Physically Based Modeling and Animation of Fire

Duc Quang Nguyen
Stanford University
Industrial Light + Magic
dqnguyen@stanford.edu

Ronald Fedkiw
Stanford University
Industrial Light + Magic
fedkiw@cs.stanford.edu

Henrik Wann Jensen
Stanford University
henrik@graphics.stanford.edu

Abstract

We present a physically based method for modeling and animating fire. Our method is suitable for both smooth (laminar) and turbulent flames, and it can be used to animate the burning of either solid or gas fuels. We use the incompressible Navier-Stokes equations to independently model both vaporized fuel and hot gaseous products. We develop a physically based model for the expansion that takes place when a vaporized fuel reacts to form hot gaseous products, and a related model for the similar expansion that takes place when a solid fuel is vaporized into a gaseous state. The hot gaseous products, smoke and soot rise under the influence of buoyancy and are rendered using a blackbody radiation model. We also model and render the blue core that results from radicals in the chemical reaction zone where fuel is converted into products. Our method allows the fire and smoke to interact with objects, and flammable objects can catch on fire.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Ray Tracing;

Keywords: flames, fire, smoke, chemical reaction, blackbody radiation, implicit surface, incompressible flow, stable fluids, vorticity confinement

1 Introduction

The modeling of natural phenomena such as fire and flames remains a challenging problem in computer graphics. Simulations of fluid behavior are in demand for special effects depicting smoke, water, fire and other natural phenomena. Fire effects are especially in demand due to the dangerous nature of this phenomenon. Fire simulations are also of interest for virtual reality effects, for example to help train fire fighters or to determine proper placement of exit signs in smoke filled rooms (i.e. so they can be seen). The interested reader is referred to [Rushmeier 1994].

Combustion processes can be loosely classified into two rather distinct types of phenomena: detonations and deflagrations. In both of these processes, chemical reactions convert fuel into hot gaseous products. Deflagrations are low speed events such as the fire and flames we address in this paper, while detonations are high speed events such as explosions where shock waves and other compressible effects are important, see e.g. [Yngve et al. 2000] and [Neff and Fiume 1999]. As low speed events, deflagrations can be modeled using the equations for incompressible flow (as opposed to those for compressible flow). Furthermore, since viscous effects are small, we use the incompressible inviscid Euler equations similar to [Fedkiw et al. 2001]. As noted therein, these equations can be solved efficiently using a semi-Lagrangian stable fluid approach, see e.g. [Staniforth and Cote 1991] and [Stam 1999].

Figure 1: **A turbulent gas flame model of a flamethrower.**

An important, often neglected aspect of fire and flame modeling concerns the expansion of the fuel as it reacts to form hot gaseous products. This expansion is the reason for the visual fullness observed in many flames and is partly responsible for the visual turbulence as well. Since the incompressible equations do not account for expansion, we propose a simple thin flame model for capturing these effects. This is accomplished by using an implicit surface to represent the reaction zone where the gaseous fuel is converted into hot gaseous products. Although real reaction zones have a nonzero (but small) thickness, the thin flame approximation works well for visual modeling and has been used by scientists as well, see for example [Markstein 1964] who first proposed this methodology.

Our implementation of the thin flame model is as follows. First, a dynamic implicit surface is used to track the reaction zone where the gaseous fuel is converted into hot gaseous products. Then both the gaseous fuel and the hot gaseous products are separately modeled using independent sets of incompressible flow equations. Finally, these incompressible flow equations are updated together in a coupled fashion using the fact that both mass and momentum must be conserved as the gas reacts at the interface. While this gives rather pleasant looking laminar (smooth) flames, we include a vorticity confinement term, see [Steinhoff and Underhill 1994] and [Fedkiw et al. 2001], to model the larger scale turbulent flame structures that are difficult to capture on the relatively coarse grids used for efficiency reasons in computer graphics simulations. We also include other features important for visual simulation, such as the buoyancy effects generated by hot gases and the interaction of fire with flammable and nonflammable objects. We render the fire as a participating medium with blackbody radiation using a stochastic ray marching algorithm. In our rendering we pay careful attention to the chromatic adaptation of the observer in order to get the correct colors of the fire.

Copyright © 2002 by the Association for Computing Machinery, Inc.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1-212-869-0481 or e-mail permissions@acm.org.
© 2002 ACM 1-58113-521-1/02/0007 $5.00

2 Previous Work

A simple laminar flame was texture mapped onto a flame-like implicit primitive and then volume-traced by [Inakage 1989]. [Perry and Picard 1994] applied a velocity spread model from combustion science to propagate flames. [Chiba et al. 1994] computed the exchange of heat between objects by projecting the environment onto a plane. The spread of flame was a function of both the temperature and the concentration of fuel. [Stam and Fiume 1995] present a similar model in three spatial dimensions for the creation, extinguishing and spread of fire. The spread of the fire is controlled by the amount of fuel available, the geometry of the environment and the initial conditions. Their velocity field is predefined, and then the temperature and density fields are advected using an advection-diffusion type equation. They render the fire using a diffusion approximation which takes into account multiple scattering. [Bukowski and Sequin 1997] integrated the Berkeley Architectural Walkthrough Program with the National Institute of Standards and Technology's CFAST fire simulator. The integrated system creates a simulation based design environment for building fire safety systems. An application of physically accurate firelight, and the impact of different fuel types on the color of flames and the scene they illuminate is given in [Devlin and Chalmers 2001]. Accurate ray casting through fire using spatially sparse measured data rather than simulated data was discussed in [Rushmeier et al. 1995] using radiation path integration software documented in [Grosshandler 1995].

