A parametric model for real-time flickering fire

Abstract

Fire is a complex natural phenomena involving chemical reactions and turbulent flow of hot gaseous products. Significant computer graphics research is devoted to modelling the turbulent dynamics of fire; however flicker which results from the chemical nature of fire has not been previously modelled. In this paper we present a technique that addresses both modelling of dynamics and chemical aspects of fire. We use a stochastic Lagrangian approach to model the dynamics. We formulate a simplified model of thermochemistry that mimics the behavior of models in combustion studies to capture the chemical aspects. The model is parameterized to enable control of flicker rate, flame height and number of flame brushes in the simulated fire. We exploit the programmability of graphics hardware for real-time rendering. The capabilities of the model are demonstrated via representative animations.

Keywords: stochastic techniques, Lagrangian approach, chemical composition, programmable graphics card

Introduction

Fire simulation and visualization is required in a number of application areas of computer graphics including the special effects industry, virtual and mixed reality for training, architectural design evaluation and the entertainment/gaming industries. These simulations should be performed in real-time in the cases of virtual/mixed reality and interactive gaming. Also, when fire is used as an element in the design of an environment, it is desirable to have control on its appearance. In this paper we focus on creating a real-time fire model with parameters that enable defining the properties that affect its appearance.

Fire is a phenomenon created by glowing combustion products in turbulent motion. The key aspects that have to be modelled while simulating fire are turbulent dynamics and the chemical reaction accompanying it.

The turbulent dynamics of fire results in rapid motion of the gaseous products of combustion in various directions. When a grid based approach is used to model such a flow several design issues arise including size and placement of the grid in the environment and choice of grid resolution. We overcome these problems by modeling the dynamics using the Lagrangian approach, which is grid-free. Another significant problem with models of turbulent flow is that the computations become unstable after carrying out a series of successive time steps of simulation. This problem is also overcome by using the stochastic Lagrangian approach for modelling the dynamics [1].

The chemical reaction (oxidation) accompanying the turbulent motion can be broadly classified into premixed and diffusion type. The premixed oxidation results in premixed flames where the fuel and oxidizer are mixed before burning. Such flames are steady as the reaction products are generated at a constant rate. The diffusion oxidation results in diffusion flames; here the fuel has to diffuse into the oxidizer before burning. Diffusion flames constitute most fires that result from objects/fuel burning in air, as there is no premixing of the fuel with the oxidizer before burning. Diffusion is non-uniform along the fuel/oxidizer interface, causing intermittency in the chemical reaction that is dependent on availability of oxidizer. Diffusion flames flicker due to local and global extinction resulting from non-availability of oxidizer.

Flicker in fire is an important visual feature. The work of Lamorlette and Foster [2] identify an intermittent flame region in fire, and computer vision studies by Phillips et al. [3] use flicker of fire as a property to recognize existence of fire in a scene. Local extinction behavior is captured in the turbulent motion of the gaseous products; however global extinction has to be explicitly modelled. Unfortunately there is no model for flicker from global extinction in the existing computer graphics models for fire. The work reported here explicitly addresses this problem. Our approach enables us to model two important phenomena in fire, namely, flicker caused by global extinction, and “flame brushes,” which are regions of greater brightness in flames that occur where there is more chemical reaction. We exploit knowledge from the Euclidian Minimum Spanning Tree (EMST) mixing approach [4], which models the evolution of chemical composition of combustion products, to develop a simplified solution that results in a visually convincing global extinction behavior in real-time. We design the model such that we can parametrically control the flame flicker rate, flame height and number of flame brushes.

The rest of the paper is organized as follows: We describe previous work in the following section. We then present our model of fire. This has two main aspects the stochastic Lagrangian model for the dynamics and the model that represents the chemical evolution of composition. The rendering of the model using programmable graphics hardware is then detailed. This is followed by a discussion of results demonstrating the capabilities of the technique. We present our conclusions in the final section.

Previous work

The existing literature that addresses fire simulation in graphics broadly falls into the categories of physics based and non-physics based techniques.

