Interactive Control of Fire Simulation based on Computational Fluid Dynamics

Keisuke Mizutani[1], Yoshinori Dobashi[2] and Tsuyoshi Yamamoto[1]
[1]*Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan*
[2]*Graduate School of Information Science and Technology, Hokkaido University / JST CREST, Sapporo, Japan*

Keywords: Fire simulation, Control, Fluid Dynamics, Interactive Editing.

Abstract: Visual simulation of fire plays an important role in many applications, such as movies and computer games. In these applications, artists are often requested to synthesize realistic fire with a particular behavior. This paper presents two methods in order to help artists meet such requirements. First, we propose a method for controlling fire simulation by extending a previous method for smoke simulation. Controlling fire simulation with the previous method is difficult because of strong buoyancy forces caused by high-temperature. To address this problem, our method locally adjusts magnitudes of control forces. Second, we present an interactive editing method for external force field. The user can interactively design the shape of fire by placing a set of control points. Our method generates a force field to form the shape of the fire. Experimental results show that our method can control the fire into an arbitrary shape specified by the user.

1 INTRODUCTION

In computer graphics, many methods have been proposed for simulating natural phenomena such as smoke, fire, water, and so on. Fluid animations represented as various shapes are used in movies and TV games. In order to enhance reality of these animations, controlling motions of fluids based on numerical fluid analysis is effective. Several methods have been proposed for controlling the fluid simulation (Hong and Kim, 2004) (McNamara et al., 2004) (Shi and Yu, 2005). In previous methods, the motions of smoke or water are controlled into user-specified shapes using external forces.

Visual simulation of fire is important in creating synthetic images of in a scene, such as a burning house and a dragon breathing out fire. Computational fluid dynamics (CFD) is effective for synthesizing realistic fire (Nguyen et al., 2002) (Hong et al., 2007) (Horvath and Geiger, 2009). Although methods using CFD can generate realistic fire, there are many physical parameters that influence the motion of simulated fire. Therefore, animators usually attempt to create the desired fire motion by repeating fluid simulation with different parameter settings until satisfactory results are obtained. Moreover, it is almost impossible to adjust the parameters manually so that simulated fire forms the desired shapes.

Several methods have been proposed for procedurally creating desired motion of fire (Lamorlette and Foster, 2002) (Fuller et al., 2007). However, these methods do not use computational fluid dynamics to generate the fluid flow. Therefore, the resulting animation is less realistic than that created using physically-based simulation.

Our purpose in this paper is to control fire simulation based on CFD. A straight forward approach to achieve this goal is to use the previous methods for controlling smoke or water. However, we found that those previous methods could not control fire simulation accurately. The reason is that fire has strong buoyancy force, which is not observed in other phenomena. The strong buoyancy forces interfere with control forces. Therefore, we developed a new method that computes the control force taking into account the buoyancy forces.

We propose two methods to control fire simulation: a target-driven method and an interactive editing of external force field. The first method controls fire simulation by using a potential field computed from a user-specified target shape. This method is an extension of the previous method proposed by Fattal and Lischinski (Fattal and Lischinski, 2004) so that it can handle strong buoyancy forces caused by high temperature of fire. In the second method, the user can interactively design the shape and motion

Mizutani, K., Dobashi, Y. and Yamamoto, T.
Interactive Control of Fire Simulation based on Computational Fluid Dynamics.
DOI: 10.5220/0005746902400245
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 1: GRAPP*, pages 242-247
ISBN: 978-989-758-175-5

of fire by placing a set of control points. External force fields are automatically generated according to the user-specified control points.

Although several methods have been proposed for simulating realistic fire, we employ a relatively simple method for the simulation of fire. We simulate fire as a heated gas generated by combustion. Although the behavior of the simulated fire is slightly less realistic than that by the two-fluid model (Nguyen et al., 2002), our method is far more efficient and is easy to implement. We employ the simple method because our primary purpose is to synthesize the desired shape interactively with plausible appearances. Our method is fully implemented on the GPU and the user can interactively edit the fire animation.

