Volume Graphics Workshop (2003)
I. Fujishiro, K. Mueller, A. Kaufman (Editors)

Volumetric Ablation Rendering

Hari Varadhan Klaus Mueller

{hari, mueller}@cs.sunysb.edu
Center for Visual Computing, Computer Science, Stony Brook University

Abstract

In this paper, we propose a physically-based method for simulating the process of ablation on volumetric models. We demonstrate the visual effect of ablative processes, such as a beam of heat emitted from a blow torch or a pencil of sand expelled from a sandblaster. Users are able to control ablative properties, such as energy propagation, absorption, and material evaporation, via a simple transfer function interface, while the effect of different beam shapes can be modeled by ways of weighting functions.

Continuous evaporation of material to expose interior object features can eliminate smooth object boundary layers required for good gradient estimation. To prevent this adverse effect, our method leaves the original volume intact and instead operates on a smooth energy volume. The renderer then uses the energy volume to determine the current, smooth object boundaries, for the opacity and gradient calculations, while the original volume provides the visual material properties, such as color and shading coefficients.

1. Introduction

The simulation of natural phenomena is one of the fundamental research areas in volume graphics [4][7][11][14][22]. In this paper, we will consider the effects of ablation. Ablation describes the process of erosion or vaporization that results when subjecting an object to the directed beam of a high-energy source. Example energy modalities include heat, such as a blow torch, or high-speed particles, such as a sandblaster. Apart from these more traditional tasks, ablation has also found many medical uses, such as ultrasound ablation for recanalizing occluded arteries in atherosclerosis, or radiofrequency ablation for heating and destroying cancerous tissue.

In the process of ablation, as heat energy impinges onto the material, the material responds accordingly by eroding, evaporating, or generally changing its state. Determining factors are the inherent energy the material requires to change its state as well as its ability to absorb the incoming energy. The ablative processes we model tend to have a localized nature, as opposed to a global one, due to the presence of a directed energy source. The material that is not in direct contact with the beam of energy, e.g. the heat, is virtually unaffected. This is unlike global phenomena, such as thawing [7] or weathering [4][14].

The physical mechanisms of material ablation are complex. Understanding these processes is a still a challenge, despite their common occurrence. Recent findings show that when a solid melts, it undergoes not just one phase change, as had been predicted previously, but two [3]. The ways in which heat propagates through the material, and from one type of material to another, are highly complex to model. To avoid such complicated and tedious mechanisms, phenomenological models are often used which yield similar results, but are not very related to the underlying physical process.

Instead of phenomological models, in this paper we choose to use a simulation that is more closely related to the actual physics of ablation, yet our approach does not claim to be completely physically accurate. It does not employ the Navier-Stokes equations to accurately model the transfer of heat through a medium [20], nor does it model phase transformations, such as the transition from solid to liquid or the qualitative decay of material under burning or melting. We also currently do not model the displacement of transitional products of the process, such as ash or smoke. Nevertheless, our method is able to cover the basic effects of ablation with convincing results, and it also allows users to exert control over the ablative processes. And, in contrast to Navier-Stokes solvers, our method is quite easy to implement. It is similar to volume rendering and runs at comparable computational complexity. In fact, it is straightforward to modify an existing volume renderer to implement our method. In the present work, we employ an existing volume renderer based on ray casting and convert it to deposit energy into the volume, where the amount of energy required to melt the material and the amount of energy the material absorbs can be controlled via user-provided parameters. Finally, a simultaneous rendering process allows users to visualize the ablative process while it is occurring.

Figure 1: *A chair evaporates under the influence of a directed heat source.*

An important issue in volume rendering is the need for smooth gradients at object boundaries. Without smooth gradients, the object will have a binarized appearance. Datasets obtained via volumetric scanning modalities, such as CT and MRI, generally already have smooth boundaries, due to the lowpassing effect of the acquisition process. However, volumes obtained by ways of voxelization have to be carefully anti-aliased by adding a smooth boundary layer [18] or a distance map [9]. If the ablative process proceeds by simple peeling of voxels, the smooth boundary will have been eliminated within a few iterations, giving way to the binary boundary of an excised object. This problem is heightened by the fact that ablation will yield a highly irregular boundary (which is in contrast to the planar boundary created by a cutting plane where all normals point in the same direction). Renderings of these objects will exhibit unnatural jagged edges and surfaces. Binary ablation techniques, such as morphological operators [7], suffer from similar problems. Hence, we would like to have an ablation process that subtracts material from the volume in a smooth fashion. Even further, we would like an ablation process where we can adjust the degree of smoothness, that is, the granularity of material removal. Then we could, for example, simulate the sandblasting of hard and soft objects. Finally, we would also like the ablation to be feature-sensitive, that is, to react differently for different material components of the object. For example, we may find it useful to burn off the flesh of a head model, but leave the skull intact. Or, in a virtual surgery training session, we may want to use a radiofrequency ablation simulation to zap tumor voxels with certain density from a medical dataset, but leave the surrounding tissue unaffected.

