An Interactive Editing System for Visual Appearances of Fire and Explosions

Y. Dobashi[1,2,3], Y. Shibukawa[1], M. Tada[1], S. Sato[3], K. Iwasaki[2,3,4] and T. Yamamoto[1]

[1]Hokkaido University, Sapporo Japan,   [2]JST CREST, Tokyo Japan,
[3]UEI Research, Tokyo Japan,   [4]Wakayama University, Wakayama Japan

Abstract

Synthetic volumetric fire and explosions are important visual effects used in many applications such as computer games and movies. For these applications, artists are often requested to achieve the desired visual appearance by adjusting some parameters for rendering them. However, this is extremely difficult and tedious, due to the complexity of these phenomena and the expensive computational cost for the rendering. This paper presents an interactive system that assists this adjustment process. Our method allows the user to interactively change the ratio of smoke and flame regions, the emissivity of the flame, and the optical thicknesses of the smoke and flame separately. The image is updated in real-time while the user modifies these parameters, taking into account the multiple scattering of light. We demonstrate the usefulness of our method by applying our method to editing of the visual appearances of fire and explosions

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Volume Rendering

1. INTRODUCTION

Volumetric objects, such as smoke and fire, are important elements that enhance the realism of a synthetic scene. Among such volumetric objects, fire and explosions are often the most important visual effects in movies and computer games. Artists are often requested to produce a particular visual appearance in order to achieve a specific visual effect. One must adjust the parameters used in both the simulation and rendering processes to create the desired visual appearance of fire and explosions. Here, we focus on the rendering part. Fire and explosions consist of emission and absorption regions, corresponding to flame and smoke, respectively. The visual appearances of fire and explosions therefore change according to the ratio of these two regions and their colors and opacities. Adjusting these factors is a very time-consuming and tedious task.

We propose a system that assists the user in designing the visual appearances of fire and explosions. Our system allows the user to interactively modify the ratio of smoke and flame regions, their opacities, and their colors. Images are updated in real-time, taking into account multiple scattering. A set of basis images are precomputed to realize interactive feed-
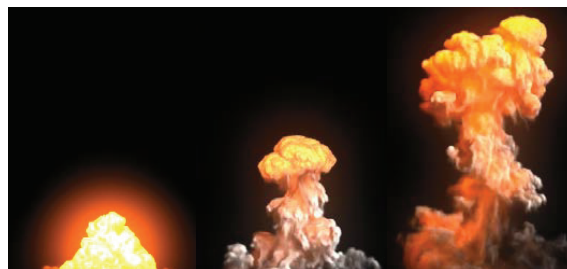

Figure 1: *A visual appearance of an animated explosion edited by our system.*

back. Fig. 1 shows examples of the editing of the visual appearance of an explosion by using our system.

2. RELATED WORK

The visual appearances of volumetric fire and explosions can be designed by using a transfer function that assigns a color and an opacity for each voxel. In the field of scientific visualization, many methods have been proposed for

constructing suitable transfer functions [Kin02]. Recently, many researchers have focused on the interactive design of transfer functions by directly painting rendered images [WQ07, CM11, GY13]. However, these methods basically assign the desired colors/opacities directly to each voxel, and therefore the colors of the rendered image do not match the desired colors. In addition, these methods do not take into account the physical process of light scattering.

Several methods have been proposed for designing volumetric effects caused by subsurface scattering [STPP09, HR13]. However, these methods are limited to relatively opaque objects and are not suitable for fire and explosions. Klehm et al. [KISE13] proposed a method for computing the voxel properties so that the rendered image of the volume becomes the same as a user-specified image. Although this method can design the visual appearances of the volumes such as smoke, multiple scattering is not taken into account.

3. SYSTEM DESIGN

We designed our system by taking into account the following requirements. We assume that the input volume data stores a density and a temperature at each voxel.

1. The system must take into account important factors that determine the visual appearance of fire and explosion.
2. The number of manipulatable parameters should be as small as possible.
3. The system must provide interactive feedback with the user.

As for the first requirement, there are three important factors that affect the visual appearance of fire and explosions: the ratio of the smoke and flame regions, their colors, and their opacities. Multiple scattering of light is also an important factor. We need to take into account these factors in designing our system.

For the smoke and flame regions, we follow the common approach: using a threshold temperature to classify each voxel into either smoke or flame. For the color and opacity, a traditional system uses a color map and an opacity map. These maps often depend on both the temperature and the density. However, this approach violates the second requirement; the parameters are too many. We thus propose the following model. The opacity is controlled by using only two extinction coefficients corresponding to smoke and flame regions, respectively. The colors are controlled by using a color map that converts the temperature to the color and intensity of the emissivity of the flame. We found this simplified model can produce various visual appearances, as demonstrated in Section 6.