Although we do not consider high-speed combustion phenomena such as detonations in this paper, there has been some notable work on this subject. [Musgrave 1997] concentrated on the explosive cloud portion of the explosion event using a fractal noise approach. [Neff and Fiume 1999] model and visualize the blast wave portion of an explosion based on a blast curve approach. [Mazarak et al. 1999] discuss the elementary blast wave equations, which were used to model exploding objects. They also show how to incorporate the blast wave model with a rigid body motion simulator to produce realistic animation of flying debris. Most recently, [Yngve et al. 2000] model the propagation of an explosion through the surrounding air using a computational fluid dynamics based approach to solve the equations for compressible, viscous flow. Their system includes two way coupling between solid objects and surrounding fluid, and uses the spectacular brittle fracture technology of [O'Brien and Hodgins 1999]. While the compressible flow equations are useful for modeling shock waves and other compressible phenomena, they introduce a very strict time step restriction associated with the acoustic waves. We use the incompressible flow equations instead to avoid this restriction making our method more computationally efficient.

3 Physically Based Model

We consider three distinct visual phenomena associated with flames. The first of these is the blue or bluish-green core seen in many flames. These colors are emission lines from intermediate chemical species, such as carbon radicals, produced during the chemical reaction. In the thin flame model, this thin blue core is located adjacent to the implicit surface. Therefore, in order to track this blue core, we need to track the movement of the implicit surface. The second visual phenomenon is the blackbody radiation emitted by the hot gaseous products, in particular the carbon soot. This is characterized by the yellowish-orange color familiarly associated with fire. In order to model this with visual accuracy we need to track the temperatures associated with a flame as depicted in Figure 2 (read from left to right). If the fuel is solid or liquid, the first step is the heating of the solid until it undergoes a phase change to the gaseous state. (Obviously, for gas fuels, we start in

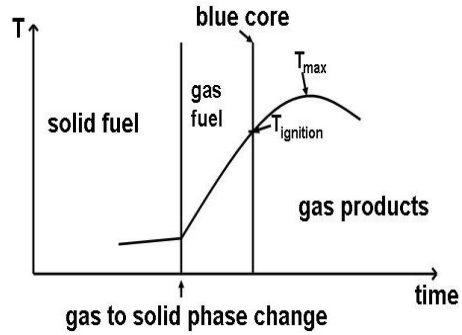

Figure 2: **Flame temperature profile for a solid (or gaseous) fuel.**

this gaseous state region in the figure.) Then the gas heats up until it reaches its ignition temperature corresponding to our implicit surface and the beginning of the thin blue core region. The temperature continues to increase as the reaction proceeds reaching a maximum before radiative cooling and mixing effects cause the temperature to decrease. As the temperature decreases, the blackbody radiation falls off until the yellowish-orange color is no longer visible. The third and final visual effect we address is the smoke or soot that is apparent in some flames after the temperature cools to the point where the blackbody radiation is no longer visible. We model this effect by carrying along a density variable in a fashion similar to the temperature. One could easily add particles to represent small pieces of soot, but our focus in this paper is the fire, not the smoke. For more details on smoke, see [Foster and Metaxas 1997], [Stam 1999] and [Fedkiw et al. 2001]. Figure 3 shows smoke coupled to our gas flame.

3.1 Blue Core

Our implicit surface separates this gaseous fuel from the hot gaseous products and surrounding air. Consider for example the injection of gaseous fuel from a cylindrically shaped tube. If the fuel were not burning, then the implicit surface would simply move at the same velocity as the gaseous fuel being injected. However, when the fuel is reacting, the implicit surface moves at the velocity of the unreacted fuel plus a flame speed S that indicates how fast fuel is being converted into gaseous products. S indicates how fast the unreacted gaseous fuel is crossing over the implicit surface turning into hot gaseous products. The approximate surface area of the blue core, A_S, can be estimated with the following equation

$$v_f A_f = S A_S, \tag{1}$$

Figure 3: **The hot gaseous products and soot emit blackbody radiation that illuminates the smoke.**

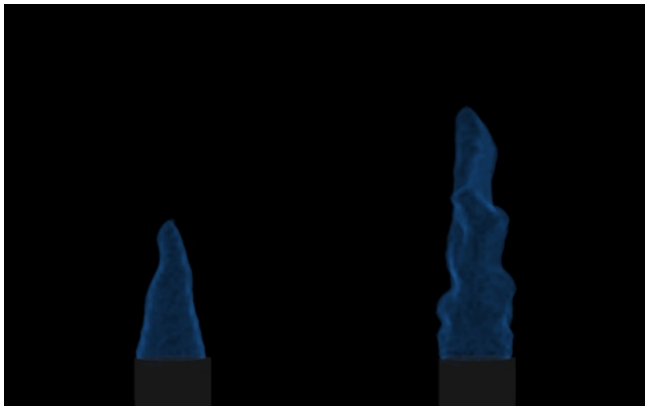

Figure 4: **Blue reaction zone cores for large (left) and small (right) values of the flame reaction speed S. Note the increased turbulence on the right.**

where v_f is the speed the fuel is injected across the injection surface with area A_f, e.g. A_f is the cross section of the cylindrical tube. This equation results from canceling out the density in the equation for conservation of mass. The left hand side is the fuel being injected into the region bounded by the implicit surface, and the right hand side is the fuel leaving this region crossing over the implicit surface as it turns into gaseous products. From this equation, we see that injecting more (less) gas is equivalent to increasing (decreasing) v_f resulting in a larger (smaller) blue core. Similarly, increasing (decreasing) the reaction speed S results in a smaller (larger) blue core. While we can turn the velocity up or down on our cylindrical jet, the reaction speed S is a property of the fuel. For example, S is approximately $.44m/s$ for a propane fuel that has been suitably premixed with oxidizer [Turns 1996]. (We use $S = .5m/s$ for most of our examples.) Figure 4 shows the effect of varying the parameter S. The smaller value of S gives a blue core with more surface area as shown in the figure.