We would also like to retain the visual characteristics of the volumetric material. The look and feel of newly exposed material should be unaffected by the ablation process. In other words, the created smooth boundaries should not change a voxel's mapping into the color (and other) transfer functions. Therefore, we need to maintain a static voxel map into the color lookup tables. This map is naturally provided by the original volume. Thus, our algorithm uses a two-tier approach. The ablation occurs in a newly created volume that keeps track of the current state of the ablation process, while the voxel appearance map is provided by the original volume.

We shall now explain our technique in closer detail. First, in Section 2 we provide an overview on previous related work, while Section 3 will discuss the theoretical framework of our algorithm. Section 4 will give details on our specific implementation, and Sections 5 and 6 will follow with results, conclusions and future work, respectively.

2. Related Work

The process of natural weathering and thawing are somewhat related to this approach. In the modeling of ice thawing, the volume model is thought of being made of ice and is left in the open (warm) air to thaw. Fuijishiro and Aoki [7] use a mathematical morphology operator to simulate the effects of thawing. The mathematical morphology operator is a phenomenological modelling operator and is shown in their work to provide a very good approximation of the physical model. Further, to simulate the relegation of water on the base of the volumetric ice statues, a cellular automata mechanism is employed.

Dorsey et al. [4] model the weathering of stone by employing a simulation of the flow of moisture through the surface into the stone. Here, the model governs the erosion of material from the surface and the weathering process is confined to a thick crust on the surface of the volume. In an earlier paper, Ozawa and Fujishiro [14] use their mathematical morphology technique also for the weathering of stone. By applying a spatially variant structuring element for the morphology, they are able to simulate the stochastic nature of real weathering phenomena. A number of papers have recently appeared that use Navier-Stokes solvers [2] or advanced cellular automata methods (Lattice-Boltzmann) [23][24] to simulate the process of melting and flowing of viscous materials, as well as sand, mud and snow [19].

The present approaches have been using global heating or weathering effects, while in our work the evaporation or melting of the material is localized. This enables us to describe a method that is more related to the physical model than a phenomenological one, while still retaining good speed. Our framework is not designed to model global energy effects, at least not in the current stage.

Lastly, the process of directed ablation is also related to digital sculpting applications [1][8][15][16]. For example, Baerentzen [1] has suggested both spray paint and CSG tools to add or subtract (fuzzy) layers of volumetric objects. However, in this model, the volumes are made of homogeneous clay, while in our ablation framework the object can be heterogeneous.

3. Ablation Theory

According to standard physics literature (see for example [21]), melting is not an event that occurs suddenly. Rather, it is more of a gradual molecular process. The melting starts at the surface and then propagates across the material. When heat energy is incident on a material, the material's molecules absorb this energy, and this absorption continues as long as the heat energy is present. As the energy of the molecules gradually increases, eventually the energy exceeds the material's latent heat energy. This causes the molecules to undergo a transformation of state. The material then either melts or evaporates, depending on its behavioral properties. In summary, the process of ablation can be described by the following sequence of events:

1) A localized and directed heat source acts upon the material.
2) The heat energy propagates through the material until it is completely dissipated. All dissipated energy is absorbed by the material.
3) When the material has sufficient (absorbed) energy, i.e., more than its latent heat energy, it changes its state. It then either melts or directly evaporates.

Thus, all object portions that receive energy undergo change, whereas other object portions that are not exposed to heat will not be affected. We may model the heating process as a heat source that emits a set of rays. These rays hit the material, deposit energy into it, and if they have energy left, penetrate the material further. The penetration stops once the energy of the rays is totally consumed by the material.

To explain this process more formally, let us consider a single ray with an initial energy E_{ray}. This is illustrated in Fig. 2. When this ray falls onto the material surface (the top layer), then the energy is distributed to set of K molecules within a close neighborhood of the ray penetration site. The energy $E_{rec,i}$ received by one of these molecules $i \in K$ is then:

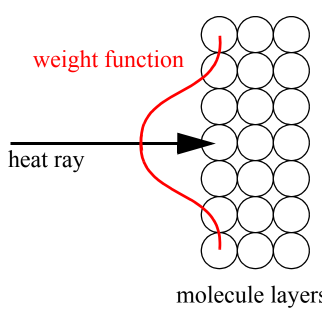

Figure 2: *Heat ray penetrating a few layers of molecules. The weight function controls the distribution of the dissipated energy.*

$$E_{rec,\,i} = w_i \cdot E_{ray} \qquad \sum_{i=1}^{N} w_i = 1 \tag{1}$$

where w_i is a weight factor, determined by the distance of i from the ray's penetration site. We use a trilinear function at the moment, but a Gaussian is also a viable option. The energy $E_{abs,i}$ absorbed by i is given by:

$$E_{abs,\,i} = E_{rec,\,i} \cdot \alpha_{abs,\,i} \tag{2}$$

where $\alpha_{abs,i}$ is the energy absorbtion coefficient of the material to which molecule i belongs. This increases the energy E_i stored in i:

$$E_i = E_i + E_{abs,\,i} \tag{3}$$

All K molecules in the top layer are updated similarly, and the remaining energy of the ray is then given by the law of conservation of energy:

$$E_{ray} = E_{ray} - \sum_{i=1}^{K} E_{abs,\,i} \tag{4}$$

If E_{ray}> 0 then the ray moves into the next material layer, along the same direction vector. This energy dissipation repeats until E_{ray}=0, which means that all of the ray's energy has been consumed by the material. All rays from the source undergo this process.

