

# Interactive Simulation of Burning Objects

Zeki Melek  
melekzek@neo.tamu.edu  
Texas A&M University  
Computer Science Department

John Keyser  
keyser@cs.tamu.edu  
Texas A&M University  
Computer Science Department

## Abstract

*In this paper we describe a fast and interactive method for simulating and controlling the combustion process together with the decomposition of burning solids. The combustion and decomposition processes are integrated to create a physically-based model that runs at interactive rates.*

## 1 Introduction

We present a method for simplified modeling and visualization of fire that is easy to implement and works at 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. 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. The decomposition is defined as a moving boundary problem, modeled using level sets. This allows us to easily model the complex, topology changing decomposition process.

## 2 Flame Modeling in Computer Graphics

Most recent works on modeling fire, including Chiba et al. [2] and Takahashi et al. [13], use a gridded representation of the space, where each grid cell contains a certain amount of fuel gas and heat. Perry and Picard [8] represent the flame front by a set of particles, adding new particles as the front expands. Stam and Fiume [10] use a map covering the object defining the amount of fuel and temperature on every point on the object. Beaudin et al. [1] use a similar but more accurate flame front technique to model the spreading of the flames.

Presented last year, [4] is a production model that combines a particle model with a flame skeleton, and [6] defines

the flame front as a moving boundary between two fluids and an expansion term is introduced to add fine scale detail without using any randomized term.

## 3 Simple Flame Model

### 3.1 3 Gas System

In our simple flame 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 graphical fire model we know of.

To model the air flow, we use a modified version of the Stable Fluids [9, 3] approach. We assume there is only one moving gas (air), which is inviscid, incompressible and constant density. The fluid motion (air) solution is applied to advect three quantities: fuel gas  $g$ , exhaust gas (including smoke)  $a$ , and heat  $T$  [5]. Since we treat cells as having constant density, we can use heat and temperature interchangeably. Some cells (those containing objects to be burned) are marked as filled. Finally, a vorticity confinement term [11, 3] is used to compensate for the numerical dissipation in the system.

### 3.2 Combustion Process

We simulate the combustion process by combining fuel and oxidizer in a cell, creating additional exhaust gas in its place. If the heat in the cell is beyond the combustion threshold, we decrease the level of fuel gas, increase the level of exhaust gas, and introduce additional heat into the cell [5].

Note that we define the amount of oxygen in a cell indirectly, by specifying the fuel and exhaust gas in the cell. This simplification is not necessarily physically correct, but it enables us to use a single-fluid fuel solver and to model oxygen use and self-extinguishing flames by keeping track of only two gases. Note that although air is not all oxygen,

this can be easily accounted for by adjusting the stoichiometric mixture appropriately.

### 3.3 Heat Transfer

In our system we model heat transfer in three stages: heat transfer in the air, heat transfer between the air and the solid, and heat conduction within solids. This three-stage heat transfer model enables us to treat solids with varying thermal properties. Also, this model is flexible enough that different simulators can be used for any of the three stages, enabling us to work with slower but more accurate simulations of any portion, if desired.

Basic heat convection in air is handled using semi-Lagrangian advection, which simulates moving air currents carrying heat. On the other hand, radiative heat transfer is modeled as a diffusion process using implicit integration, enabling us to distribute the heat coming from the combustion process. The semi-Lagrangian advection scheme and implicit integration for the diffusion steps follow the same pattern as in the Stable Fluids approach[9].

Heat transfer between the solid and the air is handled separately. For each boundary voxel, we exchange heat to/from the adjacent unfilled “air” voxels. We use a simple method for heat exchange, finding the heat differential, and transferring a percentage between them. If additional accuracy is required for specific cases, a direct radiant heat term can be incorporated at the expense of interactivity by using a radiosity calculation for the heat coming directly from the visible part of the flame.

Finally, heat transfer inside solids is modeled by simulating diffusion using the heat equation,  $\frac{dT}{dt} = k\nabla^2 T$ , where  $k$  is a constant based on density, thermal conductivity, and specific heat of the material. Simulating heat diffusion is relatively simple, and an implicit integrator similar to that for heat diffusion in air (as above) can be used.

## 4 Burning Solids and Decomposition

### 4.1 Basics

The solids to be burned have two implicit representations, each represented by values along different regular 3D grids. Note that these grids do not need to be the same as the grid used to model air motion in the flame simulator. The first representation is as a signed distance field, which will be used to model decomposition of the object, and the grid resolution depends on how detailed we want our initial/intermediate object boundaries to be. The second representation is the amount of solid fuel in the cells, initially based on the percentage of the cell volume occupied by the solid. These representations can be generated from

any initial geometric definition, with a polygonal boundary representation being the most common.

