Modeling Decomposing Objects under Combustion

Zeki Melek
*

Texas A&M University
Computer Science Department

John Keyser
†

Texas A&M University
Computer Science Department

ABSTRACT

We present a simple yet effective method for modeling of object decomposition under combustion. A separate simulation models the flame production and generates heat from a combustion process, which is used to trigger pyrolysis of the solid object. The decomposition is modeled using level set methods, and can handle complex topological changes. Even with a very simple flame model on a coarse grid, we can achieve a plausible decomposition of the burning object.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically Based Modeling;

Keywords: physically based modeling, solid object decomposition

1 INTRODUCTION

There has been an increasing interest within the computer graphics community for simulation and visualization of natural phenomena such as water motion, smoke and fire. While recent fire models [3, 4, 2] are promising, none of them yet model decomposition of the burning solids and solids are rather treated only as a fuel source. Some recent visualization research has focused on tracking the motion of the flame front, yet it assumes the underlying geometry is fixed. We present a new method for modeling and visualization of decomposing objects under combustion. A coarse grid fire simulation is implemented to provide the heat distribution and fuel gas motion required by the model. 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.

Figure 1: Wooden bunnies, decomposing

*e-mail:melekzek@neo.tamu.edu
†e-mail:keyser@cs.tamu.edu

2 SIMPLE FLAME MODEL

We have implemented a three gas flame simulation that is simple, stable, and yet capable of simulating a wide and complex variety of flame and burning effects. This flame model provides the necessary heat information we use to drive the decomposition process, and receives the fuel gas that the burning object outputs. We keep track of three gasses over a grid: fuel, exhaust, and oxidizer. The air flow is modeled using a modified version of the Stable Fluids [7, 1] approach. We treat the 3-gas system as a single moving gas, which is inviscid, incompressible and constant density. The fluid motion is applied to advect three quantities: fuel gas, exhaust gas, and heat. Allowing heat to flow enables us to model heat distribution inside the computational domain. We simulate the combustion process by combining fuel and oxidizer in a cell, creating additional exhaust gas and additional heat. 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. Basic heat convection in air is handled using semi-Lagrangian advection, simulating moving air currents carrying heat. Radiative heat transfer is approximated as a diffusion process using implicit integration, enabling us to distribute the heat coming from the combustion process. Heat transfer inside solids is similarly modeled by simulating diffusion. For most objects being burned, this heat diffusion is quite slow, and constant through the object. For nonuniform material, we incorporate variations of thermal conductivity. Heat transfer between the solid and the air is handled separately.

3 BURNING SOLIDS

Every solid to be burned has two implicit representations. The first representation is as a signed distance field, a boundary representation of the solid at zero crossing, which will be used to model the surface of the decomposing object. The grid resolution we use depends on how detailed we want our initial/intermediate object boundaries to be. We will generate intermediate decomposition states using this representation. The second and complementary representation is the amount of solid fuel stored in the cells. This representation usually does not require as fine a grid resolution as for the distance field. The resolution of these representations can be adjusted for simulation behavior (both representations) and visual quality (distance field representation). Although the two representations are equivalent, being generated from the same initial boundary representation, at the start of the simulation, we intentionally allow them to diverge during the simulation. The different behavior of the two representations during decomposition simulation allows us to consume solid fuel but leave additional material behind in form of ash residue. Pyrolysis is the process by which a solid emits combustible gases. Once a solid cell reaches the pyrolysis temperature, a pyrolysis process is applied at every simulation time step. This temperature can be set low for volatile solids, arbitrarily high for nonvolatile solids, or vary through the solid to model mixed solids. During the pyrolysis step, some of the solid fuel is converted into fuel gas as a function of the temperature.

Figure 2: Time sequence of a decomposing solid (fire is not displayed for clarity)

4 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 (from the solid fuel density grid) at a speed based on the rate of solid fuel consumption.

$$v = r_r \, \nabla \frac{\partial F}{\partial t} \tag{1}$$

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 ash is left in the solid once all the fuel has been used.