When a molecule i reaches its latent energy threshold E_{thresh} at which it changes its state, it can no longer absorb energy. In that case, if $E_{rec,i} > (E_{thresh} - E_i)$, then:

$$E_{abs,\,i} = E_{thresh} - E_i \tag{5}$$

and the excess energy $E_{rec,i}$ - $E_{abs,i}$ remains with the ray and is carried to the next material layer. Equations (2) and (5) combine to:

$$E_{abs,\,i} = min(E_{rec,\,i} \cdot \alpha_{abs,\,i}, E_{thresh} - E_i) \tag{6}$$

Note, that in this model the deposition of energy is not binary in nature. There is a gradual increase in the energy of the molecules until they reach their melting or evaporating threshold. The energy function is smooth at the boundaries throughout the ablation process, as long as no surface resistant to ablation is encountered. We shall discuss later how this adverse case could be treated in our implementation.

4. Implementation

The implementation consists of two parts. One part deals with the distribution of the energy, while the other part deals with the rendering of the volume, taking the deposited energy into account. These two processes can occur simultaneously. We use two volumes: The volume V_O holds the original volume data and the volume V_E holds the energy volume data. The original voxels are never removed. Only their energy level indicates if they have evaporated or not. Hence, V_O does not undergo any change, whereas V_E is dynamic and gets updated whenever a ray deposits energy into the volume.

In addition to the usual transfer functions that map interpolated raw densities to color, opacity, and other visual rendering attributes, we require two additional transfer function controls. One is for the energy threshold E_{thresh} and the other is for the energy absorbing coefficient α_{abs}, as discussed in Section 3. These two transfer functions are usually set up, or modified, before a simulation begins (but this is not a requirement). They give users the ability to change a volumetric object's response to incoming energy to whatever desired. The object becomes "magic clay" in that respect.

4.1. Process 1: Depositing the Energy

In practice, we do not add energy to a volume with base-line energies at object-occupied voxels. Rather, we subtract energies from base-line energies, which creates an energy profile that falls off towards the object boundaries. In this way, since we perform the gradient estimation in the energy volume, we obtain gradients that point away from the object.

The algorithm for depositing the energy of the heat rays into the volume proceeds as follows:

1) Set the energy volume V_E to V_O. Alternatively, set V_E to *Baseline(V_O)*, where *Baseline( )* is a function that translates the voxel values of V_O to baseline energies E_i. This can be facilitated by ways of an additional transfer function.
2) From the heat source, cast a set of rays towards the volume. In the simplest case, we use a pencil of parallel rays, where each of these rays has constant energy E_{ray}. More advanced scenarios use a diverging cone of rays, with a Gaussian energy profile across the ray front.
3) For each ray, penetrate the volume and distribute the ray energy according to equations (1) through (5). Use the absorption and energy threshold transfer functions to specify the interaction with of the ray with the material. The distribution of the energy within a voxel neighborhood is done using Gaussian or trilinear weighting (see Fig. 3). Stop the ray once E_{ray} becomes 0.
4) Once all rays have been processed, render the volume and/or jump to 2 for the next ablation iteration.

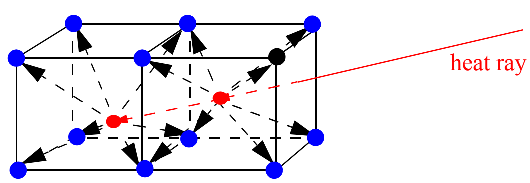

Figure 3: *Spreading the ray energy into a local voxel neigborhood using trilinear weighting.*

4.2. Process 2: Rendering

For rendering, we cast rays in lock step across V_E and V_O. We use trilinear interpolation to interpolate the densities in both volumes. Our ablative process does not change appearance, such as color and shading coefficients. It only changes the opacity of low-energy, evaporated voxels. Thus, the densities from V_E are used to determine transparency (opacity), while the densities from V_O are used to determine color. The gradients (for shading) are also computed from V_E since V_O would not have proper gradients along the current ablation front. We use central differencing to compute the gradients. See Fig. 4 for a block diagram of the volume usage.

We apply the standard RGBA transfer functions in conjunction with post-classified shading. The only difference is that RGB is indexed with densities interpolated from V_O, while A is indexed with densities coming from V_E.