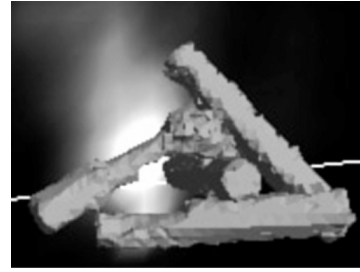


Figure 1. Decomposing solid material

Although the two representations are equivalent at the start of the simulation, we intentionally allow them to diverge during the simulation. The solid fuel amount represents how much solid fuel we still have available at the cell. On the other hand, the distance field represents the decomposed solid including both the solid fuel left and the residue (ashes) in that cell. During simulation we consume solid fuel, but leave something behind in form of residue, hence the different behavior of the two representations.

### 4.2 Pyrolysis

We define a pyrolysis temperature for every cell in the fuel grid that is filled. Recall that pyrolysis is the process by which a solid emits combustible gases. Once the cell reaches the pyrolysis temperature a pyrolysis process is applied at every simulation time step. This temperature can be set low for volatile solids and arbitrarily high for nonvolatile solids.

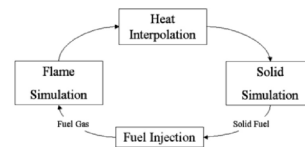


Figure 2. The pyrolysis computation.

During simulation, when a filled cell enters pyrolysis, fuel gas is released into this cell with velocity in a direction normal to the surface defined by the implicit distance representation. The precise amount of fuel released is a function of the temperature of the solid. The newly introduced fuel gas will participate in the fluid dynamics solution and flame simulation.

### 4.3 Decomposition as a moving boundary

The decomposition of the burning solid is modeled as a moving boundary in the distance field representation of the

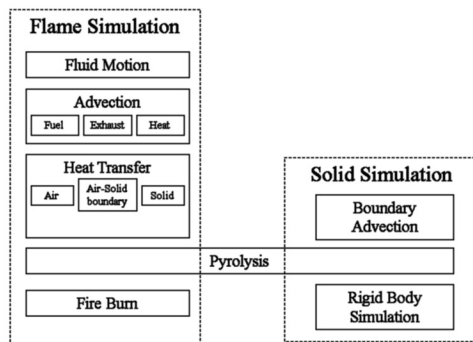
solid. The motion of the boundary is defined to take place in the direction of the fuel consumption gradient at a speed based on the rate of solid fuel consumption.

$$v = r_r \nabla \frac{\partial F}{\partial t} \quad (1)$$

where  $v$  is the velocity,  $F$  is the solid fuel set, and  $r_r$  is the ratio of burnable vs residue material inside the solid. This ratio is used to define how much residue is left in the solid once all the fuel has been used.

The distance field implicit representation of the solid can easily handle complex boundaries and topology changes, and also can be easily polygonized for interactive visualization [7]. We apply level set methods on the distance field representation to track the moving boundary. The semi-Lagrangian time-stepping scheme defined by Strain [12] works efficiently when adapted to this problem. By representing the solids on different grids than the fluid simulator, the decomposition process can execute in its own thread.

We make a significant simplification to the level set simulation, by skipping the reinitialization step, without introducing any visual artifacts. This is possible because our object boundary is always “shrinking” inward instead of moving arbitrarily. Although the grid values will no longer form an accurate distance field as the simulation progresses, they will still be accurate at the boundary.



**Figure 3. Flowchart of our simulation.**

The two simulations are coupled together by the pyrolysis step (transferring fuel from the solid representation to the fluid representation) and the heat interpolation (transferring heat information from the fluid simulator to the solid representation—see Figure 2. A chart showing the interactions between the major stages of the simulations is shown in Figure 3.

#### 4.4 Rigid body simulation

We implement a simple rigid body simulation by first finding separated pieces. This is done by marking grid cells

(in the distance field representation) of fixed portions of the solid as “grounded”, then recursively marking cells that border grounded cells as grounded. Those not marked as grounded are considered separate. Separated pieces are the “active” objects whereas the rest acts as “passive” in the rigid body simulation. “Passive” objects are only used for collision and are static in the rigid body simulation, whereas “active” bodies do not participate in the decomposition process, until they stop and become “passive” again. This simple paradigm enables us to quickly approximate the effects of gravity on separated pieces in the implicit domain, but does not include internal stresses.

## 5 Implementation and Analysis

### 5.1 Implementation

We have implemented the method presented above and tested it on a number of examples. Our simulations are run on an ordinary P4 1.8GHz machine with 512MB of RAM and a 32MB GeForce2 card. We are able to simulate a flame interactively ( $\sim 20$  fps) for an environment defined on a  $20^3$  simulation grid. This simulation includes the simulation of air motion, burning and decomposition for one solid, and our simple visualization. The grids used to describe the solid are  $40^3$  for the distance field representation and  $20^3$  for the fuel set representation.