2 RELATED WORK

There are many methods for visual simulation of fire. They are roughly classified into two categories: the procedural approach and physical-based approach.

Methods based on the procedural approach have focused on efficiency and control. Particle systems are the most widely used method. The earliest reported fire model, presented by Reeves(Reeves, 1983), used a particle system to animate fire. This method required a large number of particles to achieve natural visual effects. After this work, some methods have been proposed for procedural modeling of fire animation (Beaudoin et al., 2001)(Lamorlette and Foster, 2002) (Fuller et al., 2007). Although the computational cost for the procedural methods is low and the desired fire shapes are easy to generate, some important visual features, such as vortex motions, cannot be reproduced.

On the other hand, methods based on the physically-based approach can generate realistic visual results with evolving and dynamic fires. Various fire simulation methods have been developed based on incompressible fluid solvers (Nguyen et al., 2002) (Hong et al., 2007) (Horvath and Geiger, 2009). Although these methods have the potential to generate realistic fires, many physical parameters have to be adjusted to obtain the desired results. This indicates that adjusting the parameters manually to synthesize fire with desired shapes is time-consuming or, furthermore, almost impossible.

Some researchers have developed methods for controlling simulation of phenomena related to fluids. Treuille et al.(Treuille et al., 2003) developed a method for keyframe control of smoke simulation. McNamara et al.(McNamara et al., 2004) proposed an adjoint method for controlling smoke and water. Hong et al. (Hong and Kim, 2004) proposed to use a geometric potential field generated from given target shapes to generate control forces. Fattal and Lischinski(Fattal and Lischinski, 2004) developed a target-driven method to control smoke simulation by using some external forces. The basic concept of this method is similar to ours, however, their method is not directly applicapable to fire simulation since it does not take into account the strong buoyancy force. Shi and Yu (Shi and Yu, 2005) proposed a method using a feedback force field and the negative gradient field of geometric potential function for creating the animation of water towards to the rapidly changing targets. Thurey et al (Thürey et al., 2006) introduced control particles for controlling the motion of water. Kim et al. (Kim et al., 2006) proposed a method to advect smoke along a user-specified path. However, these methods do not cover fire simulation. Controllable fires have been an active reserch area in the last few years. Lever and Komura (Lever and Komura, 2012) controlled fire simulation with textured force for generating volumetric patterns similar to input textures on the fire volume. Hong et al. (Hong et al., 2010) introduced a method for controlling blue core for modeling fire propagation along complex curves or surface. Zhang et al. (Zhang et al., 2011) applied high geometric motion constraints to the fire animation. Bangalore and House (Bangalore and House, 2012) proposed artistic control of flames using a set of curves drawn by the user. This method deals with target shapes formed by a set of curves only. However, there are no methods for interactively controlling the motion of fire based on computational fluid dynamics.

3 FIRE SIMULATION

Before describing our method, this section explains the numerical simulation of fire. We simulate fire by using a grid-based approach. The simulation space is subdivided into a grid, and a fire source is placed at an arbitrary position in the simulation space. Velocity $\mathbf{u}$ and temperature T are assigned at each grid point (see Figure 1). Motions of the fire are simulated in the following way. Velocity $\mathbf{u}$ is updated at each time step by solving the following Navier-Stokes equations.

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) - \nabla p + \mathbf{f}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where t is time, p is a fluid pressure, and $\mathbf{f}$ represents any external forces such as gravity or wind. For the fire simulation, temperature and buoyancy forces are calculated by using the previous techniques proposed by Nguyen et al. (Nguyen et al., 2002). Temperature T is calculated by the following equation.

$$\frac{\partial T}{\partial t} = -(\mathbf{u} \cdot \nabla)T - C_r \left(\frac{T - T_{amb}}{T_{max} - T_{amb}} \right)^4, \quad (3)$$

where T_{amb} is ambient temperature, T_{max} is the maximum temperature, and C_r is a cooling constant. As an external force, the buoyancy force is taken into account. The buoyancy force $\mathbf{f}_{buo}$ is given by:

$$\mathbf{f}_{buo} = \kappa_b (T - T_{amb}) \mathbf{y}, \quad (4)$$

where κ_b is the coefficient for the buoyancy and $\mathbf{y}$ is a unit vector pointing upward vertical direction. Temperatures and velocities at the grid points around the fire source are fixed on those of the fire source.