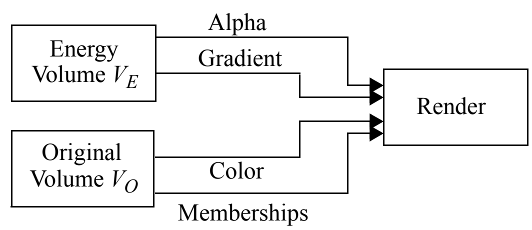

Figure 4: *Volume usage for rendering.*

5. Results

We selected a few standard datasets to illustrate the effectiveness of our technique. For all examples shown the ray grid has about the same resolution than the ablated volume. A simple first application is the evaporation of a chair (size 64^3), as shown in Fig. 1. In Fig. 5, we demonstrate the importance of using a smooth energy volume for rendering. Fig. 5a shows the result of stripping voxels from the original volume without reducing the energy gradually, while Fig. 5b shows the same sequence when a gradual energy reduction is applied as the material is continuously heated. We observe that the former method yields a much grainier surface than the latter technique.

The next example demonstrates the need for decoupling the density mapping of transparency and appearance. Consider Fig. 9a (colorplate) which shows a multi-layered sphere (size 64^3), subjected to a heat source directed at its center. Note that over time, all layers are exposed and then evaporated in front to back order. The transfer function that assigns specific hues and opacities to the various (radial) layers as a function of layer density is shown in Fig. 6a, while Fig. 6b illustrates the (smooth) radial density profile in the evaporation region as the hole grows progressively deeper. If we evaporated the original volume itself, we would need to map

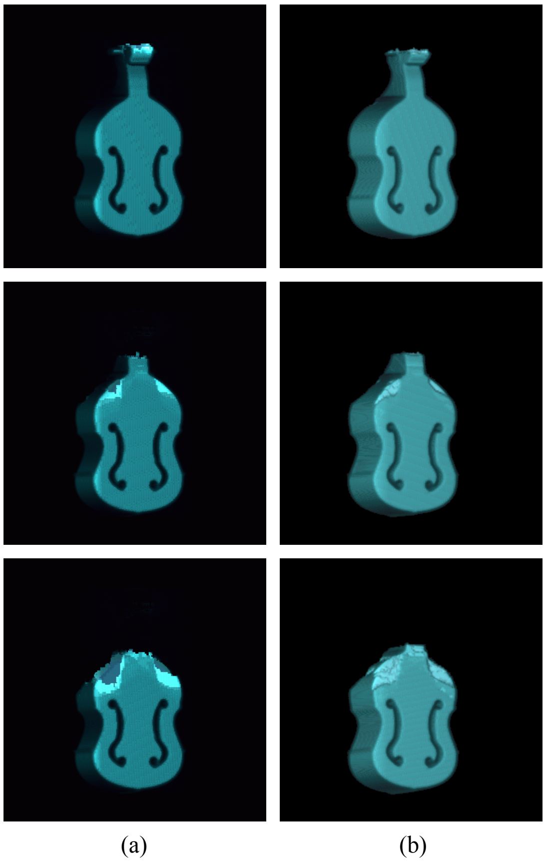

Figure 5: *(a) Ablation with binary voxel peeling from V_O; (b) ablation using a smooth energy volume V_E.*

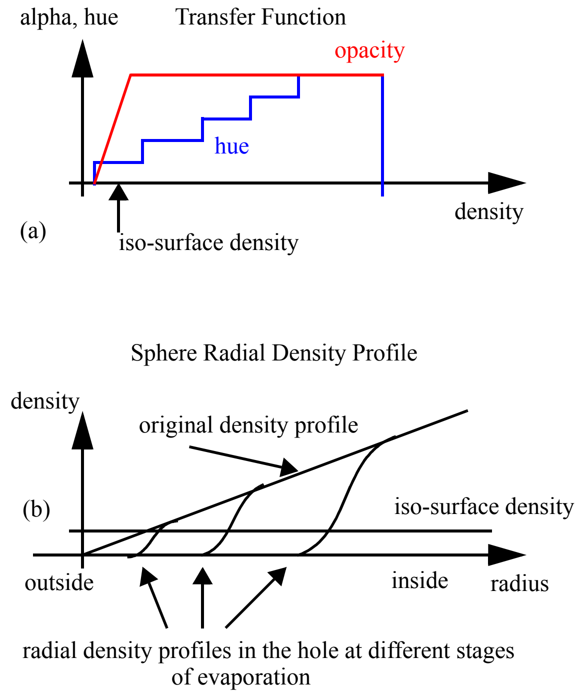

Figure 6: *Smooth sphere dataset: (a) The transfer function that assigns different hues to different radial sphere layers, but keeps the iso-surface at the outermost layer; (b) The shrinking surface and the smooth density profile in the developing hole. Since the iso-surface density does not change, and the density profile always contains the iso-surface density, the color will always be that of the original iso-surface layer. If we, however, only use the hole density profile to map opacities, but use the original density profile to map hue, we will assign the correct hue for the exposed layer.*

these densities to both color and opacity. Note that we need to keep the iso-threshold constant since we would like to preserve the boundaries of the unheated volume portions. Thus, since the iso-threshold does not change, the density-based assignment of appearance parameters, such as color, will not change on the ablated surface as well. This is due to the (required) smooth density profile at the ablation site (as demonstrated in the previous paragraph), which contains the iso-surface density as well. Thus, although we would see a hole develop, we would never see the colors change as different layers are exposed. Instead, we would always see the color of the original iso-surface layer. On the other hand, if we leave the original volume intact (for color mapping) and only modify a separate energy volume (for transparency mapping), then the proper visual results are obtained, as shown in Fig. 9a.

