Interactive Simulation of Fire

Zeki Melek John Keyser

Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA
E-mail: z0m8905@cs.tamu.edu
Phone: (979) 845-5007
Fax: (979) 847-8578

Technical Report 2002-7-1

July 15, 2002

Abstract

In this paper we describe a fast and interactive model to simulate and control the fire phenomenon. We use a modified interactive fluid dynamics solver to describe the motion of a 3-gas system. We simulate the motion of oxidizing air, fuel gases, and exhaust gases. The burning process is simulated by consuming fuel and air based on the amounts of fuel and air inside each grid cell. The combustion process produces heat, and we model the resulting spread of temperature through the system. The heat distribution induces convection currents in the air, causing the flame to take the appropriate shape. By modeling heat distribution, we also simulate the spread of fire to and self-ignition in other combustible solids.

Key Words: Physically based modeling, fire simulation, fluid dynamics

1. Introduction

There has been an increasing interest within the computer graphics community for simulation and visualization of natural phenomena such as water and smoke motion. These physical simulations play an increasingly important role as a part of the story within the motion picture and game industries, and can be important cues in applications such as military training. Generally, these applications require visually compelling but not strictly accurate models. However, the models need to be capable of capturing the visual characteristics of the phenomenon, and still be interactive and easy to choreograph. In this paper, we focus on the simulation of fire.

Recently, a number of advances have made realistic interactive modeling of smoke and other gaseous phenomena possible. These coarse-grid fluid-dynamic equation solvers are capable of capturing fine scale swirling motion of the air, with a very small computational cost.

Like smoke and water, the simulation of fire has many potential uses in entertainment and training applications. To date, however, graphical simulations of fire have either not modeled the physics of flame production, or have not approached interactive rates. In our work we model a physically based flame production process at interactive rates.

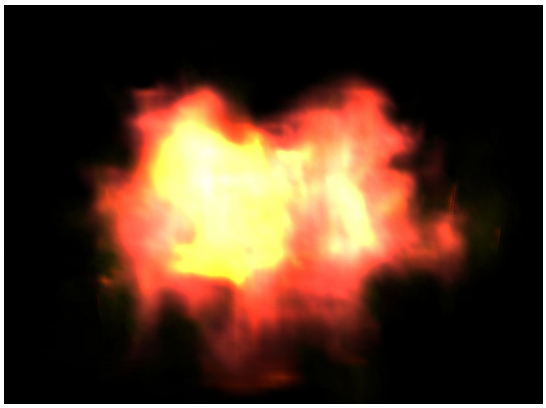


Figure 1. An example of our flame simulation, showing a fireball

1.1. Main Results

We present a method for realistic modeling and visualization of fire that is simple to implement and achieves interactive rates. We apply state of the art coarse grid fluid dynamic equation solvers to model the motion of fuel, air, and exhaust gasses in a unified system. Unlike other models, the heat produced by combustion affects the motion of the air within the computational domain, which in turn affects the shape and motion of the flame. In addition, this heat transport allows us to simulate self-ignition of objects away from the flame itself.

1.2. Organization

The rest of the paper is organized as follows. Section 2 provides background material, including prior approaches to flame modeling and fluid flow. Section 3 gives a brief overview of our method. Section 4 describes our fluid motion and gas combustion models. Section 5 discusses material self-ignition and flame spread. Section 6 describes aspects of visualization and flame control. Section 7 discusses the implementation of our approach. Section 8 concludes and points to directions of future work.

2. Background and Prior Work

We give here a brief background into the modeling of flames, techniques used previously in computer graphics to simulate fire, and an overview of the recent results in simulating fluids.

2.1. Models of Flames

Combustion is an ongoing area of research. Chemically, the combustion process is given by the equation:

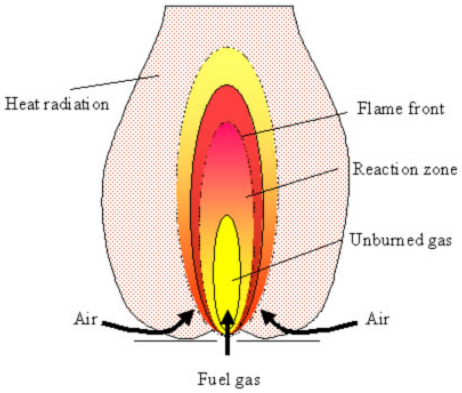


Figure 2. A simple flame model

$$a\ fuel + b\ oxidant \Rightarrow c\ CO_2 + d\ H_20 + .. \quad (1)$$

Fuel is preheated by ignition, and the gases that are mixed with air (the oxidant) in ”good” concentration start the combustion, giving an exothermic reaction. Besides water and carbon dioxide, the burning process results in other combustion byproducts, some of which might be subject to further combustion themselves.