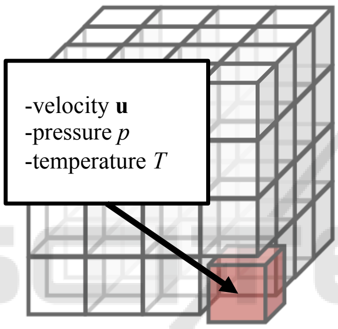

Figure 1: Simulation space.

4 TARGET DRIVEN CONTROL

Our first method to control simulation of fire uses a driving force term (Fattal and Lischinski, 2004) as the external force. Figure 2 shows an overview of our control method. First, we calculate a target temperature using a target shape specified by the user. Next, we calculate the driving force term by using a gradient of the target temperature and then we apply the driving force to the fire simulation.

However, controlling the fire simulation by applying the driving force only is difficult because of the strong buoyancy forces. Therefore, we locally adjust coefficients of the driving force term by calculating a cumulative distribution (see Section 4.2), representing accumulated differences between the shape of the simulated fire and the target temperature. The shape of the simulated fire is calculated by binarizing the temperature distribution of fire. Then, we locally adjust coefficients for the driving force term in proportion to the cumulative distribution. In the following subsections, we describe the details of each process.

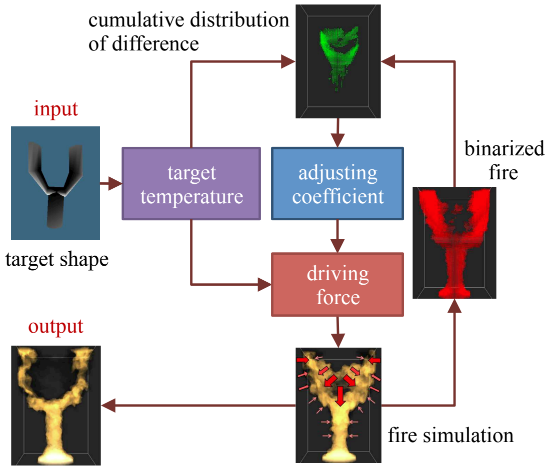

Figure 2: Target driven control method.

4.1 Control Force

In this section, we describe a calculation of the driving force term $\mathbf{F}$. The purpose of this force is to cause the fluid to advect the current fire temperature T towards the target temperature T^*. The target temperature T^* has a binary value: 0 or 1. The evolution of fire simulation is governed by the Navier-Stokes equations. We add the driving force term $\mathbf{F}$ to the external force term $\mathbf{f}$ in the Navier-Stokes equation (Equation (1)). The driving force term is defined by the following equation.

$$\mathbf{F}(T, T^*) = \nu_f \tilde{T} \frac{\nabla \tilde{T}^*}{\tilde{T}^*}, \quad (5)$$

where ν_f is an user specified coefficient, $\tilde{T}$ and $\tilde{T}^*$ are obtained by convolving T and T^* with a Gaussian kernel respectively. The gradient of the target temperature ∇T^* always points uphill towards higher concentrations of T^*. This means that the gradient ∇T^* points in the direction of flow to the target. However, T^* might be uniform in some regions, causing $\nabla T^* = 0$ there. In order to avoid this problem, we substitute $\nabla \tilde{T}^*$ for ∇T^*. $\tilde{T}^*$ ensures $\nabla T^* \neq 0$ everywhere. In addition, the magnitude of $\nabla \tilde{T}^*$ is very small in the region away from the target shape. Therefore, we use a normalized gradient of the target temperature, which is obtained by dividing the gradient $\nabla \tilde{T}^*$ by $\tilde{T}^*$. The magnitude of the driving force is proportional to the target temperature $\tilde{T}$. Consequently, this prevents the driving force from being applied in the areas where the temperature is zero.