This thin flame approximation is fairly accurate for premixed flames where the fuel and oxidizer are premixed so that the injected gas is ready for combustion. Non-premixed flames, commonly referred to as diffusion flames, behave somewhat differently. In a diffusion flame, the injected fuel has to mix with a surrounding oxidizer before it can combust. Figure 5 shows the injection of fuel out of a cylindrically shaped pipe. The cone shaped curve is the predicted location of the blue core for a premixed flame while the larger rounded curve is the predicted location of the blue core for a diffusion flame. As can be seen in the figure, diffusion flames tend to have larger cores since it takes a while for the injected fuel and surrounding oxidizer to mix. This small-scale molecular diffusion process is governed by a second order partial differential equation that is computationally costly model. Thus for visual purposes, we model diffusion flames with larger blue cores simply by using a smaller value of S than that used for a corresponding premixed flame.

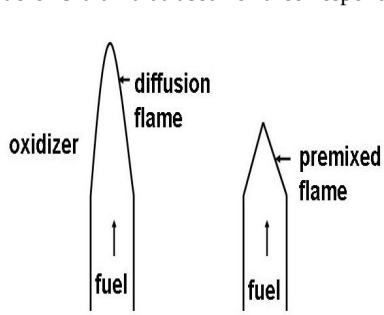

Figure 5: **Location of the blue reaction zone core for a premixed flame versus a diffusion (non-premixed) flame**

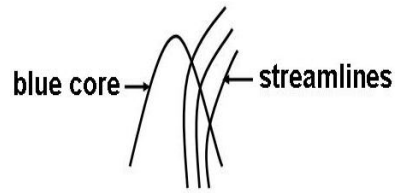

Figure 6: **Streamlines illustrating the path of individual fluid elements as they across the blue reaction zone core. The curved path is caused by the expansion of the gas as it reacts.**

3.2 Hot Gaseous Products

In order to get the proper visual look for our flames, it is important to track individual elements of the flow and follow them through their temperature histories given by Figure 2. This is particularly difficult because the gas expands as it undergoes reaction from fuel to hot gaseous products. This expansion is important to model since it changes the trajectories of the gas and the subsequent look and feel of the flame as individual elements go through their temperature profile. Figure 6 shows some sample trajectories of individual elements as they cross over the reaction front. Note that individual elements do not go straight up as they pass through the reaction front, but instead turn outward due to the effects of expansion. It is difficult to obtain visually full turbulent flames without modeling this expansion effect. In fact, many practitioners resort to a number of low level hacks (and lots of random numbers) in an attempt to sculpt this behavior, while we obtain the behavior intrinsically by using the appropriate model. The expansion parameter is usually given as a ratio of densities, ρ_f/ρ_h where ρ_f is the density of the gaseous fuel and ρ_h is the density of the hot gaseous products. Figure 7 shows three flames side by side with increasing amounts of expansion from left to right. Note how increasing the expansion makes the flames appear fuller. We used $\rho_f = 1kg/m^3$ (about the density of air) for all three flames with $\rho_h = .2kg/m^3$, $.1kg/m^3$ and $.05kg/m^3$ from left to right.

We use one set of incompressible flow equations to model the fuel and a separate set of incompressible flow equations to model the hot gaseous products and surrounding airflow. We require a model for coupling these two sets of incompressible flow equations together across the interface in a manner that models the expansion that takes place across the reaction front. Given that mass and momentum are conserved we can derive the following equations for the coupling across the thin flame front:

$$\rho_h(V_h - D) = \rho_f(V_f - D), \quad (2)$$

$$\rho_h(V_h - D)^2 + p_h = \rho_f(V_f - D)^2 + p_f, \quad (3)$$

Figure 7: **Comparison of flame shapes for differing degrees of gaseous expansion. The amount of expansion increases from left to right making the flame appear fuller and more turbulent.**

where V_f and V_h are the normal velocities of the fuel and the hot gaseous products, and p_f and p_h are their pressures. Here, $D = V_f - S$ is the speed of the implicit surface in the normal direction. These equations indicate that both the velocity and the pressure are discontinuous across the flame front. Thus, we will need to exercise caution when taking derivatives of these quantities as is required when solving the incompressible flow equations. (Note that the tangential velocities are continuous across the flame front.)

3.3 Solid Fuels

When considering solid fuels, there are two expansions that need to be accounted for. Besides the expansion across the flame front, a similar expansion takes place when the solid is converted to a gas. However, S is usually relatively small for this reaction (most solids burn slowly in a visual sense), so we can use the boundary of the solid fuel as the reaction front. Since we do not currently model the pressure in solids, only equation 2 applies. We rewrite this equation as

$$\rho_f(V_f - D) = \rho_s(V_s - D), \tag{4}$$

where ρ_s and V_s are the density and the normal velocity of the solid fuel. Substituting $D = V_s - S$ and solving for V_f gives

$$V_f = V_s + \left(\rho_s/\rho_f - 1\right) S \tag{5}$$

indicating that the gasified solid fuel moves at the velocity of the solid fuel plus a correction that accounts for the expansion. We model this phase change by injecting gas out of the solid fuel at the appropriate velocity. This can be used to set arbitrary shaped solid objects on fire as long as they can be voxelized with a suitable surface normal assigned to each voxel indicating the direction of gaseous injection.

In figure 8, we simulate a campfire using two cylindrically shaped logs as solid fuel injecting gas out of the logs in a direction consistent with the local unit surface normal. Note the realistic rolling of the fire up from the base of the log. The ability to inject (or not inject) gaseous fuel out of individual voxels on the surface of a complex solid object allows us to animate objects catching on fire, burn different parts of an object at different rates or not at all (by using spatially varying injection velocities), and extinguish solid fuels simply by turning off the injection velocity. While building an animation system that allows the user to hand paint temporally and spatially varying injection velocities on the surface of solid objects is beyond the scope of this paper, it is a promising subject for future work.