Physics based techniques include the recent work of Nguyen et al. [5], which achieves impressive simulations of fire. Their model uses Euler equations in conjunction with a level set model for tracking the core of the flame and a model of thermal buoyancy. The model is excellent for the thin flame fires where the fuel oxidizer mixing region can be represented as a surface. Such flames do not exhibit much flickering since there is continuous combustion of the fuel at the fuel/oxidizer interface. The model of smoke propagation developed by Fedkiw et al. [6] is employed to represent the smoke generated by the fire. These results in turn build on those of Stam [7]. Stam and Fiume [8] model fire as a diffusion process with suitable sources and sinks; they evolve and warp fluid blobs based on the equations and a representation of a wind field. A Lattice-Boltzmann approach to model the dynamics of fire is presented by Wei et al [9]. They achieve real-time performance by mapping the computation of dynamics onto graphics hardware, and by rendering with texture splats. Bukowski and Sequin [10] describe a technique for simulating fire in architectural walk-throughs to enable evaluation of building design. Rushmeier et al. [11] describe a technique for rendering of pool fires and employ them to study radiative effects. The applications of modelling and visualization of fire to gain insight into their properties and analyze fire hazards is described in Forney et al. [12].

Non-Physics based techniques include the recent work of Lamorlette and Foster [2] that adopts this approach in the structural modelling of natural flames. They model fire as a collection of flames. Each flame consists of an evolving spline in a wind field [13] and an associated shape profile. They use the results of Foster and Metaxas [14] that solved the Navier-Stokes equations with a suitable model of thermal buoyancy for modelling the evolution of the gaseous products that result from the flames. Previous work by Inakage [15] created static images of flames. Other approaches that use flames as primitives to model fire include the work of Perry and Picard [16] and Beaudoin and Paquet [17]. While the simulation of fire is not physics based in [16], they provide a physics based model for the spread of fire in the same work. This approach is modified in [17] to model the spread of fire on objects represented by meshes. Devlin and Chalmers [18] present results on recreating the illumination effects of lighting by various kinds of oil lamps. They consider the spectral measurements from the light emitted by the flames for rendering.

For completeness of references we mention here related work on modelling explosions and modelling of motion of gaseous volumes. These include the seminal work of Reeves [19] that introduced the use of particle systems, and the more recent works of Yngve et al. [20] and Feldman et al. [21] that model explosions. The former develops a physically accurate model considering shock waves while the latter address the problem with the goal of visually convincing appearance. A technique for modelling huge volumes of smoke is presented in the work of Rasmussen et al. [22] which uses a combination of physics based and non-physics based approaches to achieve the desired visual results. In production environments there is often a requirement to control the natural phenomena in order to create the desired appearance in a synthesized scene. In this context Treuille et al. [23] present a technique for simulating smoke that combines a non-physics based approach of control with physics based approaches to simulate the fluid flow. Rushmeier [24] describes issues and applications for rendering participating media and includes a discussions on applications for simulation of fire. Premoze et al. [25] present a physics based technique for modelling gaseous volumes that is based on the large eddy simulation approach. Adabala and Manohar [26] developed a technique of modelling dynamics gaseous volumes using vortex element methods and implemented the computations with particle maps. A technique for simulating three-dimensional planar jets exhibiting turbulent motion with and without chemically reacting conditions is presented by Garrick and Interante [27].

The following section presents our model of fire and the parameters by which its appearance can be controlled. We also discuss the real-time performance attainable through this model. The subsequent section presents the real-time rendering method based on this model.

Model of fire

There are two main aspects of our model:

- Dynamics model
- Chemical composition evolution model

each of these aspects is explained in the following two subsections.

Dynamics model

The flow of hot gaseous products in a diffusion flame can be modelled as an incompressible turbulent flow. The equations that define this flow are the equation for conservation of mass:

$$\nabla \cdot \vec{u} = 0, \tag{1}$$

and the Navier-Stokes equations:

$$\frac{D\vec{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\vec{u} + \vec{F}, \tag{2}$$

where $D\vec{u}/Dt$ is the material derivative $\frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z} = \frac{\partial}{\partial t} + \vec{u} \cdot \nabla$, $\vec{u}$ is the velocity vector (u, v, w), p is the pressure, ρ is the density, ν is the coefficient of kinematic viscosity and $\vec{F}$ represents the external and body forces.

These equations can be solved by the Eulerian approach where one solves for the vector fields that define the flow at fixed points of a grid or by the Lagrangian approach where one solves for the trajectory of a set of particles evolving in the flow. In the case of turbulent flow the chaotic nature of the flow makes the problem of defining the size/shape/placement/resolution of the grid tricky. Also grid-based techniques require significant insight into the expected behavior of the flow; for example, the grid should be shifted in the direction of an external wind-field to keep the solutions on the grid points relevant. These problems are overcome in the Lagrangian approaches as they are grid-less.