4.2 Adjusting the Coefficient

As we describe before, the buoyancy forces interfere with the driving force and the simulated fire exceeds the target shape. In order to address this problem, we use an adaptive force coefficient $\nu_f(\mathbf{x})$ at each grid point $\mathbf{x}$, instead of the constant ν_f in the entire simulation space. The adaptive coefficient $\nu_f(\mathbf{x})$ is calculated by using the cumulative distribution, as shown in Figure 3, and given by the following equation.

$$\nu_f^{(n)}(\mathbf{x}) = k_f T_{cu}^{(n)}(\mathbf{x}) \nu_f^{(n-1)}(\mathbf{x}), \qquad (6)$$

where n is a time step, k_f is a control parameter specified by the user and $T_{cu}(\mathbf{x})$ is the cumulative distribution of differences. This is expressed by the following equations.

$$T_{cu}^{(n)}(\mathbf{x}) = \{T_{bi}(\mathbf{x}) - T^*(\mathbf{x})\} + k_c T_{cu}^{(n-1)}, (\mathbf{x}) \qquad (7)$$

$$T_{bi}(\mathbf{x}) = \begin{cases} 1 & (T(\mathbf{x}) \geq T_{th}) \\ 0 & (T(\mathbf{x}) < T_{th}) \end{cases} \qquad (8)$$

where k_c is a coefficient to control the degree of accumulation and ρ_{bi} is a distribution obtained by binarizing fire temperature with a threshold temperature T_{th}. In this paper, T_{th} is experimentally determined. $T_{bi}(\mathbf{x}) - T^*(\mathbf{x})$ is set to zero when it is negative. k_c is specified between 0 and 1. The force coefficient $\nu_f(\mathbf{x})$ is proportional to the cumulative distribution. Increasing k_c results in stronger driving forces in regions where the cumulative difference is large. By using the above method, our method can control motions of the fire so that the fire shape matches the user-specified target shape.

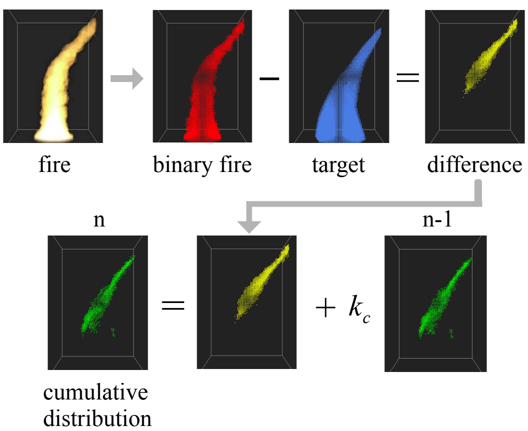

Figure 3: Process of calculating cumulative distributions of difference.

5 INTERACTIVE EDITING OF FORCE FIELD

This section describes the interactive editing of the force fields to generate the desired flow and fire shape. In this method, the user can design the desired shape and motion of fire by interactively placing a set of control points. When the user places a control point on the screen, it is projected onto a plane that is perpendicular to the xy components of the view direction (see Figures 4). Let us denote the projected position by $\mathbf{P}_s$. The user drags the control point on this plane to an arbitrary position, $\mathbf{P}_e$. A force field is then generated around the control point so that the temperature at the dragged position $\mathbf{P}_e$ becomes an user-specified target temperature. The force field is generated inside a cylinder with the user-specified radius r and height l. The top and bottom centers of the cylinder are $\mathbf{P}_e$ and $\mathbf{P}_s$, respectively (see Figure 4). The force field is used as an external force in the Navier-Stokes equation (Equation (1)) and is defined by:

$$\mathbf{F}_e = c_f(\mathbf{P}_e - \mathbf{P}_s), \qquad (9)$$

where c_f is a coefficient for adjusting the magnitude of the force field. In our method, the coefficient is controlled by using PID control mechanism so that the temperature at the end point becomes the user-specified target temperature. In this case, we update the coefficient c_f at each time step according to the following equation.

