Proceedings of the 6th International Conference on the Foundations of Digital Games, pp. 256-258, Bordeaux, France, 2011.

Towards Fast, Believable Real-time Rendering of Burning Objects in Video Games

Dhanyu Amarasinghe
Dept. of Computer Science & Engineering
University of North Texas
DhanyuAmarasinghe@my.unt.edu

Ian Parberry
Dept. of Computer Science & Engineering
University of North Texas
ian@unt.edu

ABSTRACT

We present a framework for emulating the deformation and consumption of polygonal models under combustion while generating procedural fire. Our focus is on achieving the best visual effects possible while maximizing computation speed so that the processing power is available for other tasks in video games. We have implemented and tested our method on a relatively modest GPU using CUDA. Our experiments suggest that our method gives a believable rendering of the effects of fire while using only a small fraction of CPU and GPU resources.

Categories and Subject Descriptors

I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*hierarchy and geometric transformations, animation*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*animation*; I.6.8 [**Simulation and Modelling**]: Type of Simulation—*animation*

General Terms

Algorithms, Experimentation

Keywords

Hardware acceleration, volume rendering, freeform deformation, procedural, generation, fire modeling, CUDA.

1. INTRODUCTION

One way a new video game can make an impact on players is by increasing realism over and above what the player is accustomed to in other games. Replicating the details of a physical process such as fire can increase the believability of virtual worlds, and draw the player into willing suspension of disbelief. To date, developers mostly use model-swapping techniques to implement a crude level of model deformation by combustion. We introduce a technique for performing real-time emulation of a burning object while maintaining system performance.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
FDG '11, June 29–July 1, 2011, Bordeaux, France
Copyright 2011 ACM 978-1-4503-0804-5/11/06 ...$10.00.

The main part of this paper consists of 4 sections. In Section 2 we describe some previously published related work. In Section 3 we describe our representation framework for the internal deformation and the key features of such a strategy. Section 4 describes our approach to implementing the structural deformation framework. Section 5 contains a few notes on our CUDA implementation. For a more detailed version of this paper, see [2], and for images and video of burning objects from the CUDA implementation of our algorithm, see [1].

2. PREVIOUS WORK

Melek and Keyser [5, 7] discuss techniques that were used in selected object deformation due to fire, however, these methods are designed to maximize realism at the cost of performance. Sederberg and Parry [11] and Hsu, Hughes, and Kaufman [4] introduce some adaptive techniques of deformation. Müller and McMillan [9] discuss real-time techniques for deformation focussing on selected materials. Toivanen [12] discusses free deformation of meshes, but his technique is too computationally intensive for use in video games.

Nguyen and Fedkiw [10] introduce high quality flame simulations, but do not address object deformation. Wei and Zhao [14] use an approach similar to Melek, defining solids as a volumetric implicit field, but also do not discuss object deformation. Wei and Li [13] use splatting techniques that help to increase visual impact. Moidu, Kuffner, and Bhat [8] demonstrate an attempt to animate deformable materials such as paper, but they do not introduce complex heat transfer models. Although the paper did discuss the spring-mass model technique to emulate combusting surfaces, they focus on selective materials such as paper and cloth.

3. INTERNAL DEFORMATION

The geometry and topology of a burning object changes as heat spreads. Melek and Keyser [7] noted that there are multiple internal chemical reactions at various stages of the process, during which its properties may change from solid to liquid and from liquid to gas due to volumetric expansion caused by weakening bonds at the atomic level. The change in bond strength disturbs the stability of the internal forces between atoms. This causes the changes in the shape of the object's affected areas.

In the real world the temperature of a burning object changes over time and space. The elevated temperature generated in the model due to fire effects have a strong influence on the mechanical behavior of the object and conversely, the mechanical behavior of the given object influences the ther-

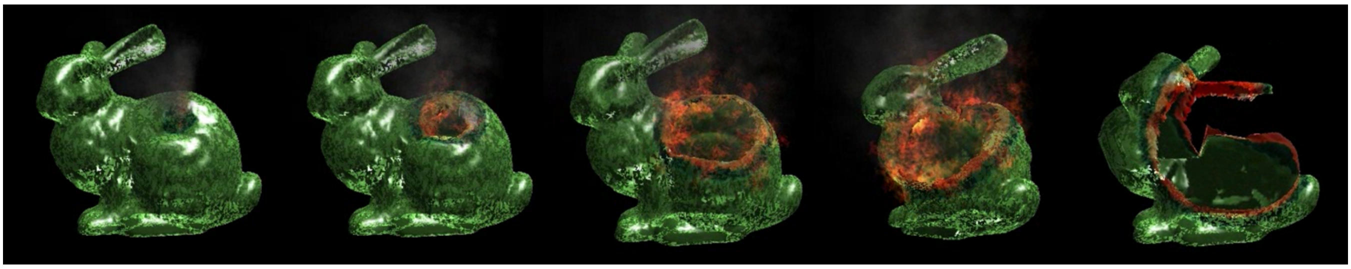

Figure 1: Animation frames for burning the Stanford bunny using our algorithm in real-time.