When computations are applied for real-time simulations they must be stable. Turbulent flows are chaotic and very sensitive to small changes in initial conditions, therefore the stability of the computations cannot be guaranteed. However this sensitivity of flow to small changes in initial conditions makes it suitable to model stochastically. We chose the stochastic Lagrangian approach to maintain the grid-less nature of the computations. In this approach the fluid flow is modelled by a set of particles whose statistical characteristics are the same as those of particles that evolve based on the equations of flow. These approaches are numerically more stable than the deterministic solutions to the equations.

The turbulent motion of the particles is simulated by using a stochastic Lagrangian approach. The following equations define the evolution of the i^{th} particle in the simulation [1]:

$$d\vec{X}^{(i)} = \vec{U}^{(i)} dt \quad (3)$$

$$d\vec{U}^{(i)} = \tfrac{3}{4} C_0 \langle\omega\rangle(\vec{U}^{(i)}(t) - \langle U\rangle)dt + \sqrt{C_0 k \langle\omega\rangle}\, d\vec{W} \quad (4)$$

$$d\omega^{(i)} = -(\omega^{(i)} - \langle\omega\rangle)C_3\langle\omega\rangle + \sqrt{(2\sigma^2\langle\omega\rangle\omega^{(i)}C_3\langle\omega\rangle)}\, dW^* \quad (5)$$

The first equation defines the position of a particle based on its velocity. The computation of velocity is based on the simplified Langevin model for stationary isotropic turbulence with constant density. The details of the derivation are beyond the scope of this paper and can be found in the book on turbulent flows by Pope [1]. The terms that are enclosed in $\langle\ \rangle$ represent the local mean values of the enclosed variables. In combustion studies they are evaluated by dividing the region occupied by the particles into a grid and considering the particles that lie in the same grid cell as the i^{th} particle. In our approach we use a kd-tree to store the particle system and evaluate these local mean values on n nearest neighboring particles of the i^{th} particle. The value of n used in the work varied between ten to twenty. This approach of storing particles in a kd-tree was introduced earlier in [26] and is called the particle map approach. The constant $C_0 = 2.1$ is the standard value used in turbulent flow simulations. $d\vec{W}$ represents an increment in the isotropic Wiener process $\vec{W}(t)$. It is implemented as a vector of three independent samples of the standard normal distribution. The next equation represents the evolution of the turbulent frequency. The value of of the constant C_3 is 1, and dW^* represents an increment in a scalar Wiener process W*(t), which is independent of the Wiener process in the previous equation.

The above enables us to model the turbulent motion of fire. In the next sub-section we describe modelling of the chemical aspects of fire.

Chemical evolution model

The chemical evolution model simulates the changing composition of fuel in the fire as the reaction progresses. Modelling of global extinction requires identification of the stage in reaction progress when it is no longer able to sustain a visible flame. At this stage global extinction occurs. After global extinction the diffusion of fuel and oxidizer continues and the conditions for combustion are again met and a flame reappears. The whole process occurs in a fraction of a second. Therefore the actual moment when no flame exists is not actually perceived but rather a flicker in the flame is observed. This phenomenon has not been modelled by the typical approaches to model fire in computer graphics that concentrate on modelling the variation of temperature in the flame.

Various approaches to modelling the chemical aspects exist in combustion studies. Many of these results are based on empirical studies of fire [28]. There is still a large gap between models in combustion and the actual phenomena as several simplifying assumptions have to be made to create a model. For example each fuel has its own unique way of burning depending on its chemical composition, diffusivity of fuel, oxidizer and products. As a result combustion studies often focus on specific fuels under controlled conditions [29]; for our work we require a more general model. The Euclidian Minimum Spanning Tree (EMST) mixing model is a general model for modelling evolution of composition of fuel during combustion in a turbulent flow. Subramaniam and Pope [30] propose the model and apply it with a simple periodic thermochemical model [31] to compare the approach to other approaches to model combustion in their subsequent work [32]. The fundamental concept of this model is to associate chemical composition parameters with the particles involved in the combustion process. The composition of the particles is initially defined using a periodic thermochemical model. The composition of the particles is subsequently evolved by constructing a EMST in composition space, and updating the composition by considering the particles' neighboring nodes in the tree. This approach of updating the composition helps to maintain the locality of chemical composition evolution with combustion; this reflects visually as the ability to simulate flame brushes. In this approach global extinction is estimated by computing the expected value of the reaction progress variable and comparing it with a threshold value. If the value is less than the threshold value then global extinction occurs.