Fig. 9b shows a hard inner cube being revealed by ablation of a soft outer cube. In this case, the absorption transfer function for the density of the inner cube was set to a value close to zero, which means that its material has a very low energy efficiency.

Fig. 7a and b demonstrate the surface appearance of hard material (low absorption) to that of a soft material (high absorption with moderate latent energy threshold). In the latter case, the rays are allowed to penetrate a few layers into the surface before they run out of energy. This has a low-passing effect on the resulting ablation surface, and indeed we observe that the surface and hole edges in Fig. 7a are considerably smoother than in Fig. 7b.

Fig. 7c shows a sequence of images rendered while melting a cello (size 67^2 x 127) with a heat source mounted on top of the scene. Fig. 8a and Fig. 8b show the difference between ablations obtained using an energy beam with a uniform beam profile and one that has a Gaussian beam profile. Finally, Fig. 9c (colorplate) shows a foot (size 256^3) that has a hole burned into it, revealing the bone inside, and Fig. 9d (colorplate) shows a CT head (size 256^3 x 225) where a large heat source strips away the skin and flesh, leaving the “hard” skull exposed.

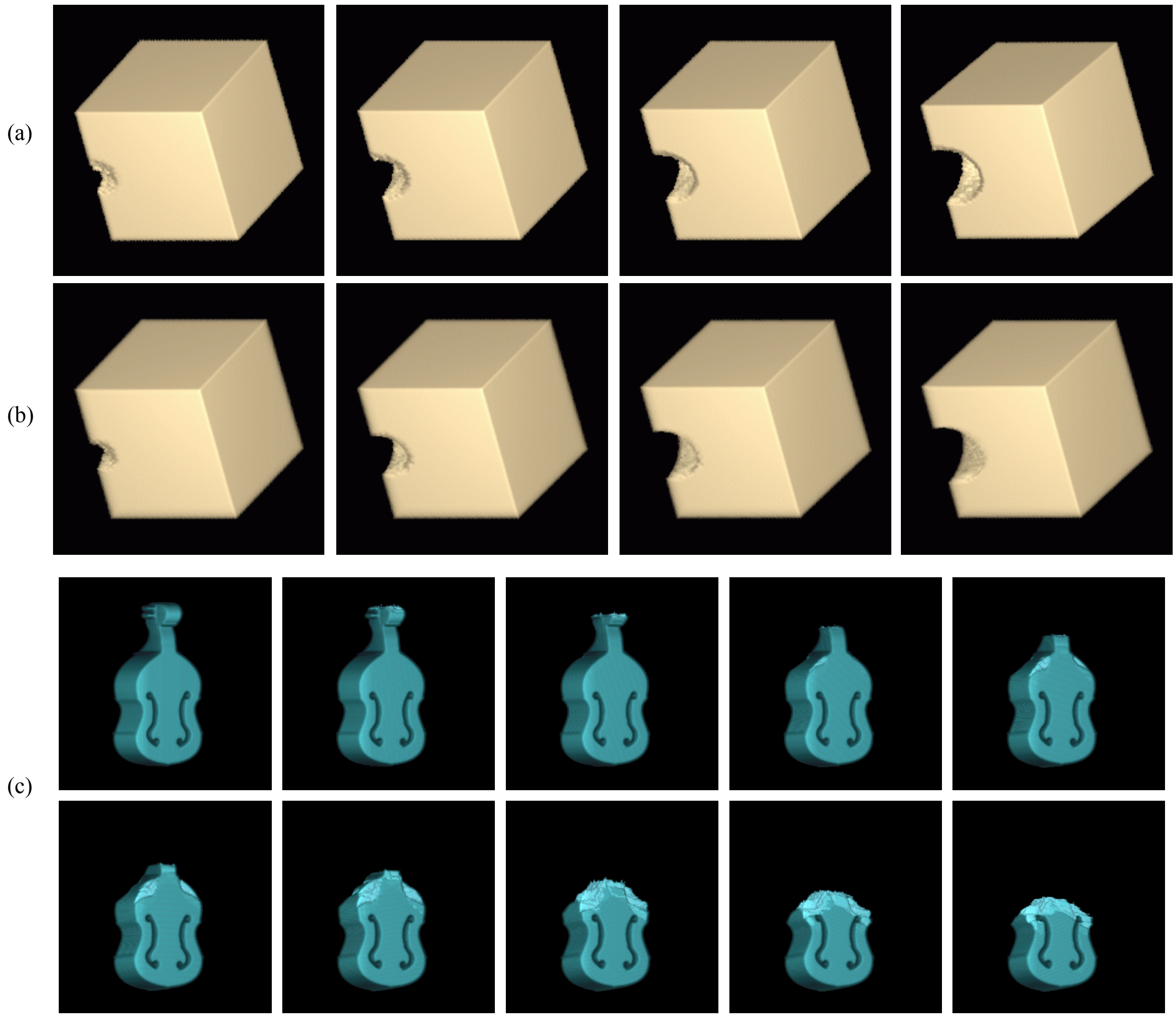

Figure 7: *Ablation experiments: (a) hard material; (b) soft material; (c) a cello slowly evaporates under heat impinging from the top.*