4 Implementation

We use a uniform discretization of space into N^3 voxels with uniform spacing h. The implicit surface, temperature, density and pressure are defined at the voxel centers and are denoted $\phi_{i,j,k}$, $T_{i,j,k}$, $\rho_{i,j,k}$ and $p_{i,j,k}$ where $i,j,k = 1,\cdots,N$. The velocities are defined at the cell faces and we use half-way index notation: $u_{i+1/2,j,k}$ where $i = 0,\cdots,N$ and $j,k = 1,\cdots,N$; $v_{i,j+1/2,k}$ where $j = 0,\cdots,N$ and $i,k = 1,\cdots,N$; $w_{i,j,k+1/2}$ where $k = 0,\cdots,N$ and $i,j = 1,\cdots,N$.

4.1 Level Set Equation

We track our reaction zone (blue core) using the level set method of [Osher and Sethian 1988] to track the moving implicit surface. We define ϕ to be positive in the region of space filled with fuel, negative elsewhere and zero at the reaction zone.

The implicit surface moves with velocity $\mathbf{w} = \mathbf{u}_f + S\mathbf{n}$ where $\mathbf{u}_f$ is the velocity of the gaseous fuel and the $S\mathbf{n}$ term governs the conversion of fuel into gaseous products. The local unit normal, $\mathbf{n} = \nabla\phi/|\nabla\phi|$ is defined at the center of each voxel using central differencing to approximate the necessary derivatives, e.g. $\phi_x \approx (\phi_{i+1,j,k} - \phi_{i-1,j,k})/2h$. Standard averaging of voxel face values is used to define $\mathbf{u}_f$ at the voxel centers, e.g. $u_{i,j,k} = (u_{i-1/2,j,k} + u_{i+1/2,j,k})/2$. The motion of the implicit surface is defined through

Figure 8: **Two burning logs are placed on the ground and used to emit fuel. The crossways log on top is not lit so the flame is forced to flow around it.**

$$\phi_t = -\mathbf{w}\cdot\nabla\phi \tag{6}$$

and solved at each grid point using

$$\phi^{new} = \phi^{old} - \Delta t\left(w_1\phi_x + w_2\phi_y + w_3\phi_z\right) \tag{7}$$

and an upwind differencing approach to estimate the spatial derivatives. For example, if $w_1 > 0$, $\phi_x \approx (\phi_{i,j,k} - \phi_{i-1,j,k})/h$. Otherwise if $w_1 < 0$, $\phi_x \approx (\phi_{i+1,j,k} - \phi_{i,j,k})/h$. This simple approach is efficient and produces visually appealing blue cores.

To keep the implicit surface well conditioned, we occasionally adjust the values of ϕ in order to keep ϕ a signed distance function with $|\nabla\phi| = 1$. First, interpolation is used to reset the values of ϕ at voxels adjacent to the $\phi = 0$ isocontour (which we don't want to move since it is the visual location of the blue core). Then we march out from the zero isocontour adjusting the values of ϕ at the other grid points as we cross them. [Tsitsiklis 1995] showed that this could be accomplished in an accurate, optimal and efficient manner solving quadratic equations and sorting points with a binary heap data structure. Later, [Sethian 1996] proposed the finite difference formulation of this algorithm that we currently use.

4.2 Incompressible Flow

We model the flow of the gaseous fuel and the hot gaseous products using a separate set of incompressible Euler equations for each. Incompressibility is enforced through conservation of mass (or volume), i.e. $\nabla \cdot \mathbf{u} = 0$ where $\mathbf{u} = (u, v, w)$ is the velocity field. The equations for the velocity

$$\mathbf{u}_t = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla p/\rho + \mathbf{f} \tag{8}$$

are solved for in two parts. First, we use this equation to compute an intermediate velocity $\mathbf{u}^*$ ignoring the pressure term, and then we add the pressure (correction) term using

$$\mathbf{u} = \mathbf{u}^* - \Delta t \nabla p/\rho. \tag{9}$$

The key idea to this splitting method is illustrated by taking the divergence of equation 9 to obtain

$$\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{u}^* - \Delta t \nabla \cdot (\nabla p/\rho) \tag{10}$$

and then realizing that we want $\nabla \cdot \mathbf{u} = 0$ to enforce mass conservation. Thus the left hand side of equation 10 should vanish leaving a Poisson equation of the form

$$\nabla \cdot (\nabla p/\rho) = \nabla \cdot \mathbf{u}^*/\Delta t \tag{11}$$

that can be solved to find the pressure needed for updating equation 9.

We use a semi-Lagrangian stable fluids approach for finding the intermediate velocity $\mathbf{u}^*$ and refer the reader to [Stam 1999] and [Fedkiw et al. 2001] for the details. Since we use two sets of incompressible flow equations, we need to address the stable fluid update when a characteristic traced back from one set of incompressible flow equations crosses the implicit surface and queries the velocities from the other set of incompressible flow equations. Since the normal velocity is discontinuous across the interface, the straightforward stable fluids approach fails to work. Instead, we need to use the balance equation 2 for conservation of mass to correctly interpolate a velocity.

Suppose we are solving for the hot gaseous products and we interpolate across the interface into a region where a velocity from the gaseous fuel might incorrectly be used. Instead of using this value, we compute a ghost value as follows. First, we compute the normal velocity of the fuel, $V_f = \mathbf{u}_f \cdot \mathbf{n}$. Then we use the balance equation 2 to find a ghost value for V_h^G as

$$V_h^G = V_f + \left(\rho_f/\rho_h - 1\right) S. \tag{12}$$

Since the tangential velocities are continuous across the implicit surface, we combine this new normal velocity with the existing tangential velocity to obtain

$$\mathbf{u}_h^G = V_h^G \mathbf{n} + \mathbf{u}_f - (\mathbf{u}_f \cdot \mathbf{n})\mathbf{n} \tag{13}$$

as a ghost value for the velocity of the hot gaseous products in the region where only the fuel is defined. This ghost velocity can then be used to correctly carry out the stable fluids update. Since both $\mathbf{n}$ and $\mathbf{u}_f$ are defined throughout the region occupied by the fuel, and ρ_f, ρ_h and S are known constants, a ghost cell value for the hot gaseous products, $\mathbf{u}_h^G$, can be found anywhere in the fuel region (even quite far from the interface) by simply algebraically evaluating the right hand side of equation 13. [Nguyen et al. 2001] showed that this ghost fluid method, invented in [Fedkiw et al. 1999], could be used to compute physically accurate engineering simulations of deflagrations.

