Grid-less Controllable Fire

Neeharika Adabala[1] and Charles E. Hughes[1,2]
[1]School of Computer Science
[2]School of Film and Digital Media,
University of Central Florida

Introduction

Gaming scenarios often involve fire; objects/vehicles/buildings on fire, torches of fire, fireplaces, etc. Fire is a phenomenon created by glowing combustion products in turbulent motion. Fire simulation techniques used in computer graphics usually involve solving the equations for dynamics of fluids on grids. These approaches are often computationally intensive and do not work in real-time [Nguyen01]. Other approaches [Wei02] work in real-time; however they are based on computations on three dimensional grids that introduce significant design issues in terms of choice of grid size, resolution and position. For example, what should be done when a wind blows on a fire? Should a grid be defined so that it encompasses the whole region that may ever contain the fire, or should it be designed to move with the fire? In the latter case, knowledge of the possible regions the fire could flow into is required. Also grid based computations are often not guaranteed to be stable, and relate to the resolution of the selected grid; this adds to the complexity of applying these approaches. Thus grid based simulations of fire demand skillful choice of grid size/position and resolution in every scenario that involves fire, and this can be a tedious task.

In this chapter, we present a grid-less technique for modeling fire based on a stochastic Lagrangian process [Pope00]. In this approach, the equations for dynamics simulation define the trajectory of each particle; therefore they can be directly evaluated to yield the position of each particle at successive time steps. The stochastic nature of the approach makes the computations relatively stable.

Most fires that are created in gaming scenarios are diffusion fires, or fires in which the oxidizer and fuel are not premixed (unlike the steady flame of a burner where the fuel and oxidizer are premixed). The fuel/object that is burning has to evaporate and come in contact with the oxidizer before it can burn. Since this process does not occur uniformly, fires flicker and exhibit a characteristic "jumping" behavior. Most of the existing approaches do not allow us to capture this distinct property of fire. In this chapter, we present a simplified approach to model this flickering fire that enhances its realism. The work of Lamorlette and Foster [Lamorlette02] identify the intermittent flame region in fire; however the model presented is for production environment rather than for real-time applications. In this chapter, the flickering of fire is captured by creating a simplified model for the "global extinction" behavior of fire. Global extinction refers to the moment when the combustion in the fire is so low that no flame is visible. The technique also models "flame brushes," which are regions of greater brightness in flames that occur in areas of the flame where there is higher rate of chemical reaction.

An additional issue in modeling of fire in games is the need to have parameters to control the appearance of fire. The approach presented in this chapter, enables control

of the flicker rate, flame height and number of flame regions in the fire. A technique for rendering in real-time that utilizes the programmability of graphics hardware is also described.

The chapter is organized as follows: the model for fire is presented in the next section. There are two main aspects to the model - the stochastic Lagrangian model for the dynamics and the chemical evolution model that represents the combustion accompanying the fire. The rendering of the model using programmable graphics hardware is then detailed. This is followed by a discussion with examples demonstrating the capabilities of the technique. Conclusions are given in the final section.

Model of fire

The key aspects that have to be modeled while simulating fire are turbulent dynamics and the chemical reaction accompanying it. A model for 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 modeled as an incompressible turbulent flow. The equations that define this flow are the equation for conservation of mass:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

and the Navier-Stokes equations:

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \upsilon\nabla^2\mathbf{u} + \mathbf{F} \tag{2}$$

where $D\mathbf{u}/Dt$ is the material derivative $\partial/\partial t + u\partial/\partial x + v\partial/\partial y + w\partial/\partial z = \partial/\partial t + \mathbf{u}\cdot\nabla$, $\mathbf{u}$ is the velocity vector (u,v,w), p is the pressure, ρ is the density, υ is the coefficient of kinematic viscosity and $\mathbf{F}$ 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. The stochastic Lagrangian approach to maintain the grid-less nature of the computations is used here. In this approach the fluid flow is modeled 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 [Pope00]:

$$d\mathbf{X}^{(i)} = \mathbf{U}^{(i)}dt \tag{3}$$

$$d\mathbf{U}^{(i)} = \frac{3}{4}C_0\langle\omega\rangle\mathbf{U}^{(i)}(t) - \langle\mathbf{U}\rangle dt + \sqrt{C_0 k\langle\omega\rangle}d\mathbf{W} \tag{4}$$

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

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 chapter and can be found in the book on turbulent flows by Pope [Pope00]. 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 [Bentley75] 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 in the range ten to twenty. This approach of storing particles in a kd-tree was introduced earlier in [Adabala00] and is called the particle map approach. The constant C_0 *=2.1* is the standard value used in turbulent flow simulations. *d***W** represents an increment in the isotropic Wiener process **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 the constant C_3 is 1, and W^* 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 modeling 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. Modeling 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 modeled by the typical approaches to model fire in computer graphics that concentrate on modeling the variation of temperature in the flame.

