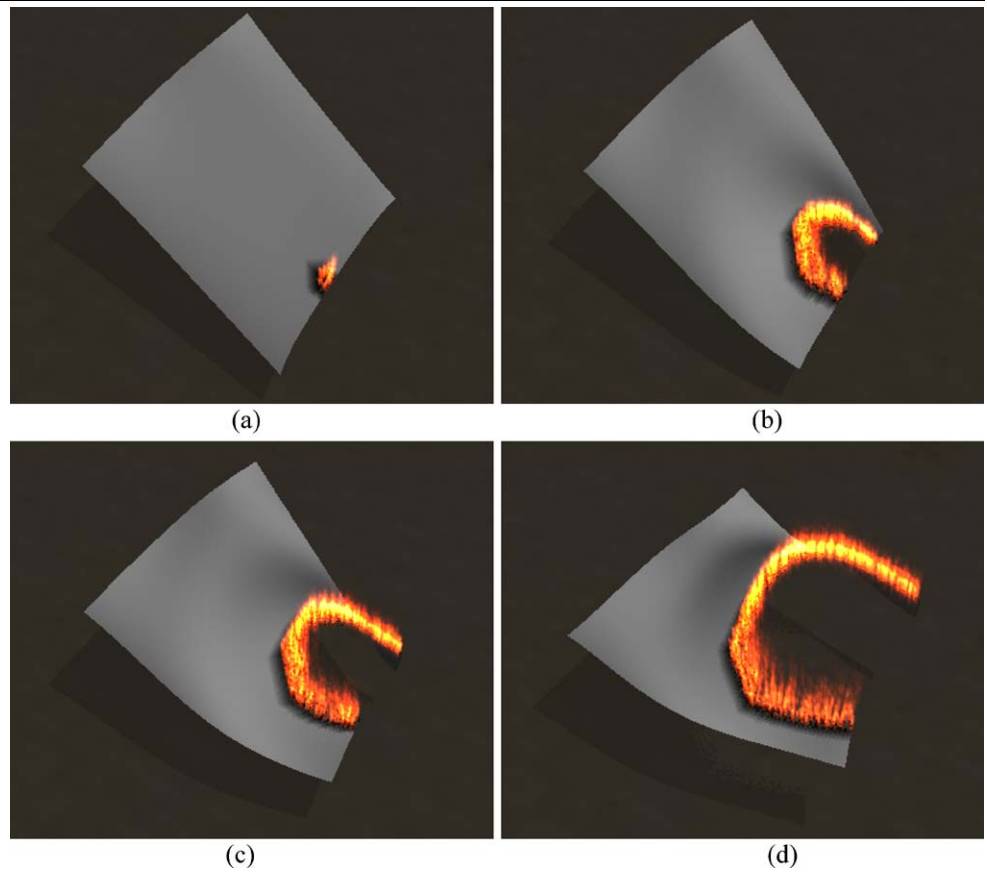


(b) Simulation result of our method

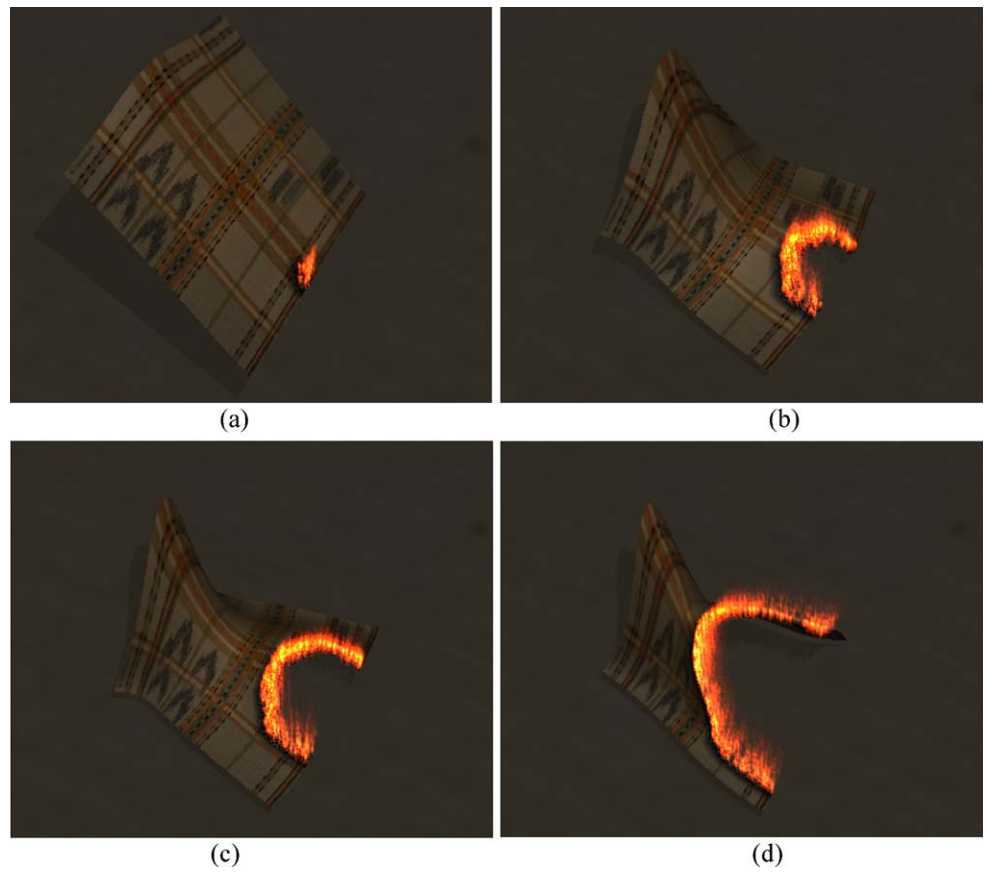
**Fig. 7** A contrast between our simulation results and the real photograph