Next, to address the third requirement while taking into account multiple scattering of light, we employ a precomputation-based approach. Basically, a set of images are precomputed with different sampled values of the parameters; we call the precomputed images basis images. In order to reduce the precomputation time, we develop an efficient method using the photon mapping technique.

4. PARAMETER REPRESENTATION

Let us denote the temperature and the density stored in the volume by T and ρ, respectively. In our system, the temperature T is sampled into n bins at a regular interval ΔT. Let us denote the center of the bin by $T_i = (i+0.5)\Delta T$ ($i = 0, 1, \cdots, n-1$). The number of bins is specified by the user.

The ratio of the smoke and flame regions and their extinction coefficients are controlled in the following way. The attenuation of light between a pair of points x and y is expressed by:

$$\tau(x, y, T_c) = \exp\left(-\int_x^y \kappa(T(s), T_c)\rho(s)ds\right), \quad (1)$$

where κ is the extinction coefficient computed from temperature $T(s)$ at point s between x and y. T_c is the threshold for the classification of a point into smoke or flame. We represent the extinction coefficient by:

$$\kappa(T(s), T_c) = \frac{\kappa_f - \kappa_s}{2}f(T(s) - T_c) + \frac{\kappa_f + \kappa_s}{2}, \quad (2)$$

where κ_f and κ_s are the extinction coefficients for flame and smoke, respectively. f is a sigmoid function modified to be fit into a range from -1 to 1, that is,

$$f(r) = 2/(1 + \exp(-r/\sigma_f)) - 1, \quad (3)$$

where σ_f determines the slope of the function around $r = 0$. We use ΔT for σ_f. The threshold temperature T_c is sampled at T_i ($i = 0, 1, \cdots, n-1$) and the extinction coefficient is computed for each sampled temperature. The attenuation ratio τ is linearly interpolated between them. By adjusting T_c, the user can control the ratio of the smoke and flame regions. κ_s and κ_f are independently controlled to adjust the optical thicknesses of the smoke and flame respectively. The use of the sigmoid function allows us to smoothly interpolate the extinction coefficient between smoke and flame. It also results in the smooth transition of the ratio of smoke and flame regions.

The emissivity is controlled in the following way. We assume that the emissivity of each voxel is a function of temperature T and wavelength λ. λ is sampled at the wavelengths corresponding to the RGB color channels. The emissivity is specified for each sampled temperature T_i. We use radial basis functions (RBFs) for the smooth interpolation; that is, the emissivity function L_e is given by:

$$L_e(T, \lambda) = \sum_{i=0}^{n-1} c_i(\lambda)\phi(T - T_i), \quad (4)$$

where c_i is the coefficient for i-th RBF and is determined from the user specified emissivity at temperature T_i. The number of RBFs is the same as the number of temperature samples. We use a Gaussian RBF: $\phi(r) = \exp(-r^2/\sigma_\phi)$,
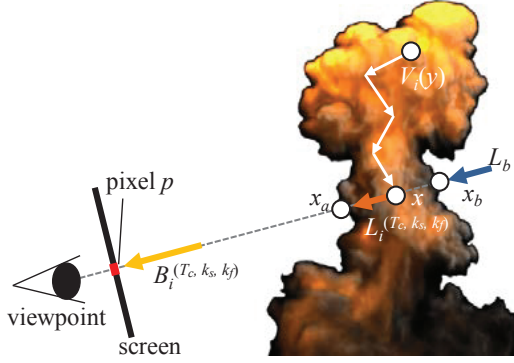
Figure 2: *Calculation of basis images.*

where σ_ϕ controls the smoothness of the emissivity function. We use $\Delta T/3$ for σ_ϕ.

5. EFFICIENT RENDERING USING BASIS IMAGES

This section describes the method for real-time rendering of fire and explosion by precomputing the basis images. We first explain the definition of the basis images and then an efficient method for computing the basis images using the photon mapping technique. In the following, let us omit the wavelength λ to simplify the notations. Our method applies to each of the color channels independently.

5.1. Definition of Basis Images

Fig. 2 illustrates the idea of the basis images $B_i^{(T_c,\kappa_s,\kappa_f)}(p)$ that stores intensity of light reaching the viewpoint through pixel p taking into account multiple scattering of light, when the emissivity and transmittance of the volume are determined by i-th RBF $\phi(T - T_i)$, and the parameters $(T_c, \kappa_s, \kappa_t)$. In this section, we derive the formulation of the basis image. For simplicity, let us assume that the light source illuminating the volume is the emission of the volume itself. We will explain later how the light sources outside the volume are incorporated. We assume isotropic scattering.