The EMST model has been successfully used to simulate flickering fire [4]. However it is very complex, computationally expensive and involves several parameters. Also the parameters do not directly relate to the visual aspects of the resulting model of fire. Therefore the model is not suitable for applying to a real-time approach to simulate flickering fire with parameters that control the appearance of fire.

We formulate a simplified model that mimics the main aspects of the EMST model model for real-time applications. In the EMST model with the simplified thermochemistry the initial composition of the particles at equilibrium is as shown by the solid line in figure 1. The composition then evolves with time to a distribution along the dotted line in figure 1. The extent to which it evolves towards the dotted line is dependent on the nature of the combustion. When the combustion is steady, the time for mixing of oxidizer and fuel is comparable to the time of combustion; hence, the line remains close to the equilibrium state indicated by the solid line. However, when the reaction is not steady (when there is global extinction), the time taken for diffusion is significantly larger than the time for chemical reaction; hence, the composition of particles evolves to the dotted line in figure1. The exact distribution of the compositions may vary significantly depending on the value of several parameters that define the EMST mixing model [32]. The essence of the composition evolution in the EMST model can be summarized as a shift from the solid curve to the dotted curve in 1 while maintaining the neighborhood in composition space.

The x axis in figure 1 is the mixture fraction $\xi(\vec{X}, t)$, and the y axis it the reaction progress variable $Y(\vec{X}, t)$. Here $\vec{X}$ is the position vector of the particle (x_1, x_2, x_3). The reac-

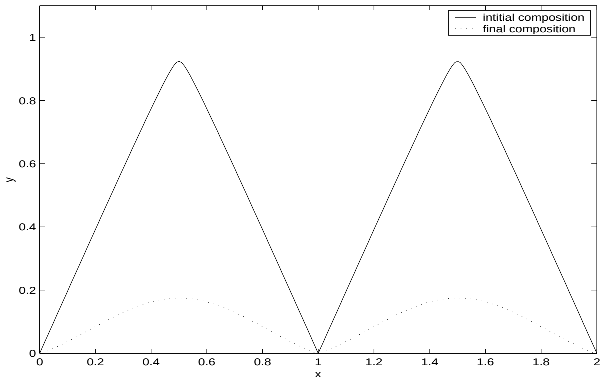

Figure 1: Plots of main curve along which composition values are distributed at the initial and final (before global extinction) time step for the EMST mixing model simulation. The x axis in the is the mixture fraction $\xi(\vec{X},t)$, and the y axis it the reaction progress variable $Y(\vec{X},t)$.

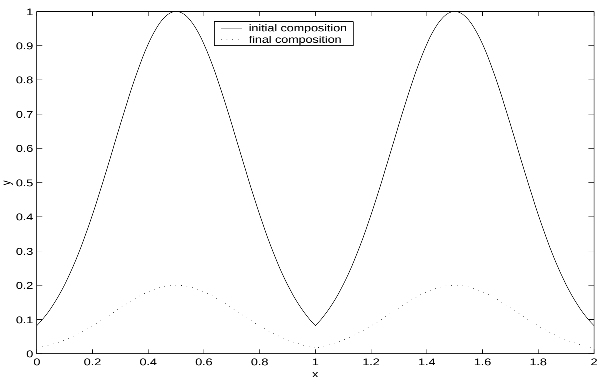

Figure 2: Plots of main curve along which composition values are distributed at the initial and final (before global extinction) time step for our model that mimics the EMST mixing model. The x axis in the is the mixture fraction $\xi(\vec{X})$, and the y axis it the reaction progress variable $Y(\vec{X},t)$.

tion progress variable is the mass fraction of product where the chemical reaction considered is fuel + oxidant $\rightleftharpoons$ product [32].