The fuel gas mixes with the oxidizing gas creating a mixture zone. The ratio of gas to air varies across the mixture zone. The pure fuel moves towards the reaction zone and is heated. Because there is no oxygen in this case, the fuel is broken down into smaller molecules and radicals, and also forms soot (carbon). This is very luminous and gives diffusion flames their distinctive yellow color. As the reaction zone is approached, the increasing amount of oxidizing gas allows the chemical combustion reaction to occur. Combustion continues until the stoichiometric contour (flame front) is passed and the reaction is completed. The heat generated during the reaction keeps the reaction zone at the maximal temperature, triggering more combustion, thus creating more heat and combustion byproducts [6, 27]. Diffusion (laminar) flames are created when ”pure” fuel gas coming out of the fuel source mixes with the oxidizing gas through the mixture zone. Diffusion flames are the type commonly seen when an object burns. On the other hand, if fuel and oxidizing gases are well-mixed before entering the reaction zone, they create a premixed flame. As an example, some natural gas burners will mix air with the gas before combustion.

Once ignited, a diffusion flame stabilizes. Flames tend to be carried upward due to convection of the air that has been heated by the combustion process. Smoke, consisting of water vapor and other byproducts, is carried along with this column of air. The heat produced by the flame can also radiate to vaporize combustible products from nearby matter through chemical decomposition. This process is called pyrolysis [2]. Thus, when an object (such as a piece of wood) burns, the flames are formed

from combustible portions of the object being vaporized and then oxidizing in the region just beyond the surface.

The color of a flame in the reaction zone is determined by the energy released in the exothermic oxidation process. Thus, a ”pure” burning chemical may emit a single wavelength of light, while a more complex substance (e.g. wood) may emit at numerous wavelengths. In addition, the heat generated by combustion can cause combustion materials to emit light in form of thermal radiation [2, 27].

2.2. Flame Modeling in Computer Graphics

Within the computer graphics community, early models of fire include those of Reeves and Perlin. Reeves's approach [20], using particle systems, requires a very large number of particles to hide the pointillistic nature of the technique, and is thus computationally expensive. Perlin's approach [17, 18] uses a procedural noise function, which has problems including external forces such as wind into the simulation. Inakage [11] uses a physical model to emit light in the regions of combustion, but his model is computationally expensive and deals with still images rather than multiple animated flames.

More recently, Chiba et al. [4] and Takahashi et al. [26] both use a gridded representation of the space, where each grid cell contains a certain amount of fuel gas and heat. This is similar to our representation. Fire itself is modeled by placing geometric primitives around particle trajectories, which are influenced by a vector field. Perry and Picard [19] have a similar representation for the flames. They reuse the same polygons used for modeling to spread the fire. The flame front is represented by a set of particles, adding new particles as the front expands. Stam and Fiume [24] use a map covering the object defining the amount of fuel and temperature on every point on the object. Fire animation is done using turbulent wind fields [23] and rendering is based on warped blobs. The approach is costly and user controls are not so flexible and intuitive.

Beaudin et al. [1] use a similar but more accurate flame front technique to model the spreading of the flames. They guarantee that the boundary lies on the object, unlike Perry and Picard's work. They plant the root of the flame skeleton on the surface and displace the skeleton with the air vector field. They use an implicit surface representation to dress up the flame skeletons during the rendering. Their method is fast, interactive, and allows for quick preview, but lacks accurate air and smoke motion inside the simulation volume.

Yngve et al. [28] proposed a solution for the compressible version of the fluid flow equations, modeling the shock waves created by explosions, but it includes strict time step restrictions, making the solution computationally expensive. A recent paper Nguyen et al. [16] treats two phase incompressible flow where one phase is being converted into the other, e.g. vaporization of liquid water.

The spread of fire has been another area of research. NIST has developed the CFAST fire simulator [12]. CFAST gives an accurate simulation of the impact of fire and its byproducts, but operates in batch mode, providing the user graphs and data dumps as output. Bukowski and Squin [3] have integrated CFAST and the Berkeley Architectural Walkthru program, making the results of CFAST more understandable. However, the aim of this work is not to create visually appealing images, and the fire simulation itself is done offline.

Finally, two fire-related papers are scheduled to be presented at Siggraph 2002[14, 15]. The authors are not releasing these papers at this time, however, and so we cannot compare our approach to theirs.

2.3. Modeling Fluid Flow

Our model is based on the direct simulation of the incompressible fluid dynamic equations on coarse grids. Kajiya and Von Herzen [13] worked on very coarse grids, using the very limited computer power they had at that time. Foster and Metaxas [8, 9] reintroduced the method to the CG community, and were able to capture nice swirling motions using relatively coarse grids. Their model used an explicit integration scheme, bounding their time step in order to keep the simulation stable. They also use a costly relaxation step to ensure mass conservation. Yoshida et al. [29] used smoke particles (puffs) to model the motion of the smoke and vorticity fields to take obstacles into account. Stam [21, 22] used a semi-Lagrangian advection scheme and implicit solver to make the simulation unconditionally stable; even large time steps are allowed in his Stable Fluids model. He also used pressure-Poisson equations to conserve mass. Because he used a first order integration scheme, these simulations suffered from numerical dissipation. Fedkiw at al. [7] included vorticity confinement in the Stable Fluids model, keeping the smoke "alive" and increasing the quality of the flow with a very small amount of additional computational cost. The Stable fluids model can handle boundaries inside the computational domain.

2.4. Stable Fluids

To model the fluid flow, we use a modified version of Stam's Stable Fluids approach. We provide a brief overview of the method here, but the reader is encouraged to refer to the original reference for more details [21].