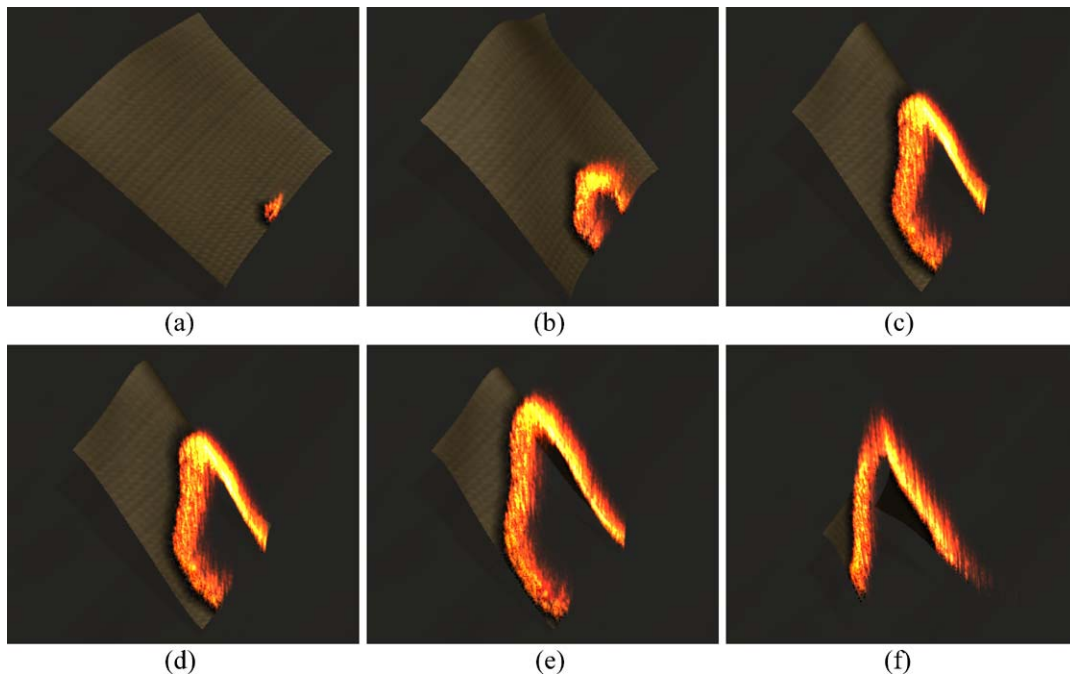
depicts the burning scene of a piece of paper with hard material; Fig. 11, the burning sequence of a piece of cloth made of silk. Figure 12 shows a burning scene of a curtain under the action of wind forces. The wind is blowing from the back of the curtain. The wind forces are considered as the outer force in (9). When the wind forces are mapped onto each control vertex, we can simulate the motion of cloth under the wind field. The method for calculating wind forces can be found in [23, 24].