$$c_f = K_P e + K_I \int_0^T e dt + K_D \frac{de}{dt}, \qquad (10)$$

$$e = T_{target} - T_{end}, \qquad (11)$$

where e is the temperature difference between the temperature T_{end} at $\mathbf{P}_e$ and the target temperature T_{target}. The target temperature is set to the threshold temperature that defines boundaries between smoke and fire. Let us now explain the meaning of Equation (10). The first term on the right updates c_f in proportion to the current difference. This term is called the proportional controller and K_P is called proportional gain. The second integral term updates c_f in proportion to both the magnitude of the difference and the duration of the difference. The integral controller is sum of the instantaneous difference over time and gives the accumulated offset that should have been corrected previously. T is the duration for the accumulation. The accumulated difference is then multiplied by the integral gain K_I. The contribution of the third term is proportional to the derivative of the process difference. This term is calculated by determining the slope of the difference over time and multiplying this rate of change by the derivative gain K_D. Derivative action predicts system behavior and thus improves settling time and stability of the system. The control parameters K_P, K_I and K_D are determined experimentally.

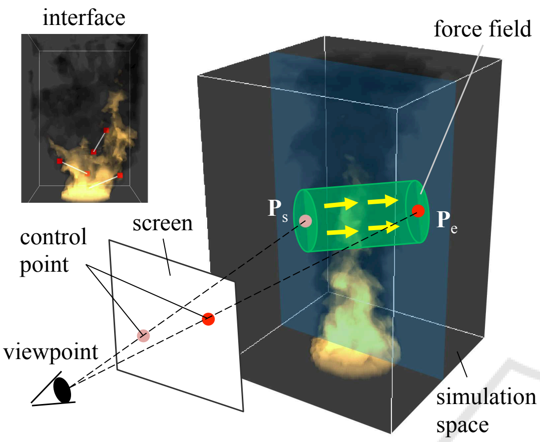

Figure 4: Overview of our editing system.

6 RESULT

This section shows some examples created by our method. We used a desktop PC with Intel Core i7 (8GB of RAM) and NVIDIA GeForce GTX 780 as a GPU to compute all the examples shown in this section. Please refer to the accompanying video file for animations of the examples shown in this section.

Fig. 5 shows examples of fire simulation controlled by our method. The simulation space is subdivided into $64 \times 64 \times 96$ grid points. The computation time for each step took 0.03 seconds. The target shapes are the Y-shape and the spiral-shape in Figs. 5(a)(b) and Figs. 5(c)(d), respectively. The target shape is shown with blue dots. Fig. 5(a)(c) are examples without our method for automatically adjusting the coefficient of the driving force and Figs. 5(b)(d) are examples with our method. As shown in this figure, the fire shapes in Figs. 5(b)(d) are much closer to the target shapes than those in Figs. 5(a)(c). In both examples, the cumulative coefficient k_c is set to 0.95. The threshold temperature is 0.4 that is chosen experimentally.

Next, Figs. 6 and 7 show examples of fire simulation edited by our method. The simulation space is subdivided into $128 \times 192 \times 128$ grid points. The computation time for each step took 0.1 seconds. Figs. 6(a) and 7(a) are original fire simulations and Figs. 6(b) and 7(b) are examples edited by using our editing method. In Fig. 6, the shape of fire from the Dragon is edited so that the shape becomes wider. Fig. 7 shows an example of a burning house. By using our editing method, we can make fire wrapping around the house like Fig. 7(b) without computing interactions between fire and the house.

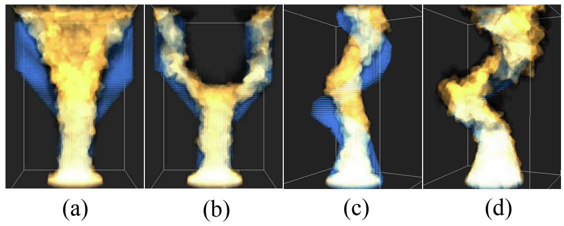

Figure 5: Comparison of results with and without our method. In (a)(c), our method is not used while (b)(d) use our method.