Let us define an operator $\mathcal{M}$ that computes the radiance at voxel x taking into account multiple scattering, given an emission volume V_e storing the emissivity L_e at each voxel. $\mathcal{M}$ is a function of T_c, κ_s, and κ_f. Let us denote this dependence by using superscripts like $\mathcal{M}^{(T_c,\kappa_s,\kappa_f)}$. We express the radiance at x by the following equation.

$$L(x) = \mathcal{M}^{(T_c,\kappa_s,\kappa_f)}(V_e). \qquad (5)$$

$\mathcal{M}$ is a linear operator with respect to the emissivity L_e. Since L_e is expressed by a linear combination of RBFs as shown in Eq. 4, the emission volume V_e is also represented by a linear combination of basis emission volumes V_i storing a value of i-th RBF $\phi(T - T_i)$ at each voxel, that is,

$V_e = \sum_i^{n-1} c_i V_i$. Then, Eq. 5 is rewritten as:

$$L(x) = \sum_{i=0}^{n-1} c_i L_i^{(T_c,\kappa_s,\kappa_f)}(x), \qquad (6)$$

where $L_i^{(T_c,\kappa_s,\kappa_f)}$ is the radiance due to the basis emission volume V_i and is given by:

$$L_i^{(T_c,\kappa_s,\kappa_f)}(x) = \mathcal{M}^{(T_c,\kappa_s,\kappa_f)}(V_i). \qquad (7)$$

The final image is obtained by accumulating $L(x)$ along the viewing ray. By using Eqs. 6 and 7, the radiance reaching the viewpoint through pixel p is expressed by:

$$
\begin{aligned}
L_v(p) &= \int_{x_a}^{x_b} \rho(x)L(x)\tau(x_a,x)dx + L_b(p)\tau(x_a,x_b) \\
&= \sum_{i=0}^{n-1} c_i B_i^{(T_c,\kappa_s,\kappa_f)}(p) + L_b(p)\tau(x_a,x_b), \qquad (8)
\end{aligned}
$$

where,

$$B_i^{(T_c,\kappa_s,\kappa_f)}(p) = \int_{x_a}^{x_b} \rho(x)L_i^{(T_c,\kappa_s,\kappa_f)}(x)\tau(x_a,x)dx, \quad (9)$$

and x_a and x_b are the intersections of the viewing ray and the volume, x is a point between x_a and x_b (see Fig. 2). L_b is the radiance of light reaching from behind the volume. Eq. 9 is the definition of the basis image.

As shown in Eq. 8, an image of the volume is represented by the weighted sum of the basis images. Therefore, by precomputing the basis images, we can obtain the image very efficiently. The illumination due to light sources outside the volume is easily incorporated by preparing an additional set of basis images for the external light source. The number of basis images increases in proportion to the number of the light.

5.2. Efficient Computation Using Photon Mapping

We use the photon mapping technique [JC98] to compute the basis images. The photon mapping firstly computes a photon map by shooting many photons from light sources. For the emissive volume like fire and explosion, light sources are the volume itself. Every voxel with positive temperature is potentially a light source. We choose a voxel with positive temperature by using a uniform random number and generate a photon at the center of the voxel. The direction of the photon is also determined by using a uniform random number.

Let us now consider a photon q generated at point y inside the volume. The initial light energy assigned to this photon is computed by using the emissivity function $L_e(T_q)$, where T_q is the initial temperature of the photon (i.e., $T_q = T(y)$). The path of the photon is constructed by using random numbers that determine the successive locations where the scattering events occur. The photon is absorbed when a random number is smaller than the albedo at the scattering location.