The distance field implicit representation of the solid can handle complex boundaries and topology changes, and also can be easily polygonized for interactive visualization. We apply level set methods on the distance field representation to track the moving boundary [6]. We make a significant yet non-trivial simplification to the level set simulation, which results in much more efficient running times. We are able to skip the reinitialization step for the distance field, without introducing any visual artifacts. This is possible because our object boundary is always “shrinking” inward instead of moving arbitrarily. The two simulations, flame and decomposition, 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). At the beginning of the time step, heat is exchanged to/from the air. Afterward, the decomposition of each solid could be run in parallel in separate threads, since there is no interaction between them until the end of the time step. The decomposition process needs only to keep track of the derivative of the solid fuel representation to modify the distance field representation. At the end of the time step, the newly introduced fuel gas is passed to the flame simulator, together with the flow velocities.

5 Rigid body simulation

As objects decompose, disconnected pieces are created, which should be detected. The polygons created from the implicit representation (the visualization polygons) are used to detect such separations. Once the separate pieces are detected within the given solid, we split the volume and distance fields accordingly, creating two or more separate solid objects. This structure allows us to simulate the motion of individual pieces, while each still burns and decomposes inside its local computational domain. For collision detection, we use a set of particles placed at the vertices of the visualization polygons. Interpolation on the distance field of the other object directly gives us the approximate distance to the boundary.

6 Visualization of Decomposing Solid

The distance field representation is polygonized by an isosurface generation using tetrahedral decomposition of the boundary cells [5]. After each timestep, previous visualization polygons are discarded and new ones are created. This working scheme simplifies the implementation, yet the lack of frame-by-frame tracking prevents consistent texture coordinate inheritance. Instead, we used projection methods to map texture coordinates from the implicit form onto the visualization polygons. We use multiple textures to display wood in various stages of burning. We switch textures as the burning progresses, based on the amount of solid fuel released. A combination of heat and the *solid fuel left / solid fuel initially contained* ratio is used to advance the texture index. Here one should note that in our simulation we do not track the flame front geometrically; a flame front is not even defined. Yet our results still allow us to visualize the flame front as the solid progresses through textures, which is the outcome of the underlying simulation.

Figure 3: Textures used in decomposition

7 Conclusion

We present a model for decomposing solids under a combustion process. A simple fire simulation is also presented to simulate simple combustion and provide the heat to drive the pyrolysis process. Our system integrates the fire and combustion process together with the ignition, burning and decomposition of solids. The parameters available in our model give a great deal of control over both the way objects burn, and the way the resulting flames behave, and we can achieve a variety of complex physically-based effects.

References

[1] R. Fedkiw, J. Stam, and H. W. Jensen. Visual simulation of smoke. *Proc. of ACM SIGGRAPH ’01*, pages 15–22, 2001.

[2] Bryan E. Feldman, James F. O’Brien, and Okan Arikan. Animating suspended particle explosions. In *Proc. of ACM SIGGRAPH ’03*, pages 708–715, aug 2003.

[3] Kevin B. McGrattan, Howard R. Baum, Ronald G. Rehm, Anthony Hamins, Glenn P. Forney, Jason E. Floyd, Simo Hostikka, and Kuldeep Prasad. Fire dynamics simulator technical reference guide. *Tech. Rep. NISTIR 6783*, 2002.

[4] D. Nguyen, R. Fedkiw, and H. W. Jensen. Physically based modeling and animation of fire. *Proc. of ACM SIGGRAPH ’02*, pages 721–728, 2002.

[5] P. Ning and J. Bloomenthal. An evaluation of implicit surface tilers. *IEEE Computer Graphics and Applications*, pages November 33–41, 1993.

[6] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2002.

[7] J. Stam. Stable fluids. *Proc. of ACM SIGGRAPH ’99*, pages 121–128, 1999.