Compressible flow equations introduce a very strict time step restriction associated with the acoustic waves. To avoid this strict restriction incompressible flow equations are preferred whenever possible. When the speed of the flow is below the speed of sound, the compression effects are negligible [7]. Using coarse grids also lets us neglect viscous effects, which occur at a scale far below our grid cell size. In compact form, the incompressible Navier-Stokes equations governing fluid flow are as follows:

$$\nabla \cdot u = 0 \tag{2}$$

$$\frac{\partial u}{\partial t} = f - (u \cdot \nabla)u - \nabla p \tag{3}$$

where u is the velocity field, and p is a pressure field. Eq. 2 describes incompressibility, hence divergence free flow, and Eq. 3 describes fluid flow. Looking at Eq. 3 more closely, f is the external force, including buoyancy and gravity forces. The second term is the nonlinear part of the Navier-Stokes equation - it is the advection of the velocity field on itself. The last term accounts for the pressure induced motion. The solution of the equations involves three main steps:

- The first step determines the force term. More about the force is described below, but briefly we find the buoyancy force based on the temperature and the gravity force based on the amount of fuel and exhaust gas densities in each grid cell.
- The second step advects the velocity field itself to the next time step. A particle at each cell center is traced back in time over the time step and the new velocity for the cell is the interpolation of the velocities that the particle had a time step ago.

- $$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot u \tag{4}$$

  The third step combines Eq. 2 and 3 and makes the velocity field divergence free and incompressible using a projection method [5]. The vector field found in previous steps is made incompressible by subtracting the gradient of the pressure. The pressure is found by solving the Poisson equation (Eq. 4) with the Neumann boundary condition. The Poisson equation becomes a sparse linear system, making the implementation straightforward, using multigrid methods such as the conjugate gradient [10, 25].

A non-reactive substance like smoke is advected at the same time with the fluid:

$$\frac{\partial a}{\partial t} = -(u \cdot \nabla)a - \alpha_a a + S_a \tag{5}$$

where α_a is a dissipation rate and S_a is a source term. Since Eqs. 3 and 5 have identical structure, the same source code can be reused.

3. Overview of Our Method

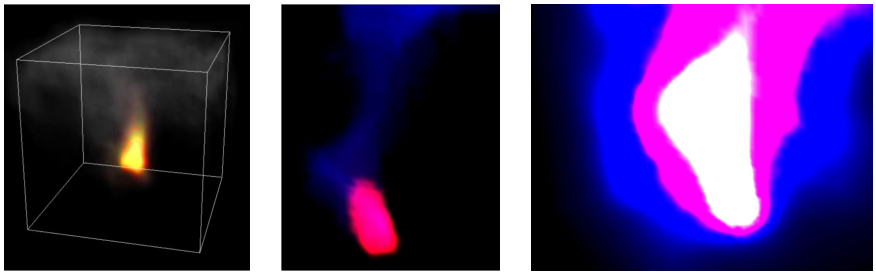
In our model, we transport the fuel and exhaust gases with the motion of the air, creating a dynamic 3-gas system. Heat is also transported with the flow of the air, enabling us to model heat distribution inside the computational domain more accurately than any other known fire model. Heat affects the buoyancy forces and changes the flow accordingly. Sufficient heat and an appropriate mixture of the air and fuel gas creates a combustion reaction, releasing smoke and more heat into the system. Under the correct conditions, a stable flame front is produced. With insufficient heat, oxidizing gas, or fuel, the flame is not sustainable, and dies out, as one would hope.

Transporting the heat within the system allows us to pyrolyse material inside the computational domain. Thus, we can model self ignition of the material even if it is away from the flame. Among the key assumptions and simplifications we make are:

- The fuel gas is uniform. That is, we assume that fuel is either burned or unburned, that any combustible byproducts behave exactly the same as the original fuel, and that the amount of heat produced is a function of just the amount of fuel burned.
- We assume that the viscosity of the mixed gas system is negligible. Although different ratios of mixed gasses will have different viscosities, we assume that any effects of this are too small to be seen over a coarse grid.
- The compression of the gases as a result of combustion is ignored. Simulating a compressible fluid is significantly more difficult than an incompressible fluid, and it is unlikely that a simulation of compressible fluids could be achieved at interactive rates. Furthermore, since we are simulating basically stable flames (i.e. not explosive situations), it is unlikely that this compression will have a significant visual effect on the solution. The total amount of gas (fuel, oxidizer, and exhaust) in a unit volume is assumed to be constant.

- Heat and temperature are treated interchangeably. While this is reasonable for a single gas, a mixed-gas system would not necessarily behave this way, however we assume the difference is minimal.
- The pressure effects of gases at different temperatures are not modeled directly (see the bullet about compression). However, the most significant effects (buoyancy of hot air, spread of heat through the fluid) are modeled.

4. Simulating the Flame



a b c

Figure 3. Our 3-gas fluid system. (a) flame and smoke in a closed environment, (b) motion of the gas (red is the fuel and blue is the smoke), (c) temperature distribution around the flame.