In our simplified model we do not evolve the values of the mixture fraction; therefore we represent it by $\xi(\vec{X})$ by removing its dependence on time. The values of $\xi(\vec{X})$ for a particle are defined such that that the gradient $\partial\xi/\partial x_1$ is a constant as in the case of [32]. However we defined this constant as a parameter $\eta \in (0.0, \infty]$ in our approach. This parameter is used to control the number of flame brushes. The number of flame brushes that occur in a spatial region where the value of x_1 varies by one unit is equal to the value of η. Therefore, when $\eta = 1.0$ there is a single flame brush in the spatial region, where the value of x_1 varies by one unit. The value of $Y(\vec{X},t)$ at $t = 0$ in our model is defined as:

$$Y(\vec{X},0) = Y(\xi(\vec{X})) = e^{(-(\xi(\vec{X}) - \lfloor \xi(\vec{X}) \rfloor - 0.5)^2/\lambda)} \tag{6}$$

This is the representation of adjacent overlapping Gaussian distributions. The parameter $lamda \in (0.0, \infty]$ controls the overlap between two neighboring flame brushes. Lower values result in less overlap and well separated flame brushes, while higher values result in greater overlap. Figure 2 gives the plot of the values of $Y(\vec{X},0)$ with $\eta = 1.0$ and $\lambda = 0.2$.

In our model we begin by distributing the composition of particles as a Gaussian distribution as given by equation 6 and illustrated in figure 2. We then evolve the composition to values about the curve shown in dotted lines in figure 2. We choose the Gaussian distribution as the starting distribution because it has been shown empirically and numerically that the distribution of composition should relax to a Gaussian with time. The EMST model is formulated so that the composition distribution relaxes to a Gaussian with successive updates of composition. In our simplified approach we start the fire visualization from the first step of simulation; there are no preprocessing simulation steps that allow the distribution to relax to a Gaussian. Therefore the distribution of composition should be a Gaussian from the start, this is ensured by the use of equation 6 to initialize the composition.

We evolve the composition $Y(\vec{X},t)$ with the equation:

$$Y(\vec{X},t) = (\chi + rd) * Y(\vec{X},t-1) \tag{7}$$

where χ is the rate of decrease of the reaction progress variable. $rd \in [0, 0.01 * \chi]$ is a small random perturbation in the value of χ. This simple approach to updating the composition mimics the essence of the EMST mixing model as the neighborhood regions are maintained in composition space and there is an evolution between the initial and final curves that have a similar appearance. The value of χ can be in the range $(0, 1.0]$. It was found that values of χ in the range $[0.85, 1.0]$ give visually realistic results. Values of χ tending towards 1.0 result in high flames as the reaction progress of the particle remains in the visible range for more time steps of the dynamics simulation.

Global extinction is identified as the stage during combustion when the overall reaction progress is not enough to sustain the flame. At this stage the flame reduces in intensity and reappears in the next time step when the compositions of the particles are redistributed at thermochemical equilibrium (the values at $Y(\vec{X},0)$). In the EMST model the stage of global extinction is predicted by computing an extinction index that relates directly to the expected value of the reaction progress variable. The extinction index is compared with a fixed value; if the index is less, then global extinction is said to occur. In our simplified approach we compute the mean value of the reaction progress variable; if it is less that a threshold value, then global extinction occurs and we restart the simulation with new particles and composition as the old particles are no longer visible after global extinction. Thus,

$$if \quad mean(Y(\vec{X},0)) < \theta \quad global \quad extinction \tag{8}$$

where θ is a threshold parameter that can be adjusted to control the frequency of global extinction. The justification for varying the value of threshold θ is that various fuels produce different kind of flames and, depending on the fuel a different value of minimum reaction progress is needed to sustain a flame. We use the $mean(Y(\vec{X},0))$ rather than the sum of the reaction progress values to estimate the overall reaction progress in the system so that the threshold value is independent of the number of particles involved in the simulation.

A model of flickering fire that works in real-time is achieved with the techniques described in this section. In the

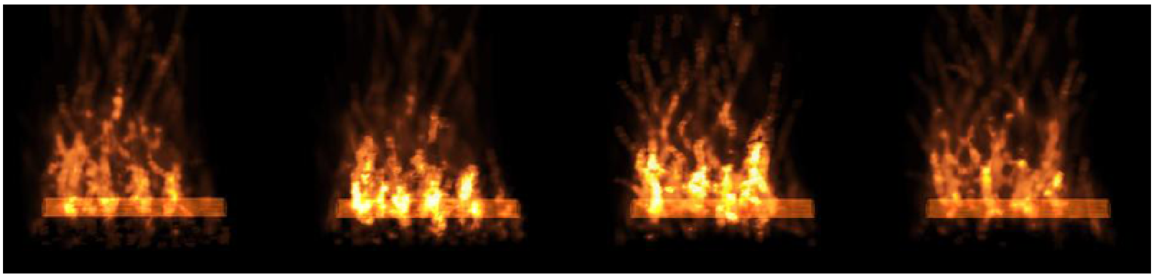

Figure 3: Images of fire between two stages of global extinction. Several frames exist between two time instances of global extinction; these are not consecutive frames.

following section we describe a method for rendering the particle system that is evolving based on the model described in this section.