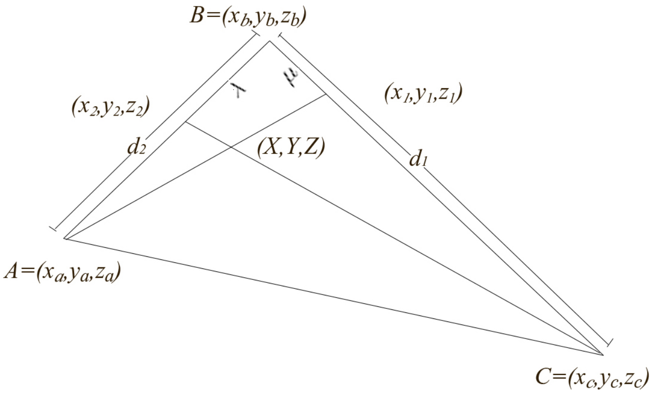

Figure 2: The deformation coordinates of a single triangle.

Name	Type	Description	Level
L	Integer	Flammability	Vertex
ε	$0 < \varepsilon < 1$	Meltability	Edge
β	$0 < \beta < 1$	Displacement scale	Block
ρ	$0 < \rho < 1$	Material density	Block
ϕ	$0 < \phi < 1$	Bond strength	Block

Table 1: Designer set constants.

mal response due to the thermal conductivity of the material. Heat transfer calculations depend on many parameters including environmental factors such as humidity (see Ang and Gumel [3]).

To speed computation we approximate the expansion of the heat boundary by calculating it around a single fixed point. This creates a roughly spherical but irregular heat boundary around the heat source. However, heat sources multiply when the flame distributes throughout the model. Heat spread over a given material depends on the *thermal conductivity* of that material (Melek and Keyser [6]), which indicates its ability to conduct heat. We model thermal conductivity using a *heat index* constant ψ. The value of ψ should depend on the size of the triangles used in the model and the material that the model is made from. We then use the following function to emulate an approximated heat boundary expansion: $R^2 = |\sin(\pi\Theta/\Delta r) + \sin(\pi\Theta) + \psi((x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2)|$, where $R = r + \Delta r$, the radius r is incremented by Δr in each Δt time period. The angle Θ is a random value in order to make the expanding heat boundary irregular in shape (otherwise the heat boundary will be perfectly spherical, which would appear unnatural). The location of the heat source is (x_0, y_0, z_0).

Internal deformation is achieved by displacement of the vertices of the model mesh. The position of each vertex will depend on three properties: vertex distance, gravitational force, and material index. While we may assume that material is a constant over large areas of the model, vertex distance and gravitational force are more complicated, and will be examined next.

We consider each vertex of the object to be analogous to an atom. The bond between two vertices of a triangle is taken to be inversely proportional to the distance between them. Vertex displacement is inversely proportional to bond strength, that is, directly proportional to distance, and scaled by material index.

We make use of the following designer-set constants, L, ε, β, ρ, and ϕ described in Table 1. The first is an integer, the remainder are real-valued constants between zero and one. All of the values can be made constant for the entire model, but in principle L can be different for each vertex, ε can be different for each edge, and β, ρ, and ϕ can be different for each block.

Suppose B is a vertex to be displaced in triangle ABC, where $A = (x_a, y_a, z_a)$, $B = (x_b, y_b, z_b)$, and $C = (x_c, y_c, z_c)$. B is to be displaced to (X, Y, Z), as follows: $X = (x_1 x_2 (y_a - y_c) + x_1 x_a (y_c - y_2) + x_c x_2 (y_1 - y_a) + x_a x_c (y_2 - y_1))/((x_a - x_2)(y_c - y_1) - (x_c - x_1)(y_a - y_2))$ (similarly for Y and Z), where $(x_1, y_1, z_1) = \mu C + (d_1 - \mu)B$, and $(x_2, y_2, z_2) = \lambda A + (d_2 - \lambda)B$.

Figure 2 shows the coordinates and parameters used in these equations. The values λ and μ are the contraction amounts along each edge due to heat. The lengths of BC and BA are d_1 and d_2 respectively. The points (x_1, y_1, z_1) and (x_2, y_2, z_2) are a μ and λ fraction respectively of the length along the edges (respectively BC and BA) of the triangle. The values μ and λ are displacement parameters for vertex B which measure the amount that the bond between B and its neighboring vertices is changed by temperature.

In addition we use a *displacement adjustment parameter* β to allow for the variation in triangle size from one model to another. ρ denotes a *material density index*. When both vertices of an edge are inside the heat boundary, bond strength is weaker by a factor of ϕ than when one vertex is outside of the heat boundary.

Burning objects are consumed by combustion, and combustion subsides when there is nothing left to consume. We model this with a *flammability* value L at each vertex. The counter decreases each time vertex displacement is processed. A the level counter of zero indicates that there are no consumable resources left at the vertex. The designer sets the initial flammability value for each vertex to mimic the effect of having different parts of the model constructed from physical materials of varying flammability such as wood or metal. λ is then defined to be $\beta\rho L/d_2$ if A is outside the heat boundary, and $\phi\beta\rho L/d_2$ otherwise μ is defined similarly, replacing d_2 with d_1.