4.1. Gas Motion

In the "Stable Fluids" model a non-reactive substance such as smoke is carried with the motion of the fluid. We use the same vector field solver, and use the resulting vector field to advect and diffuse three quantities, fuel gas *g*, exhaust gas (including smoke) *a*, and heat *T*.

$$\frac{\partial g}{\partial t} = -(u \cdot \nabla)g - \alpha_g g + S_g \tag{6}$$

$$\frac{\partial a}{\partial t} = -(u \cdot \nabla)a - \alpha_a a \tag{7}$$

$$\frac{\partial T}{\partial t} = -(u \cdot \nabla)T - \alpha_T T + K_T \nabla^2 \tag{8}$$

where K_T is a diffusion constant, a_x ia a dissipation rate and S_x is a source term. Notice that the dissipation values can differ for the three quantities. Thus, the characteristics of the motion of the fuel, smoke, and temperature can differ significantly, although all are carried by the motion of the single 3-gas fluid system.

For fuel gases a_g is small if not zero, whereas smoke has a larger a_a. S_g is initially empty, except the initial source flame. During the simulation as objects start burning, they too become sources. Smoke and temperature do not have source terms, since smoke and heat comes from the burning process, according to the amount of fuel and air consumed within one time step.

Temperature has a large K_T which simulates diffusion of the heat and small a_T value, diffusing fast but not dissipating within the system. Diffusion of the temperature field is solved with a similar approach used in [21], using an implicit integration step, which gives a sparse linear system when discretized.

The fluid system is modified by the presence of the fuel, exhaust gas, and temperature. Referring to Eq. 3, the external force, *F*, acting in any one cell is given by:

$$F = f_g(g_g + d_a) \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} + f_T(T - T_{amb}) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{9}$$

where f_g and f_T are positive constants controlling the force components based on gravity and temperature respectively, and T_{amb} is the room temperature. Hot air will thus rise, and cold air fall, creating convection currents necessary to give the correct flame shape. The smoke and fuel gas will tend to fall, though this effect is usually not very noticeable (i.e. f_g should be fairly small).

4.2. Combustion

The combustion (burning) in a cell is defined by:
if $T > T_{threshold}$

$$C = r\ min(d_A, b\ d_g) \tag{10}$$

$$\frac{\partial d_g}{\partial t} = -\frac{C}{b} \tag{11}$$

$$\frac{\partial d_a}{\partial t} = C\left(1 + \frac{1}{b}\right) \tag{12}$$

$$\frac{\partial T}{\partial t} = T_0 C \tag{13}$$

where d_x is the density, *r* is the burning rate ($0 < r \leq 1$), *b* is the stoichiometric mixture (the amount of air required to burn one unit of fuel), $T_{threshold}$ is the lower flammability temperature where burning can occur, and T_0 is the output heat from the reaction. The burning rate is the percentage of the gas that can be burned in a second. Oxygen density d_A is defined as

$$D = d_A + d_g + d_a \tag{14}$$

where *D* is the total amount of gas in each cell, which is constant. If there is no fuel or smoke then it is filled with oxygen, and vice versa. Note that although air is not all oxygen, this can be easily accounted for by adjusting the stoichiometric mixture appropriately. At this stage we only have modeled one reaction which is controlled with the reaction ratio.

Conceptually, the combustion inside the cell is controlled with 4 parameters. $T_{threshold}$ is straightforward. *b* controls oxygen requirement of the combustion and can be used to model different reactions. T_0 controls the intensity of the reaction. *r* controls how quickly the gas can be burned. The heat output coming from a combustion cell can be sufficient to start a reaction in a neighboring cell. Or it might not be sufficient and the combustion reaction cannot continue, extinguishing the flame. Using these parameters, the shape and the stability of the flame can be controlled. Examples are shown in Section 7.

4.3. Creating a Flame

To achieve a steady flame, we simulate the effects of ignition. Initially temperature is constant and low and there is no smoke or fuel gas in the system. Initial fuel sources start to provide fuel gas to the system. Note that this fuel source may be a gaseous source, such as a gas jet, or may be the result of pyrolysis from a solid, such as a log (see section 5). Since there is no reaction yet, all the motion comes from the initial velocity of the gas coming from the fuel source, if there is any. Fuel gas coming out of the source is advected within the system. The fuel gas is ignited either by reaching a sufficient temperature, or by a special "ignition" step. In the ignition step, we temporarily set $T_{threshold}$ to 0 for the duration of the time step, simulating an ignition of the fuel (alternatively, we could increase the temperature in one cell for one time step). Assuming there is a mixture of fuel and air in a cell, combustion will occur. This combustion increases the heat in the system and can provide enough heat to the neighboring cells to continue the burning process on the next time step. Hot air rises due to the buoyancy force, and soon the flame stabilizes. Smoke is produced as a by-product of the combustion, and tends to rise with the hot air produced by the flame.

There are several possible reasons a flame might not stabilize. If the fuel and air has not mixed at ignition, no burning can occur. If an insufficient amount of fuel is present, too little heat may be generated to sustain the burning. Likewise, if the output heat from the reaction is too low, the heat produced might not be enough to propagate combustion and thereby sustain a flame.