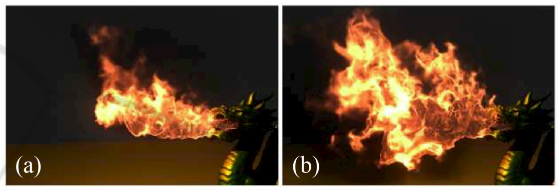

Figure 6: Fire from dragon.

Figure 7: Burning house.

7 CONCLUSION

We have proposed two methods to control fire simulation based on computational fluid dynamics: the target driven method and the interactive editing of external force field. In the first method, we locally adjust a coefficient of the driving force term by calculating the cumulative distribution of differences between the shape of the simulated fire and the target shape. Our method can control the simulated fire into an arbitrary shape specified by the user. In the second method, the user can interactively design the shape and motion of fire by a set of control points. In order to generate the desired flow and fire shape, our method computes external force fields automatically generated according to the user-specified control points. Our method provides a simple way to generate realistic fires with desired shapes and motions.

In our future work, it might be interesting to control a fire source in order to generate additional effects. In addition, we plan to extend our method to other types of fluid simulations such as particle-based methods.

REFERENCES

Bangalore, A. and House, D. H. (2012). A technique for art direction of physically based fire simulation. In *Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, CAe '12, pages 45–54, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Beaudoin, P., Paquet, S., and Poulin, P. (2001). Realistic and controllable fire simulation. In *Proceedings of Graphics Interface 2001*, GI '01, pages 159–166, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.

Fattal, R. and Lischinski, D. (2004). Target-driven smoke animation. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 441–448, New York, NY, USA. ACM.

Fuller, A. R., Krishnan, H., Mahrous, K., Hamann, B., and Joy, K. I. (2007). Real-time procedural volumetric fire. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, I3D '07, pages 175–180, New York, NY, USA. ACM.

Hong, J.-M. and Kim, C.-H. (2004). Controlling fluid animation with geometric potential: Research articles. *Comput. Animat. Virtual Worlds*, 15(3-4):147–157.

Hong, J.-M., Shinar, T., and Fedkiw, R. (2007). Wrinkled flames and cellular patterns. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, New York, NY, USA. ACM.

Hong, Y., Zhu, D., Qiu, X., and Wang, Z. (2010). Geometry-based control of fire simulation. *Vis. Comput.*, 26(9):1217–1228.

Horvath, C. and Geiger, W. (2009). Directable, high-resolution simulation of fire on the gpu. In *ACM SIGGRAPH 2009 Papers*, SIGGRAPH '09, pages 41:1–41:8, New York, NY, USA. ACM.

Kim, Y., Machiraju, R., and Thompson, D. (2006). Path-based control of smoke simulations. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 33–42, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Lamorlette, A. and Foster, N. (2002). Structural modeling of flames for a production environment. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 729–735, New York, NY, USA. ACM.

Lever, J. and Komura, T. (2012). Real-time controllable fire using textured forces. *The Visual Computer*, 28(6-8):691–700.

McNamara, A., Treuille, A., Popović, Z., and Stam, J. (2004). Fluid control using the adjoint method. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, pages 449–456, New York, NY, USA. ACM.

Nguyen, D. Q., Fedkiw, R., and Jensen, H. W. (2002). Physically based modeling and animation of fire. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '02, pages 721–728, New York, NY, USA. ACM.

Reeves, W. T. (1983). Particle systems—a technique for modeling a class of fuzzy objects. *ACM Trans. Graph.*, 2(2):91–108.

Shi, L. and Yu, Y. (2005). Taming liquids for rapidly changing targets. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, pages 229–236, New York, NY, USA. ACM.

Thürey, N., Keiser, R., Pauly, M., and Rüde, U. (2006). Detail-preserving fluid control. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 7–12, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Treuille, A., McNamara, A., Popović, Z., and Stam, J. (2003). Keyframe control of smoke simulations. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, pages 716–723, New York, NY, USA. ACM.

Zhang, Y., Correa, C. D., and Ma, K.-L. (2011). Graph-based fire synthesis. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pages 187–194, New York, NY, USA. ACM.