After computing the intermediate velocity $\mathbf{u}^*$ for both sets of incompressible flow equations, we solve equation 11 for the pressure and finally use equation 9 to find our new velocity field. Equation 11 is solved by assembling and solving a linear system of equations for the pressure as discussed in more detail in [Foster and Fedkiw 2001] and [Fedkiw et al. 2001]. Once again, we need to exercise caution here since the pressure is discontinuous across the interface. Using the ghost fluid method and equation 3, we can obtain and solve a slightly modified linear system incorporating this jump in pressure. We refer the reader to [Nguyen et al. 2001] for explicit details and a demonstration of the physical accuracy of this approach in the context of deflagration waves.

The temperature affects the fluid velocity as hot gases tend to rise due to buoyancy. We use a simple model to account for these effects by defining external forces that are directly proportional to the temperature

$$\mathbf{f}_{buoy} = \alpha (T - T_{air}) \mathbf{z}, \tag{14}$$

where $\mathbf{z} = (0, 0, 1)$ points in the upward vertical direction, T_{air} is the ambient temperature of the air and α is positive constant with the appropriate units.

Fire, smoke and air mixtures contain velocity fields with large spatial deviations accompanied by a significant amount of rotational and turbulent structure on a variety of scales. Nonphysical numerical dissipation damps out these interesting flow features, so we aim to add them back on the coarse grid. We use the vorticity confinement technique invented by Steinhoff (see e.g. [Steinhoff and Underhill 1994]) and used by [Fedkiw et al. 2001] to generate the swirling effects for smoke. The first step in generating the small scale detail is to identify the vorticity $\omega = \nabla \times \mathbf{u}$ as the source of this small scale structure. Each small piece of vorticity can be thought of as a paddle wheel trying to spin the flow field in a particular direction. Normalized vorticity location vectors, $\mathbf{N} = \nabla|\omega|/|\nabla|\omega||$ simply point from lower concentrations of vorticity to higher concentrations. Using these, the magnitude and direction of the vorticity confinement (paddle wheel) force is computed as

$$\mathbf{f}_{conf} = \varepsilon h (\mathbf{N} \times \omega), \tag{15}$$

where $\varepsilon > 0$ and is used to control the amount of small scale detail added back into the flow field. The dependence on h guarantees that as the mesh is refined the physically correct solution is still obtained. All these quantities can be evaluated in a straightforward fashion as outlined in [Fedkiw et al. 2001].

Usually a standard CFL time step restriction dictates that the time step $\triangle t$ should be limited by $\triangle t < h/|\mathbf{u}|_{max}$ where $|\mathbf{u}|_{max}$ is the maximum velocity in the flow field. While this is true for our level set equation 6 with $\mathbf{u}$ replaced by $\mathbf{w}$, the combination of the semi-Lagrangian discretization and the ghost fluid method allows us to take a much larger time step for the incompressible flow equations. We choose our incompressible flow time step to be about five times bigger than that dictated by applying the CFL condition to the level set equation, and then stably update ϕ using substeps. This reduces the number of times one needs to solve for the pressure, which is the most expensive part of the calculation, by a factor of five.

4.3 Temperature and Density

The temperature profile has great effect on how we visually perceive flames, and we need to generate a temperature time history for fluid elements that behaves as shown in figure 2. Since this figure depicts a time history of the temperature of fluid elements, we need a way to track individual fluid elements as they cross over the blue core and rise upward due to buoyancy. In particular, we need to know how much time has elapsed since a fluid element has passed through the blue core so that we can assign an appropriate temperature to it. This is easily accomplished using a reaction coordinate variable Y governed by the equation

$$Y_t = -(\mathbf{u} \cdot \nabla) Y - k, \tag{16}$$

where k is a positive constant which we take to be 1 (larger or smaller values can be used to get a good numerical variation of Y in the flame). Ignoring the convection term, $Y_t = -1$ can be solved exactly to obtain $Y(t) = -t + Y(0)$. If we set $Y(0) = 1$ in the region of space occupied by the gaseous fuel and solve equation 16 for Y, then the local value of $1 - Y$ is equal to the total time elapsed since a fluid element crossed over the blue reaction core.

We solve equation 16 using the semi-Lagrangian stable fluids method to first update the convection term obtaining an intermediate value Y^*. Then we separately integrate the source term analytically so it too is stable for large time steps, i.e. $Y^{new} = -k\Delta t + Y^*$.

We can now use the values of Y to assign temperature values to the flow. Since $T_{ignition}$ is usually below the visual blackbody emission threshold, the temperature we set inside the blue core is usually not important. Therefore, we can set $T = T_{ignition}$ for the points inside the blue core. The region between the blue core and the maximum temperature in figure 2 is important since it models the rise in temperature due to the progress of a complex chemical reaction (which we do not model for the sake of efficiency). Here the animator has a lot of freedom to sculpt temperature rise curves and adjust how the mapping corresponds to the local Y values. For example, one could use $T = T_{ignition}$ at $Y = 1$, $T = T_{max}$ at $Y = .9$ and use a linear temperature function for the in between values of $Y \in (.9, 1)$. For large flames, this temperature rise interval will be compressed too close to the blue core for our grid to resolve. In these instances we use the ghost fluid method to set $T = T_{max}$ for any characteristic that looks across the blue core into the gaseous fuel region. The blue core then "spits" out gas at the maximum temperature that immediately starts to cool off, i.e. there is no temperature rise region. In fact, we did not find it necessary to use the temperature rise region in our examples as we are interested in larger scale flames, but this temperature rise region would be useful, for example, when modeling candle.