Real-time rendering

We use a programmable graphics card to realize the rendering of the particle system evolving based on the model presented in the previous section. Specifically, we exploit the ability to render to the p-buffer in our approach to rendering.

We render the particle system as streaks of light extending from the current position of the particle to its previous position. This approach is adopted because when a bright light emitting particle moves with high velocity we perceive a streak due to persistence of vision. The composition parameter is used to define the texture coordinates of the line. The current value of composition is used as the texture coordinate at the current particle position and the composition at the previous time step is used for the other end of the line. Since a particle composition and location represent the characteristics of a small volume of the fuel located at a given position we associate a thickness with these lines. We render these lines into the p-buffer. We then create a blur/halo in the upward direction to represent the scattering of particle light by hot gaseous products resulting from combustion. We use two random textures to obtain offsets to the texture coordinates for blurring. The result of the computation is stored back in the p-buffer that is being used as the source to obtain the texture coordinates. This enables creating a cumulative blur.

The blurred texture is then used as a texture coordinate index into a one dimensional texture that represents the variation of light emitted with the progress of combustion.

Results and discussion

We implemented our model in C++ and OpenGL and ran it on machines with the Linux operating system. The algorithm performs at the rate of 58 to 59 frames per second both on a Pentium 4 machine running at 2.2G with a 768MB memory and GeForce 5800 graphics card and on a Athlon XP processor running at 1.46 GHz and 512 MB RAM with a GeForce 5900 Ultra graphics card. The number of particles used in all the images and animation submitted is 300.

Figure 3 shows some images of fire between two stages of global extinction. The fire in the image is generated with $\eta = 2$ and the spread of the fire is two units in the x_1 direction. Therefore there are four flame brushes. The value of λ is one and χ is 0.97. The threshold θ is 0.1.

Figure 4 shows fire generated with $\eta = 1.0$ and $\eta = 1.0$. In both cases λ was chosen to be 1.0. This creates an overlap of flame brushes that gives the fire a realistic appearance.

Figure 5 shows fires of different heights created with our model. For the tall flames the value of χ is close to one. Tall flames undergo little or no global extinction. When the value of χ is lower, global extinction occurs more frequently. This is consistent with the intuitive idea that when a flame is extinguished frequently if it has to start again from the fuel source and cannot propagate to a great height before it is extinguished again.

In our implementation the particles are introduced into the simulation by assigning an initial velocity in the upward direction to represent the initial upward velocity due to thermal buoyancy. The height of the flames with constant χ can also be controlled to some extent with the value of upward velocity assigned to the particles as they are introduced into the simulation. Higher initial upward velocity results in greater height of the flame. This again is consistent with the fact that a larger flame results when a fuel is injected/introduced into the oxidizer with greater velocity. It should be noted that the particles evolved with the turbulent flow equations 3 to 5 are not always guaranteed to move upwards. When a particle moves significantly in the outward direction it is deleted from the system; a new particle is introduced for every deleted particle. When the composition of a particle reduces so it no longer emits enough light to be visible, such a particle is also deleted from the simulation. In our examples we have not simulated smoke. A technique for simulating smoke can be introduced on top of the fire as described in [2]. In that case a particle that is no longer emitting light can be introduced into the smoke simulation system.

These images are created with constant threshold θ value. We also submit an animation in which we demonstrate the working of the parameters. The animations demonstrate our ability to create tall fires without global extinction and fires with varying frequency of global extinction. This variation is modelled by varying the threshold parameter θ. Higher thresholds result in more frequent flicker. A fire that flickers at a high frequency is often not height and appears more steady due to persistence of vision; this is clearly seen in our animation. We also demonstrate the ability of the model to simulate fires with various numbers of flame brushes. In the examples shown we maintain a constant number of flame brushes after each global extinction. However it is possible to create a more natural appearance of fire by varying the number of flame brushes after each global extinction.