Our OpenGL based visualization system is intended primarily to demonstrate the simulation methods, and is *not* intended to generate production-quality images. This visualization is limiting, but it is useful for understanding the simulation effects at interactive rates.

### 5.2 Simplifications

Our methods make a number of simplifying assumptions, which lead to some shortcomings. We describe below some of our key assumptions, simplifications, and shortcomings, along with the reason we accept them.

- *Single fluid model.* Although air, fuel gas, and exhaust gases will have different properties, we choose to model the system with a single fluid. We therefore do not capture properties such as the mixing of fluids of different viscosity. However, our model still captures the dominant large-scale motion of heated air.
- *Constant density air.* Our assumption that the sum of the air, fuel, and exhaust gas in a cell is constant is somewhat unrealistic, and the effects of combustion are not directly modeled. However, accurately simulating effects such as changes in pressure or gas compression would likely cause the simulation to become far too slow.

- *Decoupled heat transfer.* Separating heat transfer into three stages, rather than using a unified simulation is somewhat inaccurate. However, the inaccuracies do not appear significant, and the decoupling allows us to use different solvers of varying accuracy.
- *Simulating heat in fluid domain.* By simulating heat transfer in only the fluid domain, heat can diffuse farther into a solid than would be possible if heat were modeled in the representation of the object itself.
- *Simplified decomposition model.* The actual physical decomposition process of burning objects is quite complicated and would involve detailed modeling of the solid structure at a very low level, however it is impractical to attempt to model such a process in an interactive system.
- *Simplified solid simulation.* We have only modeled simple rigid body motions, and have not modeled internal stresses and strains on the objects.

### 5.3 Comparison with other methods

The two primary distinctions between our method and other recent methods are (a) that it runs at interactive rates, and (b) that it simulates the entire burning process.

The work of Nguyen et al.[6] achieves probably the best physical-simulation-based results. Their method captures the flame shape, including fine-scale details, by simulating the interface between two fluids. However, it does not model combustion, and does not address issues such as heat transport through the entire simulation domain. In our simulation, on the other hand, heat, generated by a simple model of the combustion reaction, is one of the main driving forces. They inject fuel from the solid into the fluid simulation system, however, they make no attempt to simulate any of the other effects within the solid - fuel consumption, object decomposition and heat distribution within the solid.

Finally, their simulation and rendering times, though producing more detailed results, are orders of magnitude longer than ours.

## 6 Conclusion

Our system integrates the fire and combustion process together with the ignition, burning and decomposition of solids. Although we have made a number of simplifying assumptions, our system is capable of modeling basic flame and decomposition behavior. This occurs at interactive frame rates on standard PC hardware. The simulation model we have presented is flexible, allowing for easy replacement of simulators in any one part of the model, fairly independently of the other portions of the model. There

are several avenues open for future work. For example, if we sacrifice interactivity, we can have more detailed models for both flame simulation and solid decomposition. In particular, we believe an interactive simulation of extended rigid body motions and limited internal stress calculations will be a challenging task.

## 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] 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.
- [3] 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.
- [4] A. Lamorlette and N. Foster. Structural modeling of natural flames. *In Proceedings of SIGGRAPH 02, Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 729–735, 2002.
- [5] Z. Melek and J. Keyser. Interactive simulation of fire. *Pacific Graphics 2002 Proceedings*, pages 431–432, 2002.
- [6] 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*, pages 721–728, 2002.
- [7] P. Ning and J. Bloomenthal. An evaluation of implicit surface tilers. *IEEE Computer Graphics and Applications*, pages November 33–41, 1993.
- [8] 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.
- [9] J. Stam. Stable fluids. *In Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series, ACM*, pages 121–128, 1999.
- [10] 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.
- [11] J. Steinhoff and D. Underhill. Modification of the euler equations for vorticity confinement: Application to the computation of interacting vortex rings. *Physics of Fluids*, (6):2738–2744, 1994.
- [12] J. Strain. A fast modular semi-lagrangian method for moving interfaces. *Journal of Computational Physics*, pages 165:512–536, 2000.
- [13] 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.