When photon q is absorbed, the light energy of the photon

is expressed by the following equation.

$$L_q^{(T_c, \kappa_s, \kappa_f)} = L_e(T_q)\tau_q^{(T_c, \kappa_s, \kappa_f)} \tag{10}$$

where $\tau_q^{(T_c, \kappa_s, \kappa_f)}$ is the attenuation ratio of the photon along the path and is given by:

$$\tau_q^{(T_c, \kappa_s, \kappa_f)} = \exp(-\int_C \kappa(T(s), T_c)\rho(s)ds), \tag{11}$$

where C is a set of points on the path of the photon. By putting Eq. 2 into the above equation, we obtain the following equation.

$$\tau_q^{(T_c, \kappa_s, \kappa_f)} = \exp(-ao_a(T_c) - bo_b), \tag{12}$$

where,

$$a = (\kappa_f - \kappa_s)/2,$$
$$b = (\kappa_f + \kappa_s)/2,$$
$$o_a(T_c) = \int_C f(T(s) - T_c)\rho(s)ds,$$
$$o_b = \int_C \rho(s)ds.$$

Our method computes the weighted optical depths, $o_a(T_c)$ and o_b, during the photon tracing process and stores these together with the initial temperature T_q in our photon map. Note that, as we described in Section 4, T_c is sampled at $T_i (i = 0, 1, \cdots, n-1)$ and thus o_a is computed for each sampled temperature.

Once the photon map is constructed, our method computes $B_i^{(T_c, \kappa_s, \kappa_f)}(p)$ for different sampled values of $(T_c, \kappa_s, \kappa_f)$. A viewing ray is cast through each pixel and Eq. 9 is evaluated numerically by generating a set of sample points on the ray. For sample point x on the viewing ray, we estimate the radiance $L_i^{(T_c, \kappa_s, \kappa_f)}(x)$ by using the photon map with the traditional density estimation method [JC98]. Since the storage size for the basis images becomes very large (> several gigabytes), we compress them using Singular Value Decomposition (SVD).

Using the precomputed basis images, the images are rendered efficiently according to the parameters T_c, κ_s, κ_f, and c_i. In addition to the image of the volume itself, we render the effect of lens flare at runtime.

6. RESULTS

The examples shown in this section were computed on a desktop PC with an Intel Core i7 4770K (CPU) and NVIDIA GeForce GTX TITAN 6GD5 (GPU). All of the computation was accelerated by using the GPU. In addition to the emission of the volume, we used a single parallel light source as an external light source. Videos demonstrating the editing process and the animation of the fire and the explosion are also included in the supplemental material.

We apply our system to the editing of the appearance of an animated explosion as shown in Fig. 1. We chose three frames from the animated data and separately adjusted the parameters for rendering them. The final animation is created by linearly interpolating the three sets of parameters, as shown in the figure. In this example, the precomputation took 1.6 hours and the size of the precomputed data was 1.4 GB but it is reduced to 165 MB by our compression method using SVD.

7. CONCLUSIONS

We have proposed an interactive system for designing the visual appearance of volumetric fire and explosions. Our system allows the user to adjust the amount of smoke or flame regions. The emissivity of each voxel is also determined by the user. When the user modifies these parameters, our method can render the image in real-time while taking into account multiple scattering. We have demonstrated that the visual appearances of volumetric fire and explosions can be interactively designed by using our method.

Although we have demonstrated that our system can produce various visual appearances, we have not justify our choice of the manipulatable parameters. In the future, we would like to verify if our parameter set covers the whole space of the appearances that can be produced with commonly-used parameters. We are also interested in solving an inverse rendering problem for fire and explosions, that is, to determine all of the parameters inversely from a user-specified image.

References

[CM11] CORREA C. D., MA K.-L.: Visibility histograms and visibility-driven transfer functions. *IEEE Transaction on Visualization and Computer Graphics 17*, 2 (2011), 192–204.

[GY13] GUO H., YUAN X.: Local wysiwyg volume visualization. In *IEEE Pacific Visualization Symposium* (2013), pp. 65–72.

[HR13] HAŠAN M., RAMAMOORTHI R.: Interactive albedo editing in path-traced volumetric materials. *ACM Trans. on Graphics 32*, 2 (2013), Article No. 11.

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of SIGGRAPH'98* (1998), pp. 311–320.

[Kin02] KINDLEMANN G.: Transfer functions in direct volume rendering: Design, interface, interaction. In *ACM SIGGRAPH 2002 Course Notes* (2002).

[KISE13] KLEHM O., IHRKE I., SEIDEL H.-P., EISEMANN E.: Volume stylizer: Tomography-based volume painting. In *Symposium on Interactive 3D Graphics and Games* (2013), pp. 161–167.

[STPP09] SONG Y., TONG X., PELLACINI F., PEERS P.: Subedit: a representation for editing measured heterogeneous subsurface scattering. *ACM Transactions on Graphics (Proc. SIGGRAPH 2009) 28*, 3 (2009), Article 31.

[WQ07] WU Y., QU H.: Interactive transfer function design based on editing direct volume rendered images. *IEEE Transactions on Visualization and Computer Graphics 13*, 5 (2007), 1027–1040.