Finally, among all of the external forces, gravity plays a major part in every physical based simulation. Let ε be a constant that represents the amount that the model melts due to heat, and $\vec{g}$ be the gravity vector. Then the effect of gravity is computed as follows: $Y = Y - \varepsilon\vec{g}$.

4. STRUCTURAL DEFORMATION

Deformation of a burning object can be caused by factors such as the expansion and weakening of the internal bonds, and the relative weights of cantilevered parts of the object. Exact calculation of these complex processes is costly. Therefore, we simplify the process by considering only the weight of a given point of the structure. The weight changes of the burning structure will occur due to consumption of the object by fire. Following Melek and Keyser [7], we divide the object into uniform blocks and treat each block as a single unit, propagating changes to neighboring blocks.

We start by constructing an oriented bounding box around our object, then decompose it into a grid of smaller axially aligned bounding boxes which we will call *blocks*. Deciding the number of blocks per model is up to the designer. Higher numbers of smaller blocks will make the effect more realistic at the cost of lower performance. We weight the blocks according to the number of vertices in them, and discard the empty ones of weight zero. We use block weight to model flame distribution, under the assumption that more vertices means more material, and therefore more flames.

We store for each block the amount of rotation, the midpoint of each box, the number of vertices, and the list of neighboring blocks. Since all the blocks are interconnected, a change to one block may affect all of the blocks in the model. To limit the computation required we apply changes to only immediate neighboring blocks, and rely on time to propagate the effects further.

5. CUDA IMPLEMENTATION

The images of burning models shown in this paper are screenshots from a CUDA implementation of our algorithm. The videos from which these frames are taken can be found on [1]. The flames are generated using 3000 fire particles and 1800 smoke particles. The model has almost 16K triangles. The animation runs at 70fps) on relatively modest hardware; An Intel®Core™2 Duo CPU P8400 @ 2.26GHz processor with an NVidia GeForce 9800 GTS graphics card.

6. CONCLUSION

We have proposed a method for the real-time deformation and consumption of a polygonal model during combustion by procedurally generated fire. We have focused on the performance with a reasonable amount of realism sufficient to trigger willing suspense of disbelief in the game player. We believe this our method is the first of its kind. It takes into account a variety of physical properties including material density indexes, material indexes, heat distribution, gravity, structural and internal deformation, and flame distribution.

7. REFERENCES

[1] D. Amarasinghe and I. Parberry. Fire, 2010. `http://www.eng.unt.edu/ian/research/fire/`.

[2] D. Amarasinghe and I. Parberry. Towards Fast, Believable Real-time Rendering of Burning Objects in Video Games. Tech. Report LARC–2010–04, Lab. for Recreational Computing, Dept. of Computer Science & Engineering, Univ. of North Texas, 2010.

[3] W. Ang and A. Gumel. A boundary integral method for the three-dimensional heat equation subject to specification of energy. *J. Computational and Applied Mathematics*, 135(2):303–311, 2001.

[4] W. Hsu, J. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. In *Proc. 19th Ann. Conference on Computer Graphics and Interactive Techniques*, pp. 177–184, 1992.

[5] Z. Melek and J. Keyser. An interactive simulation framework for burning objects. Tech. Report 2005-03-1, Dept. of Comp. Sci., Texas A&M Univ., 2005.

[6] Z. Melek and J. Keyser. Multi-representation interaction for physically based modeling. In *Proc. 2005 ACM Symposium on Solid and Physical Modeling*, pp. 187–196, 2005.

[7] Z. Melek and J. Keyser. Driving object deformations from internal physical processes. In *Proc. 2007 ACM Symposium on Solid and Physical Modeling*, pp. 51–59, 2007.

[8] S. Moidu, J. Kuffner, and K. S. Bhat. Animating the combustion of deformable materials. In *ACM SIGGRAPH 2004 Posters*, p. 90, 2004.

[9] M. Müller, L. McMillan, J. Dorsey, and R. Jagnow. Real-time simulation of deformation and fracture of stiff materials. In *Proc. Eurographic Workshop on Comp. Anim. and Sim.*, pp. 113–124, 2001.

[10] D. Q. Nguyen, R. Fedkiw, and H. W. Jensen. Physically based modeling and animation of fire. In *Proc. 29th Ann. Conf. on Computer Graphics and Interactive Techniques*, pages 721–728, 2002.

[11] T. Sederberg and S. Parry. Free-form deformation of solid geometric models. *ACM SIGRAPH Computer Graphics*, 20(4):151–160, 1986.

[12] J. Toivanen. A Non-Linear Mesh Deformation Operator Applied to Shape Optimization.

[13] X. Wei, W. Li, K. Mueller, and A. Kaufman. Simulating fire with texture splats. In *IEEE Visualization*, pp. 227–234, 2002.

[14] Y. Zhao, X. Wei, Z. Fan, A. Kaufman, and H. Qin. Voxels on fire. *Visualization Conference, IEEE*, p. 36, 2003.