**Fig. 8** Simulation sequences of paper burning

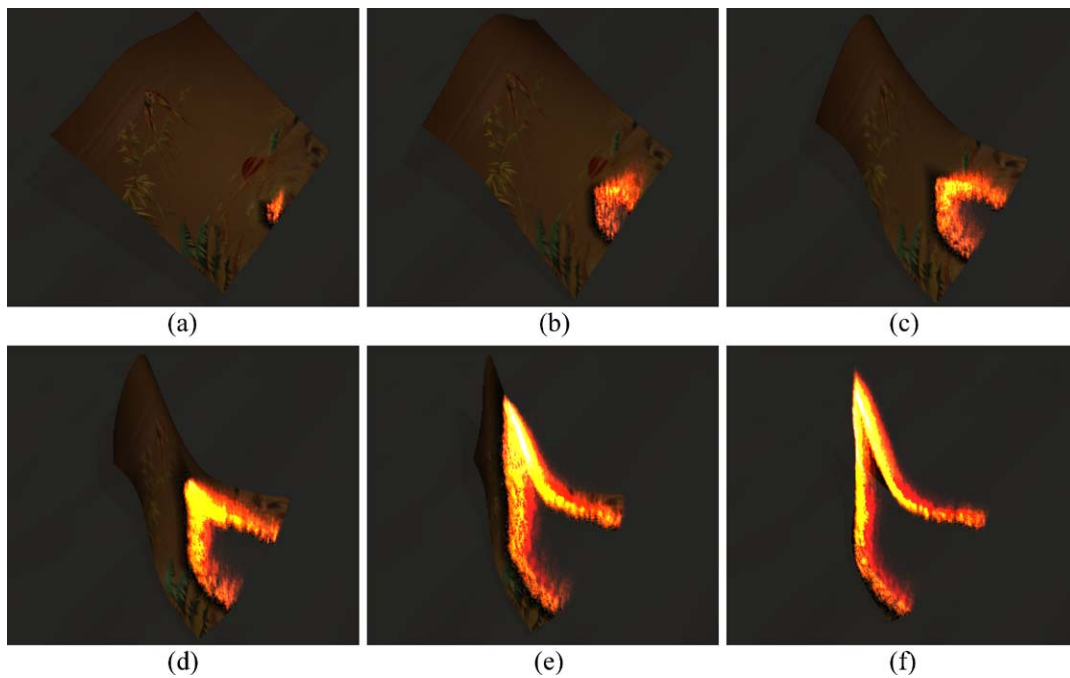


**Fig. 9** Simulation sequences of cloth burning





**Fig. 10** Simulation sequences of hard paper burning



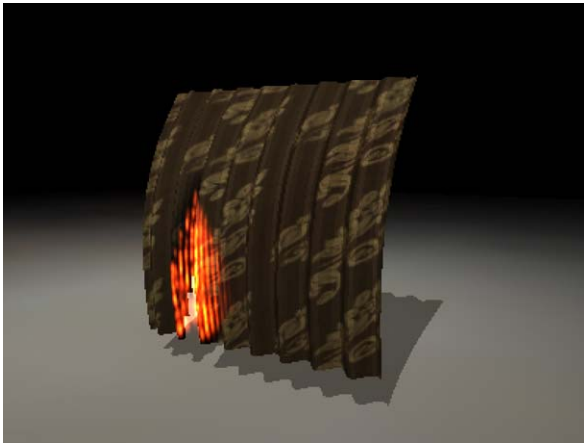
**Fig. 11** Simulation sequences of silk cloth burning

## 6 Conclusions and future works

We have proposed a framework for simulating burning phenomena of thin-shell objects such as paper, cloth, etc. As far as we know, this is the first attempt to simulate burning phenomena of thin-shell objects under a unified

framework. Our method can simulate the fire spreading and the deformation of thin-shell objects realistically. The simulation framework is easy to implement. By inputting different parameters, burning scenes of thin-shell objects made of different materials can be generated automatically.





**Fig. 12** Burning of a piece of curtain under the action of wind force

Since the method adopts a FFD-based method to simulate the deformation of thin-shell objects, some details of crumpling may be lost. It should be possible to integrate the recent mesh deformation method such as Laplacian coordinates, etc., to our framework.

Another possible enhancement lies in the acceleration of the rendering. By using modern GPUs, the rendering will be further accelerated. Thus, simulation of more complex burning phenomena such as fire disaster may be possible, which is also a part of our future work. Our future work will also include simulation of coupling between other fluids such as water, and dynamic objects.

**Acknowledgements** This work was supported by Natural Science Foundation of China under Grant No. 60803047, the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 200800561045, the Open Project Program of State Key Lab of Computer Science, Institute of Software, Chinese Academy of Sciences under Grant No. SYSKF0803, the Open Project Program of the State Key Lab of CAD&CG, Zhejiang University under Grant No. A0907, the Tianjin University Foundation for Fostering Young Teachers under Grant No. TJU-YFF-08B32. We are also grateful to the reviewers for their precise comments which have improved the quality of this paper and will benefit our future work.