Various approaches to modeling the chemical aspects exist in combustion studies. Many of these results are based on empirical studies of fire [Drysdale99]. 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. The Euclidian Minimum Spanning Tree (EMST) mixing model is a general model for modeling evolution of composition of fuel during combustion in a turbulent flow. Subramaniam and Pope [Subramaniam98] propose the model and apply it with a simple periodic thermochemical model [Lee95] to compare the approach to other approaches to model combustion in their subsequent work [Subramaniam99]. 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 an 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.

We formulate a simplified model that mimics the main aspects of the EMST 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 figure 1. The exact distribution of the compositions may vary significantly depending on the value of several parameters that define the EMST mixing model [Subramaniam99]. 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 figure 1 while maintaining the neighborhood in composition space.

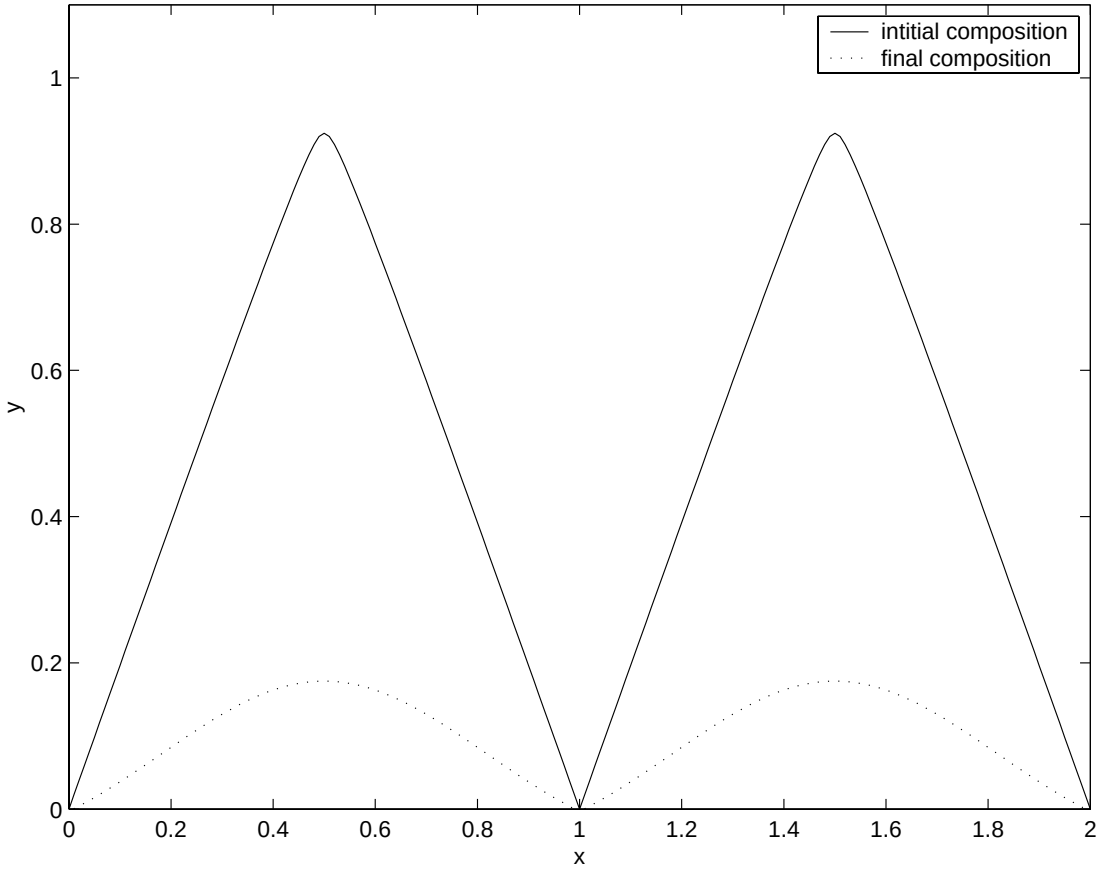

Figure1. Plots of main curve along which composition values are distributed at the initial and final (before global extinction) time step for the EMST mixing model simulation. The x axis in the is the mixture fraction $\xi(\boldsymbol{X},t)$, and the y axis it the reaction progress variable $Y(\boldsymbol{X},t)$

The x axis in figure 1 is the mixture fraction $\xi(\boldsymbol{X},t)$, and the y axis it the reaction progress variable $Y(\boldsymbol{X},t)$. Here $\boldsymbol{X}$ is the position vector of the particle (x_1, x_2, x_3). The reaction progress variable is the mass fraction of product where the chemical reaction considered is fuel + oxidant ↔ product [Subramaniam99].