5. Representing Solids and Flame Spread

Solid materials inside the computational domain are voxelized and the corresponding grid cells in the computational domain are marked as filled. This filled/empty information is used in flow calculations [21, 7]. In the advection step, the particle tracer hits and stops on filled grid cells, and in the diffusion step the filled grid cells are assigned the same amount of density as their empty neighbor, making the incoming and outgoing density flows equal. If a filled grid cell has more than one empty neighbor, average values are used, but it is possible for material one voxel thick to "leak".

Each filled grid cell is a potential fuel source. Active fuel sources emit fuel gas density into the neighboring unfilled cells at every time step. Potential fuel sources can later self-ignite, becoming active fuel sources. Every filled grid cell has a pyrolysis temperature, which is a material property (nonflammable materials simply have an arbitrarily high pyrolysis temperature). When a filled grid cell reaches its pyrolysis temperature, it becomes a fuel source. Since the self ignition temperature of the fuel is gener-

ally much smaller than the pyrolysis temperature, if there is enough air around, the resulting fuel starts burning once it mixes with air.

As mentioned before, we maintain a temperature value at each grid cell. Heat propagation is modeled within a solid object using the well-known heat flow equation:

$$\frac{\partial T}{\partial t} = k\nabla^2 T \tag{15}$$

where *k* is the diffusivity constant for the material. For the unfilled cells, heat both diffuses through the system and advects along with the gas flow. Thus, we have temperature values at the boundary between the solid and the air. These temperatures serve as boundary conditions when solving the differential equation.

Thus, we can transport heat through both the air (via the fluid solver) and the solid objects in the scene. Heat from the burning gas warms the air, which moves around, possibly heating up other solids in the scene. If these solids reach a high enough temperature, they will begin pyrolysis, becoming active fuel sources and spewing more gases, which will likely combust.

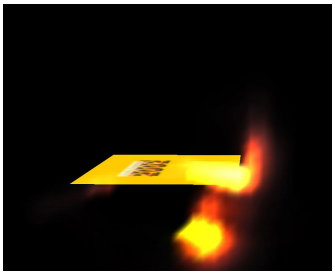


Figure 4. Burning solids

Note that our current implementation, does not model heat transfer inside the material. This limits our self-ignition demonstrations to paper-like fires, which we demonstrate in our video. It is easy to imagine an environment, however, where the solids would both conduct their own heat and possibly radiate heat to adjacent air voxels.

6. Visualization and Control

6.1. Visualization

The output of our simulation is voxelized data, so any volume rendering approach can be used for visualization. For interactivity, we have implemented a hardware based visualization scheme similar to that used in the Stable Fluids approach [21], but without self-shadowing. Voxels are replaced by a semitransparent polygon, where the level of transparency is determined by the density in the voxel, and the color and transparency interpolation is done by the graphics hardware. For this paper, we show the fuel gas in yellow and the flame front, where the reaction occurs, in red. The intensity of the color red comes from the amount of the fuel combusting in that voxel, hence simulating the energy coming out of the reaction inside the cell. Note that a different effect could easily be obtained by using a different color scheme. For example, a natural gas flame could be simulated by making the fuel gas completely

transparent, and the combustion region blue. Generally, we prefer not to show most of the smoke, making it almost transparent, so as not to obscure the flame. As the objects near their pyrolysis temperature their color is darkened to simulate charring [19].

The focus of our work has been on developing an interactive model, hence the visual quality is of secondary concern. Although the approach described works well for interactive visualization, higher quality images may be desired. If the restriction on interactivity is released, a larger grid size can also be used, giving more complex flame behavior. A low grid resolution and large time step interactive simulation can be used to choreograph the scene and then a smaller time step-larger grid solution can be rendered for better images.

6.2. Control of Flame Shape

In order to simulate different burning reactions, it is important for an animator to have some control over the flame shape. The shape of the flame can be controlled by modifying a few key parameters:

- Turbulence controls the amount of additional random motion added to the fluid motion field. If zero, the flame will eventually stabilize in a fixed shape. Introducing turbulence causes the flame to have a more realistic motion.

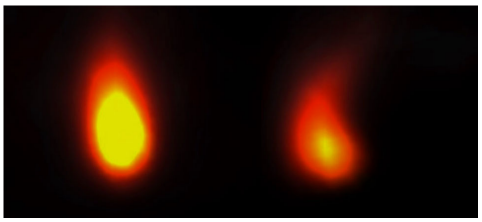


Figure 5. The flame at left experiences no turbulence whereas the right flame does. The flame at left will remain in that shape, while the one on the right will change over time.

- Burn rate gives the amount of fuel that can be burned in one second, as a percentage of the maximum amount of fuel that can combust (i.e. is at an adequate temperature and is mixed with air appropriately). Large (=1.0) burning rates tend to create very thin and sharp flames, since fuel is usually consumed soon after mixing with air, while slower burning rates create larger and more blurry flames.

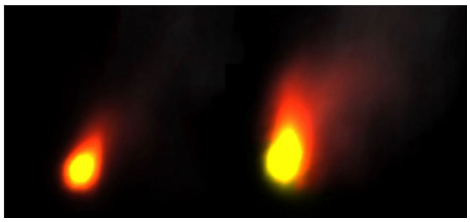


Figure 6. The flame at left has a faster burning rate than the one at right.