We demonstrate with our results that it is possible to de-

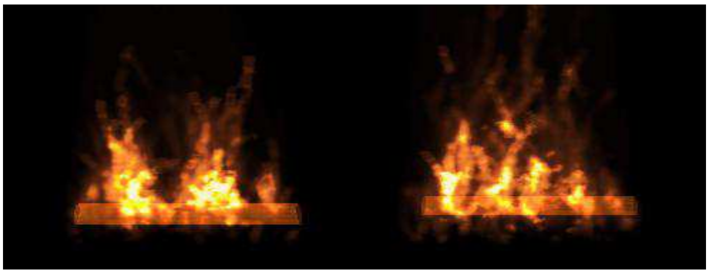

Figure 4: Comparison of fire with different number of flame brush. The fire on the left has two main regions ($\eta = 1.0$) while the one on the right has four regions ($\eta = 2.0$).

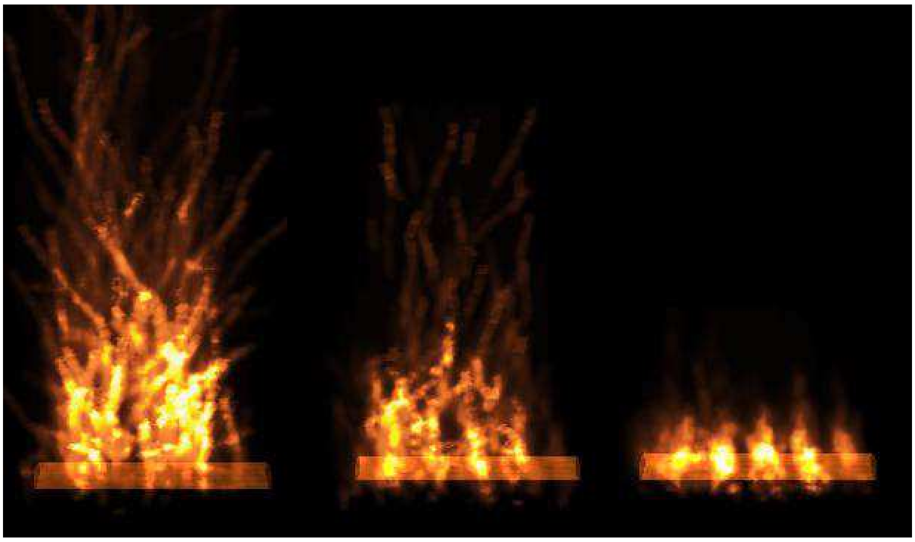

Figure 5: Comparison of fire with different heights of flames. Left image created with $\chi = 0.99999$, middle image created with $\chi = 0.97$ and right image generated with $\chi = 0.9$.

sign fires with desired visual properties using simple intuitive parameters η, χ and θ.

Conclusions

We have presented an approach to synthesize fire in real-time for computer graphics applications. The contributions of this work include:

- Present a grid-less stable numerical simulation technique for turbulent flow in the form of a stochastic Lagrangian approach based solution.
- Introduce a model for the phenomena of global extinction that enables capture of flicker in fire. This property of fire was not previously included in computer graphics models of fire.
- Parameterize the model such that flicker rate, flame height and number of flame brushes can be controlled in the model.
- Develop a hardware accelerated technique for rendering the particle system modelling fire.

We implemented the dynamics with particle maps making it grid-less, and thus overcoming the problem of having to address grid design related issues like choice of grid size, grid resolution and grid placement in space.

References

[1] S. B. Pope. *Turbulent Flows*. Cambridge University Press, Cambridge, 2000.

[2] A. Lamorlette and N. Foster. Structural modeling of natural flames. *Proceedings of ACM SIGGRAPH 2002*, pages 729–735, July 2002.

[3] W. Phillips, M. Shah, and N. da V. Lobo. Flame recognition in video. *Pattern Recognition Letters*, 23:319–327, January 2002.

[4] Adabala N., Hughes C. E., and Pattanaik S. N. A model for flicker in fire. *University of Central Florida*, CS-TR-0404, 2004.

[5] D. Q. Nguyen, R. Fedkiw, and H. W. Jensen. Physically based modeling and animation of fire. *Proceedings of ACM SIGGRAPH 2002*, 21:721–728, July 2002.

[6] R. Fedkiw, J. Stam, and H. W. Jensen. Visual simulation of smoke. *Proceedings of ACM SIGGRAPH 2001*, pages 15–22, August 2001.

[7] J. Stam. Stable fluids. *Proceedings of ACM SIGGRAPH 1999*, pages 121–128, August 1999.

[8] J. Stam and E. Fiume. Depicting fire and other gaseous phenomena using diffusion processes. *Proceedings of ACM SIGGRAPH 1995*, pages 129–136, August 1995.