Currently our entire framework is entirely software-based. The run-time of the ablation process is similar to that of the rendering process. Both can use early ray termination to reduce the rendering effort. If the location of the energy source is not changed, then one can use a depth map to start the rays close to the object surface. If the viewpoint is not changed either (that is, the user simply watches the object evaporating), then one can also use a depth map for the rendering. In that case, the system operates and renders at framerates > 5 frames/s. Else, the system requires a few seconds to render a frame.

6. Conclusions and Future Work

We have shown a physically-based, volumetric technique for modeling and rendering the ablation and evaporation of material with a directed energy source. Our approach gives users full control over the absorption behavior of the material as well as of the threshold for change of state by ways of an intuitive transfer function interface. We use a raycasting approach [13] to deposit energy into the volume, but a point-based splatting approach [25] would also be possible. Our method attempts to simulate the physical process of melting and heat propagation. It is therefore more physically-based than current phenomenological methods that generate realistic results, but via heuristic methods. However, we currently constrain ourselves to localized ablation under a directed energy source, while the existing phenomenological methods, developed for thawing and weathering, look at global effects. But we could extend our framework into the global arena as well, perhaps by using an enclosing sphere as the energy source.

(a)

(b)

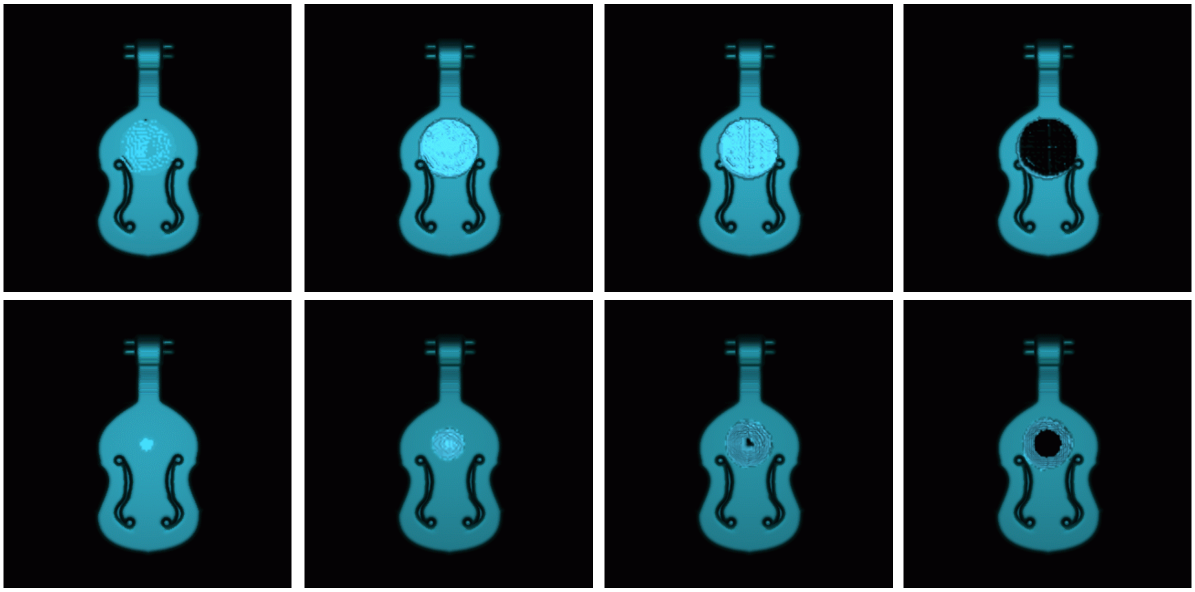

Figure 8: *Ablation experiments: a) using a uniform heat source profile; (b) using a Gaussian heat source profile.*

There are several areas in which further research should be conducted. First, although the ablated surfaces are reasonably smooth, they still appear somewhat grainy and blocky, even with low absorption and high ray penetration depths. To address this issue, we would like to investigate the effect of smoother filters, such as wider Gaussians, as well as non-linear energy decay and absorption in rays and material, respectively, in the ablation phase. Adaptively sampled distance fields (ADFs) [6][16] and detail-adaptive octrees [1] should also be considered. Another problem with our present approach is that once a hard surface with low absorption is encountered (after stripping surrounding soft layers), the energy profile tends to become a step edge. This leads to poor gradients and a binarized look of the hard surface. One solution to this problem is to use the smooth gradients of the original volume instead. The question is now how a binary surface can be detected. A binary surface is likely in regions of low absorption or high energy thresholds. Therefore we could simply use an absorption-weighted gradient vector mix, using gradients from the original and the energy volume.