The animator can also sculpt the temperature falloff region to the right of figure 2. However, there is a physically correct, viable (i.e. computationally cheap) alternative. For the values of Y in the temperature falloff region, we simply solve

$$T_t = -(\mathbf{u} \cdot \nabla) T - c_T \left(\frac{T - T_{air}}{T_{max} - T_{air}} \right)^4 \tag{17}$$

which is derived from conservation of energy. Similar to equation 16, we solve this equation by first using the semi-Lagrangian stable fluids method to solve for the convection term. Then we integrate the fourth power term analytically to cool down the flame at a rate governed by the cooling constant c_T.

Similar to the temperature curve in figure 2, the animator can sculpt a density curve for smoke and soot formation. The density should start low and increase as the reaction proceeds. In the temperature falloff region, the animator can switch from the density curve to a physically correct equation

$$\rho_t = -(\mathbf{u} \cdot \nabla) \rho \tag{18}$$

that can (once again) be solved using the semi-Lagrangian stable fluids method. Again, we did not find it necessary to sculpt densities for our particular examples.

5 Rendering of Fire

Fire is a participating medium. It is more complex than the types of participating media (e.g. smoke and fog) that are typically encountered in computer graphics since fire emits light. The region that creates the light-energy typically has a complex shape, which makes it difficult to sample. Another complication with fire is that the fire is bright enough that our eyes adapt to its color. This chromatic adaptation is important to account for when displaying fire on a monitor. See [Pattanaik et al. 1998; Durand and Dorsey 2000].In this section, we will first describe how we simulate the scattering of light within a fire-medium. Then, we will detail how to properly integrate the spectral distribution of power in the fire and account for chromatic adaptation.

5.1 Light Scattering in a Fire Medium

Fire is a blackbody radiator and a participating medium. The properties of a participating medium are described by the scattering, absorption and emission properties. Specifically, we have the scattering coefficient, σ_s, the absorption coefficient, σ_a, and the extinction coefficient, $\sigma_t = \sigma_a + \sigma_s$. These coefficients specify the amount of scattering, absorption and extinction per unit-distance for a beam of light moving through the medium. The spherical distribution of the scattered light at a location is specified by a phase-function, p. We use the Henyey-Greenstein phase-function [Henyey and Greenstein 1941]

$$p(\vec{\omega} \cdot \vec{\omega}') = \frac{1 - g^2}{4\pi(1 + g^2 - 2g\vec{\omega} \cdot \vec{\omega}')^{1.5}}. \tag{19}$$

Here, $g \in [-1, 1]$ is the scattering anisotropy of the medium, $g > 0$ is forward scattering, $g < 0$ is backward scattering, while $g = 0$ is isotropic scattering. Note that the distribution of the scattered light only depends on the angle between the incoming direction, $\vec{\omega}$, and the outgoing direction, $\vec{\omega}'$.

Light transport in participating media is described by an integro-differential equation, the radiative transport equation [Siegel and Howell 1981]:

$$\begin{aligned} (\vec{\omega} \cdot \nabla) L_\lambda(x, \vec{\omega}) \quad = \quad & -\sigma_t(x) L_\lambda(x, \vec{\omega}) + \\ & \sigma_s(x) \int_{4\pi} p(\vec{\omega}, \vec{\omega}') L_\lambda(x, \vec{\omega}') d\vec{\omega}' + \\ & \sigma_a(x) L_{e,\lambda}(x, \vec{\omega}). \end{aligned} \tag{20}$$

Here, L_λ is the spectral radiance, and $L_{e,\lambda}$ is the emitted spectral radiance. Note that σ_s, σ_a, and σ_t vary throughout the medium and therefore depend on the position x.

We solve Equation 20 to estimate the radiance distribution in the medium by using a stochastic adaptive ray marching algorithm which recursively samples multiple scattering. In highly scattering media this approach is costly; however, we are concerned about fire which is a blackbody radiator (no scattering, only absorption) that creates a low-albedo smoke (the only scattering part of the fire-medium). This makes the Monte Carlo ray tracing approach practical.

To estimate the radiance along a ray traversing the medium, we split the ray into short segments. For a given segment, n, the scattering properties of the medium are assumed constant, and the radiance, L_n, at the start of the segment is computed as:

$$\begin{aligned} L_{n,\lambda}(x, \vec{\omega}) \quad = \quad & e^{-\sigma_t \Delta x} L_{(n-1),\lambda}(x + \Delta x, \vec{\omega}) + \\ & L_\lambda(x, \vec{\omega}') p(\vec{\omega} \cdot \vec{\omega}') \sigma_s \Delta x + \\ & \sigma_a L_{e,\lambda}(x) \Delta x. \end{aligned} \tag{21}$$

This equation is evaluated recursively to compute the total radiance at the origin of the ray. Δx is the length of the segment, L_{n-1} is the radiance at the beginning of the next segment, and $\vec{\omega}'$ is a sample direction for a new ray that evaluates the indirect illumination in a given direction for the segment. We find the sample direction by importance sampling the Henyey-Greenstein phase function. Note that we do not explicitly sample the fire volume; instead we rely on the Monte Carlo sampling to pick up energy as sample rays hit the fire. This strategy is reasonably efficient in the presence of the low-albedo smoke generated by the fire.

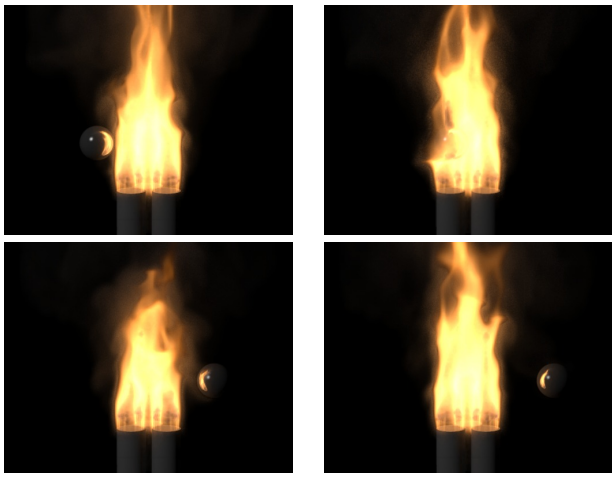

Figure 9: **A metal ball passes through and interacts with a gas flame.**

The emitted radiance is normally ignored in graphics, but for fire it is an essential component. For a blackbody we can compute the emitted spectral radiance using Planck's formula:

$$L_{e,\lambda}(x) = \frac{2C_1}{\lambda^5(e^{C_2/(\lambda T)} - 1)}, \tag{22}$$

where T is the temperature, $C_1 \approx 3.7418 \cdot 10^{-16} W m^2$, and $C_2 \approx 1.4388 \cdot 10^{-2} m^o K$ [Siegel and Howell 1981]. In the next section, we will describe how we render fire taking this spectral distribution of emitted radiance into account.