- Air use (i.e. the stoichiometric mixture) describes how much air is needed for the fuel to combust. This can be used to model different combustion reactions, reactions requiring more or less oxygen.

  Generally, as air use increases, the flame size becomes much larger and more spread out (similar to a low burn rate), since the gas must spread and diffuse more in order to mix with enough air to combust.

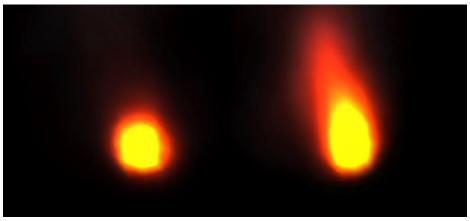


Figure 7. The combustion at left requires less air, than the flame at right to burn the same amount of fuel.

- Output temperature controls the intensity of the combustion. Lower output temperatures can lead to flames that are not self-sustaining or that let some fuel go unburned. High output temperature can cause flames to spread rapidly, ensuring full combustion and inducing self-ignition in other regions.

- Ignition threshold sets the temperature needed for fuel to combust. A lower ignition threshold is used to model highly flammable gases, while a higher ignition threshold can model almost nonflammable gases. Relating this to the pyrolysis temperature of any solids helps to simulate very flammable or nonflammable materials.

Although other parameters can be modified as well (e.g. the diffusion rate for temperature), these are more fundamental to the simulation and will tend to change the simulation as a whole, rather than just allowing simulation of different types of burning. They also tend to be less predictable in their effects.

Our examples and descriptions have focused on diffusion flames, where pure fuel is released that gradually mixes with air. These are the common flames seen when objects burn. We note, however, that our method can just as easily simulate premixed flames, where the air and fuel are mixed when introduced to the environment.

7. Implementation and Examples

We have implemented the method presented above and tested it on a number of examples. Our 3-gas method for simulating fire depends on an underlying fluid flow solver; our implementation is based on the Stable Fluids solver. Given a fluid solver, implementing our method on top of it is straightforward. Special care should be taken in handling the boundary conditions and objects inside the computational domain. Thin objects (one voxel thick) tend to ”leak” densities, but additional padding (making them at least two voxels thick) solves most of the problem.

Although the descriptions and examples given here are in terms of three dimensions, all of the simulation aspects can apply to a 2D system, as well (Fig. 8). The 2D calculations are significantly faster,

and thus a much finer simulation grid can be used. Although the results from such a system are clearly not physically realistic, the visual appearance of ”flames” produced in this way can be quite convincing. A sample application, called ”Combustion Paint” can be used by an artist to ”paint” 2D fireballs interactively (Fig. 8).

Figure 8. At left, a flame produced by the 2D simulator. At right, a scene choreographed using an early 2D model with "Combustion Paint", where the user can add fuel interactively

Our simulations are run on an ordinary PIII 850Mhz machine with 256MB of RAM and a 32MB GeForce2 card. We are able to simulate a flame interactively (around 20 fps) for an environment defined on a 20^3 grid, including internal boundaries and self ignition. Note that for the video and most images shown here, we have used a 60^3 grid, for increased visual quality. Simulation on this grid was at the rate of 0.3 fps (3 sec per frame) in 1280x960 resolution. Note that the simulation time is directly proportional to processing speed. With the continuing increases in processor speeds, even these larger grids should soon be feasible at interactive rates on a common PC.

The figures 9, 10 and 11 demonstrate several features of our implementation. Figure 9 shows the progression of a flame, the gas motion, and the temperature distribution over time. Figure 10 shows the gradual filling of an environment with smoke. Figure 11 demonstrates flame spread.

An accompanying video gives a better demonstration of our simulation. The video shows the temperature distribution, advection of gases and velocity field, and rendering of a simple flame; demonstrates the simulation at interactive rates; presents the effects of changing various parameters; demonstrates flame extinction from lack of heat output, lack of adequate air/fuel mixing, and lack of air; and demonstrates spread of flame by temperature transport.

8. Conclusions

We have presented a physically based approach for interactive simulation of fire. In contrast to previous approaches, our method uses fluid simulation to model the motion of the fuel gases, oxidizing gases, exhaust gases, and heat that are necessary components of a flame simulation. Some of the previous approaches simulate parts of the physical process, but usually use computationally expensive approaches. Our approach is capable of capturing the properties of simple flames together with the motion of the hot air/smoke and the temperature distribution within the computational domain at interactive rates. The approach can also handle several realistic effects such as propagating flame to nonadjacent objects (self-ignition) and extinction of flames by using up all available air. All of this is done at interactive speeds. We have implemented our approach, using a fluid flow solver based on the Stable Fluids approach of [21, 7], and demonstrated its application to a number of examples. We have created an

interactive hardware-based OpenGL renderer that, while not producing "eyecatching" visualizations, is able to show the strength of our model.

There are numerous avenues for expanded research on this topic. Four directions of future work that we intend to explore are the following:

- Though we have presented an approach for modeling heat transport within the solid objects, we have not implemented this yet (though it should be straightforward to do so). Along with this, a more advanced model of fuel output and consumption within solids could be used to simulate charring and decomposition of the objects themselves over time.