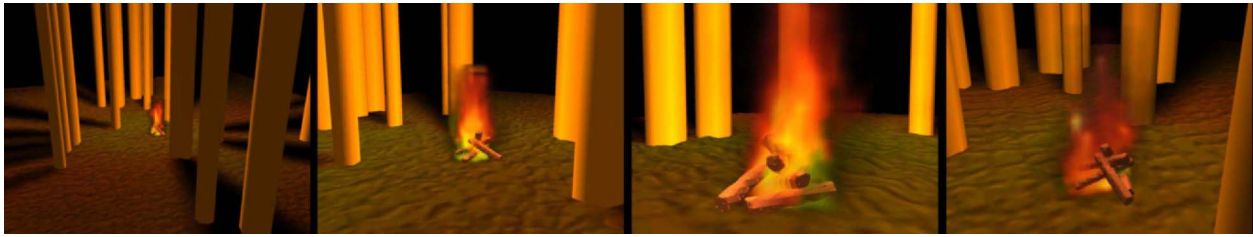


Figure 4. Camp fire

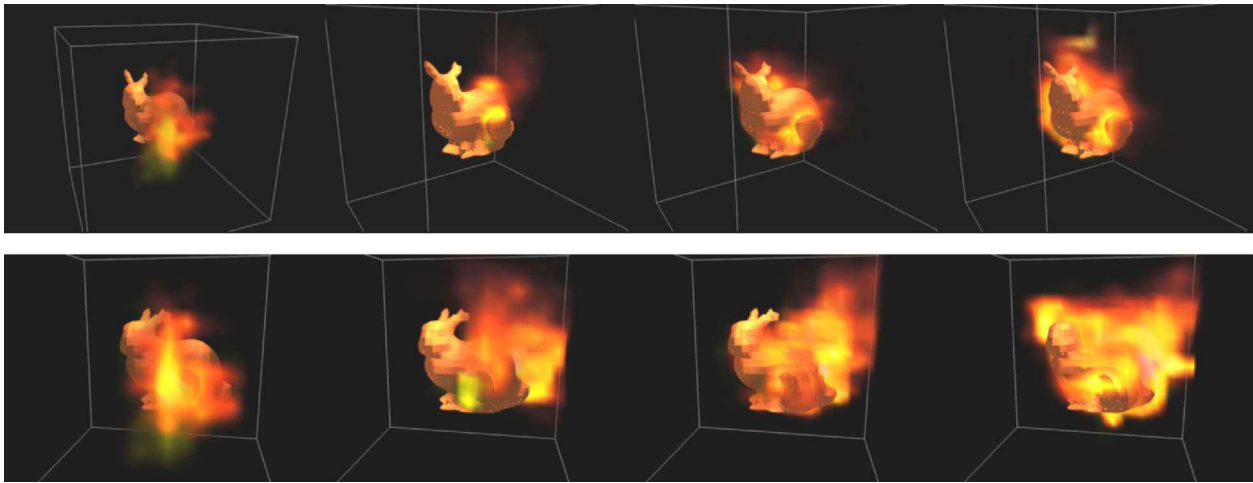


Figure 5. Two different solids: A slow fuel release (top) and a fast release (bottom)

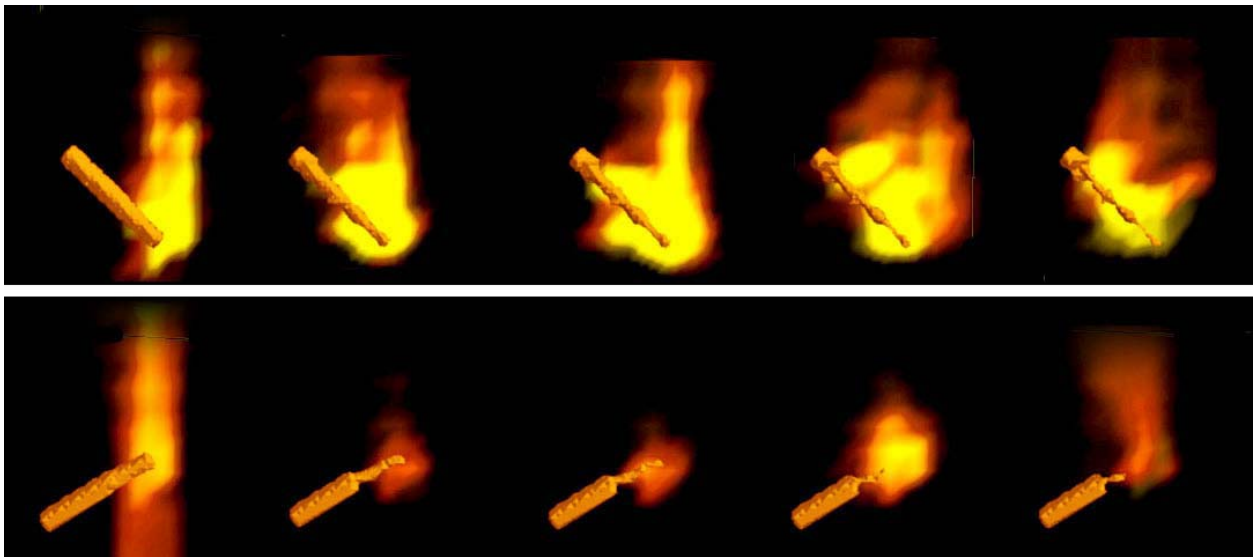


Figure 6. Burning matches