5.2 Reproducing the Color of Fire

Accurately reproducing the colors of fire is critical for a realistic fire rendering. The full spectral distribution can be obtained directly by using Planck's formula for spectral radiance when performing the ray marching. This spectrum can then be converted to RGB before being displayed on a monitor. To get the right colors of fire out of this process it is necessary to take into account the fact that our eyes adapt to the spectrum of the fire.

To compute the chromatic adaptation for fire, we use a von Kries transformation [Fairchild 1998]. We assume that the eye is adapted to the color of the spectrum for the maximum temperature present in the fire. We map the spectrum of this white point to the LMS cone responsivities (L_w, M_w, S_w). This enables us to map a spectrum to the monitor as follows. We first integrate the spectrum to find the raw XYZ tristimulus values (X_r, Y_r, Z_r). We then find the adapted XYZ tristimulus values (X_a, Y_a, Z_a) as:

$$\begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} 1/L_w & 0 & 0 \\ 0 & 1/M_w & 0 \\ 0 & 0 & 1/S_w \end{bmatrix} \mathbf{M} \begin{bmatrix} X_r \\ Y_r \\ Z_r \end{bmatrix}. \tag{23}$$

Here, $\mathbf{M}$ maps the XYZ colors to LMS (consult [Fairchild 1998] for the details). Finally, we map the adapted XYZ tristimulus values to the monitor RGB space using the monitor white point.

In our implementation, we integrate the spectrum of the blackbody at the source (e.g. when emitted radiance is computed); we then map this spectrum to RGB before using it in the ray marcher. This is much faster than doing a full spectral participating media simulation, and we found that it is sufficiently accurate, since we already assume that the fire is the dominating light source in the scene when doing the von Kries transformation.

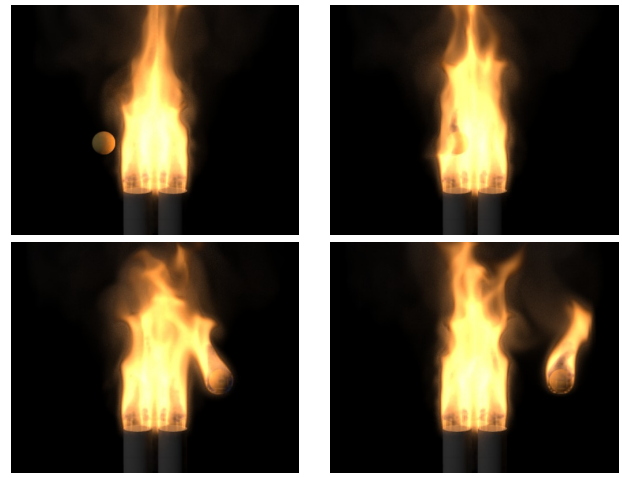

Figure 10: **A flammable ball passes through a gas flame and catches on fire.**

6 Results

Figure 1, rendered by proprietary software at ILM which is a research project not yet used in production, shows a frame from a simulation of a flamethrower. We used a domain that was 8 meters long with 160 grid cells in the horizontal direction ($h = .05$). The flame was injected at $30m/s$ out of a cylindrical pipe with diameter $.4m$. We used $S = .1m/s$, $\rho_f = 1kg/m^3$, $\rho_h = .01kg/m^3$, $c_T = 3000K/s$ and $\alpha = .15m/(Ks^2)$. The vorticity confinement parameter was set to $\varepsilon = 16$ for the gaseous fuel and to $\varepsilon = 60$ for the hot gaseous products. The simulation cost was approximately 3 minutes per frame using a Pentium IV.

Solid objects are treated by first tagging all the voxels inside the object as occupied. Then all the occupied voxel cell faces have their velocity set to that of the object. The temperature at the center of occupied voxels is set to the object's temperature and the (smoke) density is set to zero. Figure 9 shows a metal sphere as it passes through and interacts with a gas fire. Note the reflection of the fire on the surface of the sphere. For more details on object interactions with liquids and gases see [Foster and Fedkiw 2001] and [Fedkiw et al. 2001].

Since we have high temperatures (i.e. fire) in our flow field, we allow our objects to heat up if their temperature is lower than that of their surroundings. We use a simple conduction model where we increase the local temperature of an object depending on the surrounding air temperature and object temperature as well as the time step Δt. Normally, the value of the implicit surface is set to a negative value of h at the center of all voxels occupied by objects indicating that there is no available fuel. However, we can easily model ignition for objects we designate as flammable. Once the temperature of a voxel inside an object increases above a predefined threshold indicating ignition, we change the value of the implicit surface in that voxel from $-h$ to h indicating that it contains fuel. In addition, those voxel's faces have their velocities augmented above the object velocity by an increment in the direction normal to the object surface indicating that gaseous fuel is being injected according to the phase change addressed earlier for solid fuels. In figure 10, we illustrate this technique with a spherical ball that heats up and subsequently catches on fire as it passes through the flame. Both this flammable ball and the metal ball were computed on a $120 \times 120 \times 120$ grid at approximately 5 minutes per frame.