- Our assumption that the sum of the air, fuel, and exhaust gas in a cell is constant is somewhat unrealistic. Treating air as independent of fuel and exhaust will yield more accurate results, particularly for special cases such as consuming all of the air in the room. However, making such a change may necessitate accounting for changes in pressure or gas compression, which may cause the simulation to become too slow.

- Although the parameter values we discuss provide a great deal of control over the flame, it would be useful to provide a more intuitive and complete set of controls for use by animators. For the same reason, incorporating our implementation into a higher-quality rendering system (e.g. a volume ray-tracer, light emission model) would allow better pictures to be produced.

- Visual quality has not been the focus of this work, and thus there are many ways to improve the visualization. Among the improvements we would like to make are incorporation of a light emission model with a photon map renderer, adding (as a postprocess) a flame skeleton [1] or particle-based [20] approach to provide fine detail, and creating a better model for the luminance via thermal radiation of unburned fuel and actual combustion. We believe that incorporating such rendering improvements will result in images that are competitive with any other flame visualization.

References

[1] P. Beaudin, S. Parquet, and P. Poulin. Realistic and controllable fire simulation. *In Proceedings of Graphics Interface 2001*, pages 159–166, 2001.

[2] R. Borghi, M. Destriau, and G. De Soete. *Combustion and flames, chemical and physical principles*. Editions Technip, Paris, 1995.

[3] R. Bukowski and C. Séquin. Interactive simulation of fire in virtual building environments. *In Proceedings of SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series, ACM*, 1997.

[4] N. Chiba, K. Muraoka, and M. Miura. Two dimensional visual simulation of flames, smoke and the spread of fire. *The Journal of Visualization and Computer Animation*, (5(1)):37–54, 1994.

[5] A. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, (2):12–26, 1967.

[6] C. Detienne. *Physical Development of Natural and Criminal Fires*. Charles C. Thomas, 1994.

[7] R. Fedkiw, J. Stam, and W. Jensen, H. Visual simulation of smoke. *In Proceedings of SIGGRAPH 01, Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 15–22, 2001.

[8] N. Foster and D. Metaxas. Realistic animation of liquids. *Graphical Models and Image Processing*, (58(5)):471–483, 1996.

[9] N. Foster and D. Metaxas. Modeling the motion of a hot turbulent gas. *In Proceedings of SIGGRAPH 97, Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 181–188, 1997.

[10] W. Hackbusch. *Multi-grid methods and applications*. Springer Verlag, Berlin, 1985.
[11] M. Inakage. A simple model of flames. *In Proceedings of CG international*, pages 71–81, 1990.
[12] W. Jones, W, P. Forney, G, D. Peacock, R, and A. Reneke, P. A technical reference for cfast: An engineering tool for estimating fire and smoke transport. 2000.
[13] T. Kajiya, J and P. Von Herzen, B. Ray tracing volume densities. *In Computer Graphics (Proceedings of SIGGRAPH 89)*, (18(3)):165–174, 1984.
[14] A. Lamorlette and N. Foster. Structural modeling of natural flames. *In Proceedings of SIGGRAPH 02, Computer Graphics Proceedings, Annual Conference Series, ACM*, 2002. to appear.
[15] D. Nguyen, P. Fedkiw, R, and W. Jensen, H. Physically based modeling and animation of fire. *In Proceedings of SIGGRAPH 02, Computer Graphics Proceedings, Annual Conference Series, ACM*, 2002. to appear.
[16] Q. Nguyen, D, R. Fedkiw, and M. Kang. A boundary condition capturing method for incompressible flame discontinuities. *Journal of Computational Physics*, (172):179–198, 2001.
[17] H. Perlin, K. An image synthesizer. *In Computer Graphics (Proceedings of SIGGRAPH 85), ACM*, (19(3)):287–296, 1985.
[18] H. Perlin, K and M. Hoffert, E. Hypertexture. *In Computer Graphics (Proceedings of SIGGRAPH 89), ACM*, (23(3)):253–262, 1989.
[19] H. Perry, C and W. Picard, R. Synthesizing flames and their spreading. *In Proceedings of Fifth Eurographics Workshop on Animation and Simulation*, pages 105–117, 1994.
[20] T. Reeves, W. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, (2(2)), 1983.
[21] J. Stam. Stable fluids. *In Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 121–128, 1999.
[22] J. Stam. Interacting with smoke and fire in real time. *Communications of the ACM*, (43(7)):77–83, 2000.
[23] J. Stam and E. Fiume. Turbulent wind fields for gaseous phenomena. *In Proceedings of SIGGRAPH 93, Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 369–376, 1993.
[24] J. Stam and E. Fiume. Depicting fire and other gaseous phenomena using diffusion process. *In Proceedings of SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 129–136, 1995.
[25] N. Swarztrauber, P and A. Sweet, R. Efficient fortran subprograms for the solution of separable elliptic partial differential equations. *ACM Transactions on Mathematical Software*, (5(3)):352–364, 1979.
[26] J. Takahashi, H. Takahashi, and N. Chiba. Image synthesis of flickering scenes including simulated flames. *IEICE Transactions on Information Systems*, (E80-D(11)):1102–1108, 1997.
[27] WebSite. *Online Class Notes, CPE630, http://eyrie.shef.ac.uk/will/eee/*. University of Sheffield, 2002.
[28] D. Yngve, G, F. O'Brien, J, and K. Hodgins, J. Animating explosions. *In Proceedings of SIGGRAPH 00 Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 29–36, 2000.
[29] S. Yoshida and T. Nishita. Modelling of smoke flow taking obstacles into account. *Proc.of the 8th Pacific Conference*, pages 135–144, 2000.