[9] X. Wei, W. Li, K. Mueller, and A. Kaufman. Simulating fire with texture splats. *IEEE Visualization 2002*, pages 227–237, August 2002.

[10] R. Bukowski and C. Sequin. Interactive simulation of fire in virtual building environments. *Proceedings of ACM SIGGRAPH 1997*, pages 35–44, August 1997.

[11] H. Rushmeier, A. Hamins, and M. Y. Choi. Volume rendering of pool fire data. *IEEE Computer Graphics and Applications*, 15(4):62 – 67, 1995.

[12] G. P. Forney, D. Madrzykowski, K. B. McGrattan, and L. Sheppard. Understanding fire and smoke flow through modeling and visualization. *IEEE Computer Graphics and Applications*, 23(4):6–13, July/August 2003.

[13] J. Stam and E. Fiume. Turbulent wind fields for gaseous phenomena. *Proceedings of ACM SIGGRAPH 1993*, pages 369–376, August 1993.

[14] N. Foster and D. Metaxas. Modeling the motion of hot, turbulent gas. *Proceedings of ACM SIGGRAPH 1997*, pages 181–188, August 1997.

[15] M. Inakage. A simple model of flames. *Proceedings of Computer Graphics International*, pages 71–81, 1990.

[16] C. Perry and R. Picard. Synthesizing flames and their spreading. *Proceedings of 5th Eurographics Workshop on Animation and Simulation*, pages 56–66, 1994.

[17] P. Beaudoin and S. Paquet. Realistic and controllable fire simulation. *Proceedings of Graphics Interface 2001*, pages 159–166, 2001.

[18] K. Devlin and A. Chalmers. Realistic visualisation of the pompeii frescoes. *Proceedings of AFRIGRAPH 2001*, pages 43–47, 2001.

[19] W. T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *Proceedings of ACM SIGGRAPH 1983*, pages 91–108, 1983.

[20] G. D. Yngve, J. F. O' Brien, and J. K. Hodgins. Animating explosions. *Proceedings of ACM SIGGRAPH 2000*, pages 29–36, July 2000.

[21] B. E. Feldman, J. F. O'Brien, and O. Arikan. Animating suspended particle explosions. *Proceedings of ACM SIGGRAPH 2003*, July 2003.

[22] N. Rasmussen, D. Nguyen, W. Geiger, and R. P. Fedkiw. Smoke simulation for large scale phenomena. *Proceedings of ACM SIGGRAPH 2003*, pages 703–707, July 2003.

[23] A. Treuille, A. McNamara, Z. Popovic, and J. Stam. Keyframe control of smoke simulations. *Proceedings of ACM SIGGRAPH 2003*, pages 716–723, July 2003.

[24] H. E. Rushmeier. Rendering participating media: Problems and solutions from application areas. *Proc. of the 5th Eurographics Workshop on Rendering*, pages 35–56, 1994.

[25] S. Premoze, T. Tasdizen, J. Bigler, A. Lefohn, and R. Whitaker. Particle based simulation of fluids. *Proceedings of Eurographics 2003*, September 2003.

[26] Adabala N. and Manohar S. Modeling and rendering of gaseous phenomena using particle maps. *Journal of Visualization and Computer Animation*, 11:279–293, 2000.

[27] S. Garrick and V. Interrante. Stochastic modeling and simulation of turbulent reacting flows. *9th International Symposium on Flow Visualization*, pages 22–25, 2000.

[28] D. Drysdale. *An introduction to fire dynamics*. John Wiley and Sons, Chichester, New York, 1999.

[29] H. Guo, F. Liu, G. J. Smallwood, and O. L. Guelder. The flame preheating effect on numerical modelling of soot formation in a two-dimensional laminar ethylene-air diffusion flame. *Combustion Theory and Modelling*, 6(2):173–187, 2002.

[30] S. Subramaniam and S. B. Pope. A mixing model for turbulent reactive flows based on euclidean minimum spanning trees. *Combustion and Flame*, 115(4):487–514, 1998.

[31] Y. Y. Lee and S. B. Pope. Nonpremixed turbulent reacting flow near extinction. *Combustion and Flame*, 101:501–528, 1995.

[32] S. Subramaniam and S. B. Pope. Comparison of mixing model performance for nonpremixed turbulent reactive flow. *Combustion and Flame*, 117(4):732–754, 1999.