7 Conclusion

We have presented a physically based model for animating and rendering fire and flames. We demonstrated that this model could be used to produce realistic looking turbulent flames from both solid and gaseous fuels. We showed plausible interaction of our fire and smoke with objects, including ignition of objects by the flames.

8 Acknowledgment

Research supported in part by an ONR YIP and PECASE award N00014-01-1-0620, NSF DMS-0106694, NSF ACI-0121288, NSF IIS-0085864 and the DOE ASCI Academic Strategic Alliances Program (LLNL contract B341491). The authors would like to thank Willi Geiger, Philippe Rebours, Samir Hoon, Sebastian Marino and Industrial Light + Magic for rendering the flamethrower.

References

BUKOWSKI, R., AND SEQUIN, C. 1997. Interactive Simulation of Fire in Virtual Building Environments. In *Proceedings of SIGGRAPH 1997*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 35–44.

CHIBA, N., MURAOKA, K., TAKAHASHI, H., AND MIURA, M. 1994. Two dimensional Visual Simulation of Flames, Smoke and the Spread of Fire. *The Journal of Visualization and Computer Animation 5*, 37–53.

DEVLIN, K., AND CHALMERS, A. 2001. Realistic visualisation of the pompeii frescoes. In *AFRIGRAPH 2001*, ACM SIGGRAPH, A. Chalmers and V. Lalioti, Eds., 43–47.

DURAND, F., AND DORSEY, J. 2000. Interactive Tone Mapping. In *Proceedings of Eleventh Eurographics Workshop on rendering*, 219–230.

FAIRCHILD, M. 1998. *Color Appearance Models*. Addison Wesley Longman, Inc.

FEDKIW, R., ASLAM, T., MERRIMAN, B., AND OSHER, S. 1999. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method). *J. Comput. Phys. 152*, 457.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual Simulation of Smoke. In *Proceedings of SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 15–22.

FOSTER, N., AND FEDKIW, R. 2001. Practical Animation of Liquids. In *Proceedings of SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 23–30.

FOSTER, N., AND METAXAS, D. 1997. Modeling the Motion of a Hot, Turbulent Gas. In *Proceedings of SIGGRAPH 1997*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 181–188.

GROSSHANDLER, W. 1995. RADCAL: A Narrow-band Model for Radiation Calculations in Combustion Environment. *NIST Technical Note 1402* (April).

HENYEY, L., AND GREENSTEIN, J. 1941. Diffuse radiation in the galaxy. *Astrophysics Journal 93*, 70–83.

INAKAGE, M. 1989. A Simple Model of Flames. In *Proceedings of Computer Graphics International 89*, Springer-Verlag, 71–81.

MARKSTEIN, G. H. 1964. *Nonsteady Flame Propagation*. Pergamon, Oxford.

MAZARAK, O., MARTINS, C., AND AMANATIDES, J. 1999. Animating Exploding Objects. In *Proceedings of Graphics Interface 1999*, 211–218.

MUSGRAVE, F. K. 1997. Great Balls of Fire. In *SIGGRAPH 97 Animation Sketches, Visual Procedings*, ACM SIGGRAPH, 259–268.

NEFF, M., AND FIUME, E. 1999. A Visual Model for Blast Waves and Fracture. In *Proceedings of Graphics Interface 1999*, 193–202.

NGUYEN, D., FEDKIW, R., AND KANG, M. 2001. A Boundary Condition Capturing Method for Incompressible Flame Discontinuities. *J. Comput. Phys. 172*, 71–98.

O'BRIEN, J. F., AND HODGINS, J. K. 1999. Graphical Modeling and Animation of Brittle Fracture. In *Proceedings of SIGGRAPH 1999*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 137–146.

OSHER, S., AND SETHIAN, J. A. 1988. Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formualtions. *J. Comput. Phys. 79*, 12.

PATTANAIK, S., FERWERDA, J. A., FAICHILD, M. D., AND GREENBERG, D. P. 1998. A Multiscale Model of Adaptation and Spatial Vision for Realistic Image Display. In *Proceedings of SIGGRAPH 1998*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 287–298.

PERRY, C., AND PICARD, R. 1994. Synthesizing Flames and their Spread. *SIGGRAPH 94 Technical Sketches Notes* (July).

RUSHMEIER, H. E., HAMINS, A., AND CHOI, M. 1995. Volume Rendering of Pool Fire Data. *IEEE Computer Graphics and Applications 15*, 4, 62–67.

RUSHMEIER, H. 1994. Rendering Participating Media: Problems and Solutions from Application Areas. In *Proceedings of the 5th Eurographics Workshop on Rendering*, 35–56.

SETHIAN, J. 1996. A Fast Marching Level Set Method for Monotonically Advancing Fronts. *Proc. Nat. Acad. Sci. 93*, 1591–1595.

SIEGEL, R., AND HOWELL, J. 1981. *Thermal Radiation Heat Transfer*. Hemisphere Publishing Corp., Washington, DC.

STAM, J., AND FIUME, E. 1995. Depicting Fire and Other Gaseous Phenomena Using Diffusion Process. In *Proceedings of SIGGRAPH 1995*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 129–136.

STAM, J. 1999. Stable Fluids. In *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 121–128.

STANIFORTH, A., AND COTE, J. 1991. Semi-Lagrangian Integration Schemes for Atmospheric Models: A Review. *Monthly Weather Review 119*, 2206–2223.

STEINHOFF, J., AND UNDERHILL, D. 1994. Modification of the Euler Equations for "Vorticity Confinement": Application to the Computation of Interacting Vortex Rings. *Physics of Fluids 6*, 8, 2738–2744.

TSITSIKLIS, J. 1995. Efficient Algorithms for Globally Optimal Trajectories. *IEEE Transactions on Automatic Control 40*, 1528–1538.

TURNS, S. R. 1996. *An Introduction to Combustion*. McGraw-Hill, Inc.

YNGVE, G. D., O'BRIEN, J. F., AND HODGINS, J. K. 2000. Animating Explosions. In *Proceedings of SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 29–36.