## References

- Enright, D., Marschner, S., Fedkiw, R.: Animation and rendering of complex water surfaces. In: *Proceedings of SIGGRAPH*, pp. 736–744 (2001)
- Fedkiw, R., Stam, J., Jensen, H.W.: Visual simulation of smoke. In: *Proceedings of SIGGRAPH*, pp. 15–22 (2001)
- Foster, N., Fedkiw, R.: Practical animation of liquids. In: *Proceedings of SIGGRAPH*, pp. 23–30 (2001)
- Nguyen, D.Q., Fedkiw, R., Jensen, H.W.: Physically based modeling and animation of fire. In: *Proceedings of SIGGRAPH*, pp. 721–728 (2002)
- Yngve, G.D., O'Brien, J.F., Hodgins, J.K.: Animating explosions. In: *Proceedings of SIGGRAPH*, pp. 29–36 (2000)
- Carlson, M., Mucha, P.J., Turk, G.: Rigid fluid: Animating the interplay between rigid bodies and fluid. In: *Proceedings of SIGGRAPH*, pp. 377–384 (2004)
- Genevaux, O., Habibi, A., Dischler, J.M.: Simulating fluid-solid interaction. In: *Proceedings of Graphics Interface*, pp. 31–38 (2003)
- Liu, Y., Liu, X., Wu, E.: Real-Time 3D fluid simulation on GPU with complex obstacles. In: *Proceedings of Pacific Graphics*, pp. 247–256 (2004)
- Reeves, W.T.: Particle systems—a technique for modeling a class of fuzzy objects. In: *Proceedings of SIGGRAPH*, pp. 359–376 (1983)
- Pakeshi, A., Yuz, K., Nakajima, M.: Generating 2 dimensional flame images in computer graphics. *IEICE Trans.* **74**(2), 457–462 (1991)
- Nguyen, D., Fedkiw, R., Jensen, H.W.: Physically based modeling and animation of fire. In: *Proceedings of SIGGRAPH*, pp. 721–728 (2002)
- Hong, J., Shinar, T., Fedkiw, R.: Wrinkled flames and cellular patterns. In: *Proceedings of SIGGRAPH*, pp. 47.1–47.6 (2007)
- Yngve, G.D., O'Brien, J.F., Hodgins, J.K.: Animating explosions. In: *Proceedings of SIGGRAPH*, pp. 29–36 (2000)
- Feldman, B.E., O'Brien, J.F., Arikan, O.: Animating suspended particle explosions. *ACM Trans. Graph.* **22**(3), 708–715 (2003)
- Losasso, F., Irving, G., Guendelman, E., et al.: Melting and burning solids into liquids and gases. *IEEE Trans. Vis. Comput. Graph.* **12**(3), 343–352 (2006)
- Melek, Z., Keyser, J.: Interactive simulation of burning objects. In: *Proceedings of Pacific Graphics*, pp. 462–466 (2003)
- Melek, Z., Keyser, J.: Bending burning matches and crumpling burning paper. In: *Proceedings of SIGGRAPH Poster* (2006)
- 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)
- Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. *Comput. Graph.* **20**(4), 151–160 (1986)
- Terzopoulos, D., Fleischer, K.: Deformable models. *Vis. Comput.* **4**(6), 306–331 (1988)
- Baraff, D., Witkin, A.: Large steps in cloth simulation. In: *Proceedings of SIGGRAPH*, pp. 43–54 (1998)
- Provot, X.: Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In: *Proceedings of Graphics Interface*, pp. 141–155 (1995)
- Gay, K.L., Ling, L., Damodaran, M.: A quasi-steady force model for animating cloth motion. In: *Proceedings of IFIP International Conference on Computer Graphics*, pp. 357–363 (1993)
- Ling, L., Damodaran, M., Gay, K.L.: A model for animating cloth motion in air flow. *Comput. Graph.* **20**(1), 137–156 (1996)



**Shiguang Liu** is an assistant professor at School of Computer Science and Technology, Tianjin University, P.R. China. He graduated from Zhejiang University and received a Ph.D. from State Key Lab of CAD & CG in 2007. His research interests include realistic image synthesis, computer animation, virtual reality, etc.



**Qiguang Liu** is an undergraduate student at School of Computer Software, Tianjin University, P.R. China. His research interest is fluid simulation.



**Jizhou Sun** is Professor at School of Computer Science and Technology, Tianjin University, P.R. China. His research interests include virtual reality, network and parallel computing and image processing.



**Tai An** is an undergraduate student at School of Computer Science and Technology, Tianjin University, P.R. China. His research interest is interactive fire simulation.



**Qunsheng Peng** is Professor in the State Key Lab of CAD&CG at Zhejiang University. His research interests include realistic image synthesis, virtual reality, infrared image synthesis, point-based rendering, scientific visualization and biological calculation, etc. He graduated from Beijing Mechanical College in 1970 and received a Ph.D. from the Department of Computing Studies, University of East Anglia, in 1983. He is currently the Vice Chairman of the Academic Committee, State Key Lab of CAD&CG,

Zhejiang University and is serving as a member of the editorial boards of several Chinese journals.