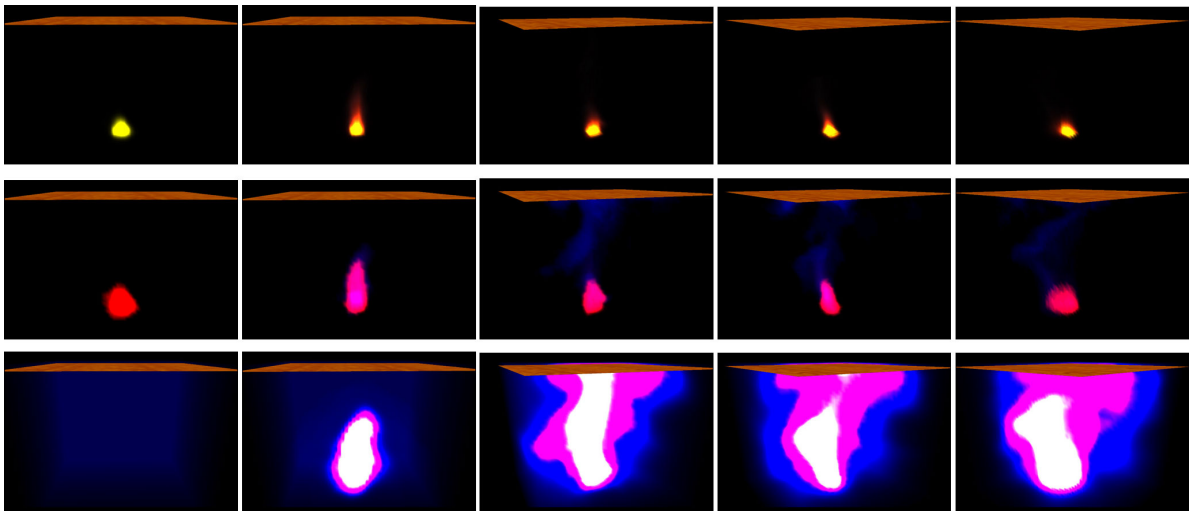


Figure 9. Simulation of a flame over time. The top row shows a visualization of the flame. Yellow areas are the fuel gas, and red areas indicate combustion (the flame itself). The middle row shows the motion of the fuel and exhaust gases. Fuel gas is shown in red, while exhaust gas (including smoke) is shown in blue. The gas motion is determined by a fluid simulation. The bottom row shows the temperature distribution. Heat is released by the combustion process and diffuses through the air. The temperature gradient causes convection currents (hot air rising, cool air falling) in the environment, causing the flame to rise upward

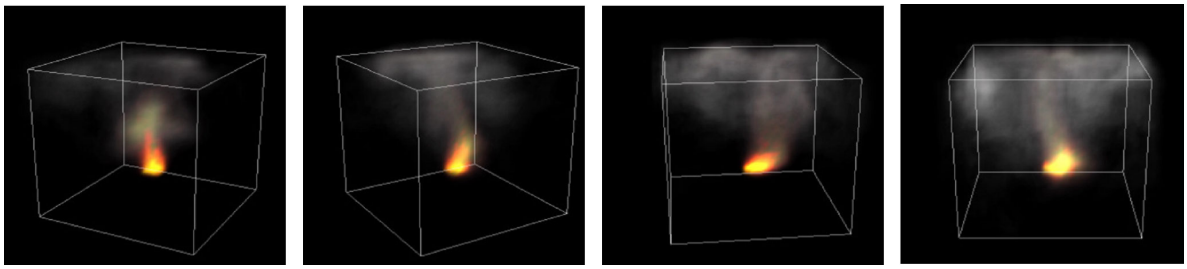


Figure 10. A simulation of a flame in a closed environment. The combustion releases smoke which fills the environment

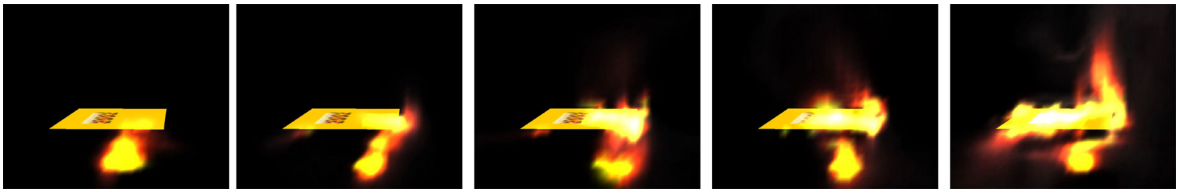


Figure 11. Self ignition of materials near a flame. As heat produced by the combustion rises, the paper begins pyrolysis, releasing fuel. This fuel also combusts, raising the temperature further and causing the flame to spread.