In our simplified model we do not evolve the values of the mixture fraction; therefore we represent it by $\xi(\boldsymbol{X})$, by removing its dependence on time. The values of $\xi(\boldsymbol{X})$ for a particle are defined such that that the gradient $\partial\xi/\partial x_1$ is a constant as in the case of [Subramaniam99]. We defined a 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 there $\eta = 1.0$ is a single flame brush in the spatial region, where the value of x_1 varies by one unit. The value of $Y(\boldsymbol{X},t)$ at $t=0$ in our model is defined as:

$$Y(\mathbf{X},t) = Y(\xi(\mathbf{X})) = \exp(-(\xi(\mathbf{X}) - \lfloor \xi(\mathbf{X}) \rfloor) - 0.5)^2 / \lambda) \qquad (6)$$

This is the representation of adjacent overlapping Gaussian distributions. The parameter $\lambda \in (0.0, \infty]$ 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 $Y(\mathbf{X},t)$ with $\eta=1.0$ and $\lambda=0.8$.

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.

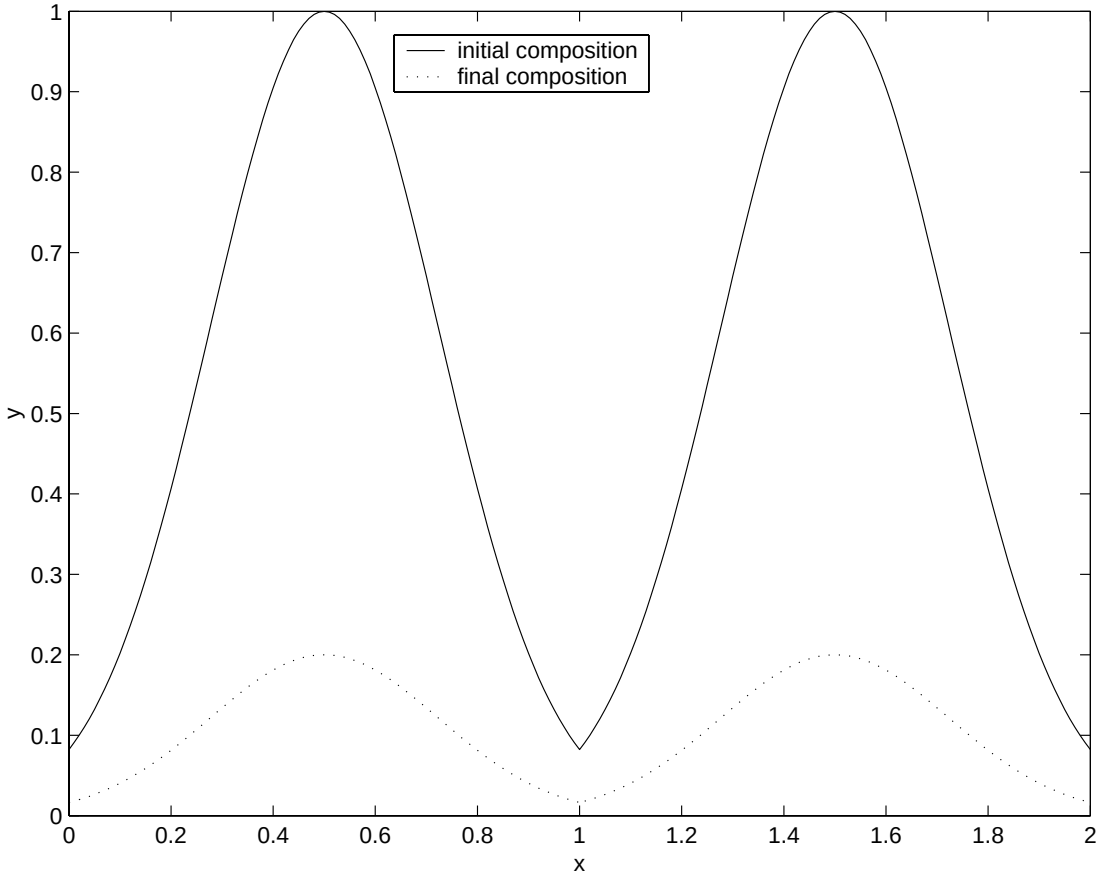

Figure 2. Plots of main curve along which composition values are distributed at the initial and final (before global extinction) time step for our model that ***mimics*** the EMST mixing model. The x axis in the is the mixture fraction $\xi(\mathbf{X})$, and the y axis it the reaction progress variable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

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

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

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 $Y(X,0)$). 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 of all the particles involved in the simulation; 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,

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

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. The value of θ is chosen in [0.0, 0.4] for visually realistic appearance. We use the $mean(Y(\mathbf{X},0))$ 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 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

A programmable graphics card is used to realize the rendering of the particle system evolving based on the model presented in the previous section. Specifically, the approach exploits the ability to render to the p-buffer.