Two other shortcomings of the present implementation are that the material just evaporates and the energy source is invisible. We would like to add mechanisms, possibly based on celluar automata [7][22][23][24], that simulate the delivery and impact of incoming energy packets as well as models the dissipation of the ablated material under the constraints of gravity and kinetic energy delivered by the beam. There are also different state changes that can be simulated. Currently, we only model the state of evaporation, that is, the material is transformed into an invisible gas. In melting, the material turns into a soft, flexible substance that deforms and flows under the force of gravity. Momental forces are also at work that lead to bending around weakened parts of the object.

To make the system more interactive, we are currently working on a hardware implementation, using a GeForce4 graphics accelerator. It will enable users to interactively move the energy source to ablate different parts of the object and simultaneously render the results at interactive speeds [5][17]. We are also planning to use the same hardware to implement the cellular automata, for the purposes mentioned above, using the framework described in [10][22][23][24].

Finally, useful extensions of the approach include the incorporation of state changes that do not remove material but perhaps only change their visual attributes, such as color (to simulate burned or hot material), or their absorption behavior (to simulate stiffening and hardening under heat). Finally, our framework could also be potentially useful for surgical simulator software that wants to incorporate tools to practice radiofrequency ablation for treating cancer, opening clogged arteries, or other surgical procedures.

Acknowledgements

This work was partially supported by NSF CAREER grant ACI-0093157. We would like thank the anonymous paper reviewers for their helpful comments.

References

[1] A. Baerentzen, “Octree-based sculpting,” *Late Breaking Hot Topics IEEE Visualization’98*, pp. 9-12, 1998.

[2] M. Carlson, P. Mucha, R. Brooks Van Horn III, and G. Turk, "Melting and Flowing," *ACM SIGGRAPH Symposium on Computer Animation*, 2002.

[3] C.-F. Chou, A. Jin, S. Hui, C. Huang, J. Ho, "Multiple-step melting in two-dimensional hexatic liquid-crystal films ," *Science,* vol. 280, no. 5368, May 19, 1998.

[4] J. Dorsey, A. Edelman, H. Jensen, J. Legakis, and H. Pedersen, "Modeling and rendering of weathered stone, *Proc. SIGGRAPH 1999,* pp. 225-234.

[5] K. Engel, M. Kraus, and T. Ertl, "High-quality pre-integrated volume rendering using hardware-accelerated pixel shading," *Proc. SIGGRAPH/ Eurographics Hardware Workshop 2001*, pp. 9-16.

[6] S. Frisken, R. Perry, A. Rockwood, Jones, "Adaptively sampled distance fields: A general representation of shape for computer graphics", *Proc. SIGGRAPH 2000,* pp. 249-254.

[7] I. Fujishiro and E. Aoki, "Volume graphics modeling of ice thawing," *Proc. Volume Graphics Workshop 2001*, pp. 69-81.

[8] T. Galyean and J. Hughes, "Sculpting: An interactive volumetric modeling technique," *SIGGRAPH 1991*, pp. 267-274.

[9] S. Gibson, "Using distance maps for accurate surface reconstruction in sampled volumes," *1998 Symposium on Volume Visualization,* pp. 23-162.

[10] M. Harris, G. Coombe, T. Scheuermann, and A. Lastra, "Physically-based visual simulation on graphics hardware," *Proc. 2002 SIGGRAPH / Eurographics Workshop on Graphics Hardware*.

[11] S. King, R. Crawfis, and W. Reid, "Fast volume rendering and animation of amorphous phenomena," *Proc. Volume Graphics Workshop 1999*, pp. 229-242.

[12] Li and J. M. Moshell, "Modeling soil: Realtime dynamic models for soil slippage and manipulation," *SIGGRAPH 93*, pp. 361-368.

[13] M. Levoy, "Display of surfaces from volume data", *IEEE Computer Graphics. & Applications*, vol. 8, no. 5, pp. 29-37, 1988.

[14] N. Ozawa and I. Fujishiro, "A morphological approach to volume synthesis of weathered stone," *Proc. Volume Graphics Workshop 1999*, pp. 367-378.

[15] K.-L. Perng, W.-T. Wang, M. Flanagan, and M. Ouhy-oung. "A real-time 3D virtual sculpting tool based on modified Marching Cubes," *ICAT (International Conference on Artificial Reality and Tele-Existence)* 2001.

[16] R. Perry and S. Frisken, "Kizamu: A system for sculpting digital characters", *Proc. SIGGRAPH 2001,* pp. 47-56, 2001.

[17] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl, "Interactive volume rendering on standard PC graphics hardware using multi-textures and multi-stage-rasterization" *Proc. SIGGRAPH/Eurographics Workshop on Graphics Hardware 2000*, pp. 109-118.

[18] M. Sramek, "Object voxelization by filtering," *1998 Symposium on Volume Visualization,* pp. 111-118.

[19] R. Sumner, J. O'Brien, and J. Hodgins, "Animating sand, mud, and snow," *Computer Graphics Forum,* vol. 18, no. 1, pp. 3-15, 1999.