The particle system is rendered 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 a thickness is associated with these lines. These lines are rendered into the p-buffer. A blur/halo is created in the upward direction to represent the scattering of particle light by hot gaseous products resulting from combustion. Two random textures are used 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.

Examples and discussion

The examples present here are implemented in C++ and OpenGL and run on machines with the Linux operating system. The algorithm perform 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 an 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 is300.

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. Values of θ in the range 0.1 to 0.4 give the most visually appealing results.

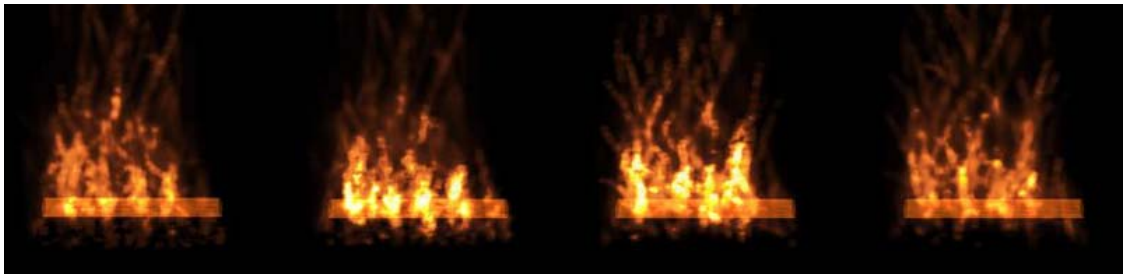

Figure 3 Images of fire between two stages of global extinction. Several frames exist between two time instances of global extinction; these are not consecutive frames

Figure 4 shows fire generated with $\eta=1.0$ and $\eta=2.0$. In both cases λ was chosen to be 1.0. This creates an overlap of flame brushes that gives the fire a realistic appearance.

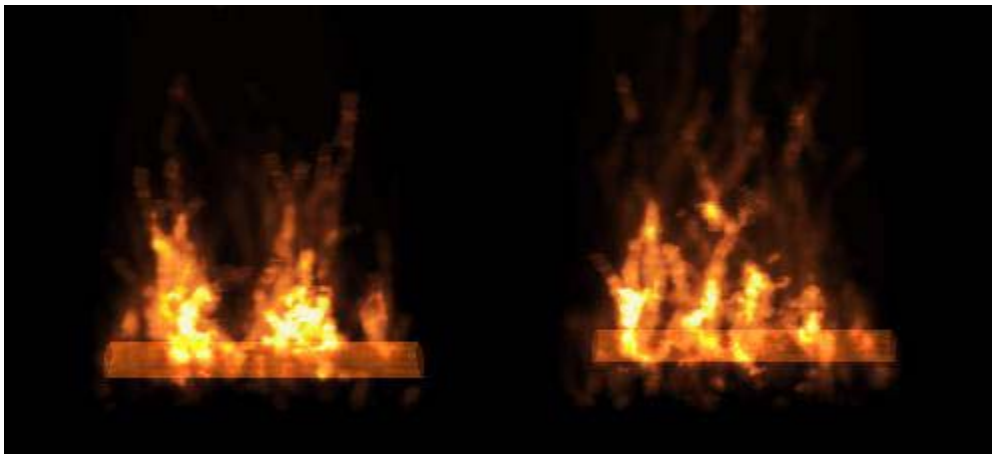

Figure 4. Comparison of fire with different number of flame brush.The fire on the left has two main regions ($\eta=1.0$) while the one on the right has four regions ($\eta=2.0$).

Figure 5 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.

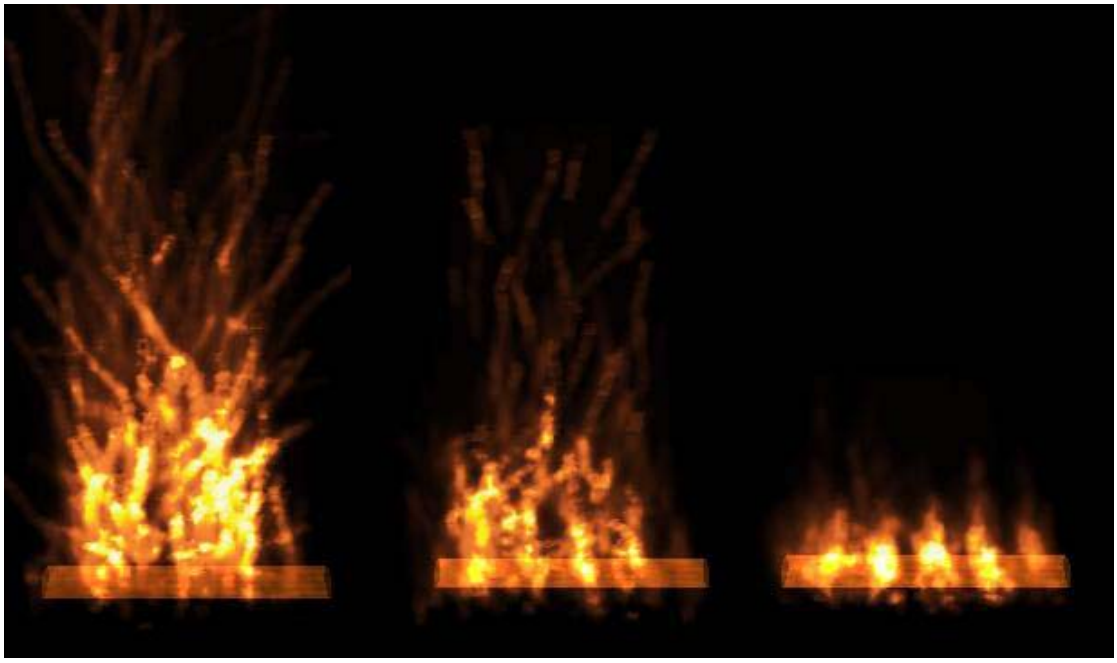

Figure 5. Comparison of fire with different heights of flames. Left image created with χ=0.99999, middle image created with χ=0.97 and right image generated with χ=0.9.

In this 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 the examples we have not simulated smoke. A technique for simulating smoke can be introduced on top of the fire as described in [Lamorlette02]. 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. The examples demonstrate that it is possible to design fires with desired visual properties using simple intuitive parameters η, χ and θ.

Conclusions

In this chapter, we have presented an approach to synthesize fire in real-time for computer graphics applications. The features of this work include:

- A grid-less stable numerical simulation technique for turbulent flow in the form of a stochastic Lagrangian approach based solution.
- 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.
- Parametric model such that flicker rate, flame height and number of flame brushes can be controlled in the model.
- A hardware accelerated technique for rendering the particle system modeling fire.

The dynamics is implemented with particle maps making the approach 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. The technique proposed is inspired by the physics and thermochemistry based models, however it is tailored for computer graphics and gaming applications where control of the visual aspects of fire is important, rather than physical accuracy.

References

[Nguyen01] Nguyen D. O., Fedkiw R., and Jensen H. W. Physically based modeling and animation of fire. *Proceedings of ACM SIGGRAPH 2002*, 21:721–728, July 2002.

[Wei02] Wei X, Li W., Mueller K., and Kaufman A. Simulating fire with texture splats. *IEEE Visualization 2002*, pages 227–237, August 2002.

[Pope00] Pope S. B. *Turbulent Flows*. Cambridge University Press, Cambridge, 2000.

[Lamorlette02] A. Lamorlette and N. Foster. Structural modeling of natural flames. *Proceedings of ACM SIGGRAPH 2002*, pages 729–735, July 2002.

[Bentley75] Bentley J. L., *Multidimensional Binary Search Trees Used for Associative Searching*. Communications of the ACM, 18(9):509--517, 1975.

[Adabala00] Adabala N. and Manohar S. Modeling and rendering of gaseous phenomena using particle maps. *Journal of Visualization and ComputerAnimation*, 11:279–293, 2000.

[Drysdale99] D. Drysdale. *An introduction to fire dynamics*. John Wiley and Sons, Chichester, New York, 1999.

[Subramaniam98] 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.

[Subramaniam99] S. Subramaniam and S. B. Pope. Comparison of mixing model performance for nonpremixed turbulent reactive flow. *Combustion and Flame*, 117(4):732–754, 1999.

[Lee95] Y. Y. Lee and S. B. Pope. Nonpremixed turbulent reacting flow near extinction. *Combustion and Flame*, 101:501–528, 1995.