[20] J. Tannehill, D. Anderson, and R. Pletcher, *Computational Fluid Dynamics and Heat Transfer,* Taylor & Francis; second edition.

[21] J. Walker, *Physics: An Introduction*, Prentice-Hall, 2001.

[22] X. Wei, W. Li, K. Mueller, and A. Kaufman, "Simulating Fire with Texture Splats," *IEEE Visualization 2002,* pp. 227-237.

[23] X. Wei, W. Li, and A. Kaufman, "Melting and flowing of highly viscous volumes in virtual environments," (poster) *IEEE Virtual Reality 2003*.

[24] X. Wei, W. Li, and A. Kaufman, "Melting and flowing of viscous volumes," *Conferemce on Computer Animation and Social Agents 2003*.

[25] L. Westover, "Footprint evaluation for volume rendering", *Proc. SIGGRAPH 1990*, pp. 367-376.

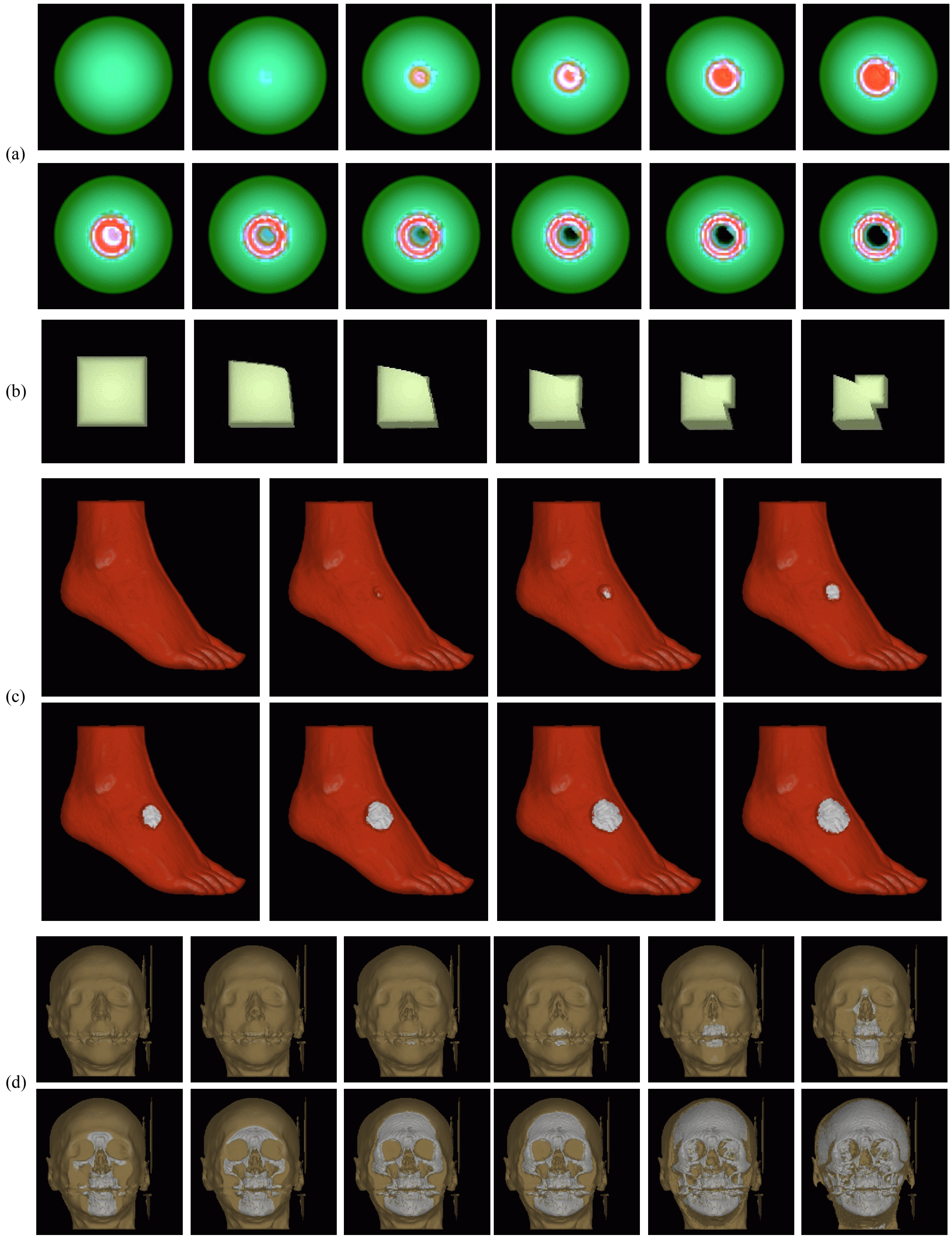

Figure 9: *More ablation experiments: (a) a smooth sphere with many layers; (b) a hard cube inside a soft cube(c); burning a hole into the flesh of a foot; (d) melting away the flesh of the CT head.*