PROCEEDINGS, Fire and Evacuation Modeling Technical Conference 2018
Gaithersburg, Maryland, October 1-3, 2018

VISUALIZING SMOKE AND FIRE

Glenn Forney[1]

[1] National Institute of Standards and Technology
Gaithersburg, Maryland, USA
e-mail: glenn.forney@nist.gov

ABSTRACT

This note discusses some of the physics and associated numerical algorithms used by Smokeview to visualize smoke and fire. Realistic visualization methods are important for applications where one wishes to observe qualitative effects of fire and smoke rather than quantitative characteristics such as temperature or velocity. Approximations and simplifications are required to display smoke and fire at interactive frame rates. The primary approximation is to take advantage of the low albedo character of smoke allowing one to either simplify or eliminate scattering terms in the radiative transport equation used to visualize smoke and fire.

INTRODUCTION

This note discusses some of the physics and associated numerical algorithms used by Smokeview [1, 2] to visualize smoke and fire. Smoke color and opacity are visualized using quantitative physics-based methods. Flame color is visualized using an arbitrary user-specified color palette where color is mapped to gas temperature. Smoke opacity is visualized using Beer's law relating smoke density and opacity.

Realistic visualization of fire is important for applications where one wishes to observe qualitative effects of fire and smoke rather than determine quantitative data such as temperature or velocity. This would be the case for a fire fighter using a computer based fire fighting simulator. Realistic visualization methods, however, complement but do not replace other more traditional visualization methods such as 2D contouring or 3D iso-surfacing which are better suited for quantitatively analyzing data.

Complete methods for visualizing smoke and fire data taking into account interactions between light and smoke require the solution of the radiation transport equation (RTE) [3] also called the volume rendering equation in the visualization literature. [4] This equation models how light is affected after interacting with smoke, a participating medium. In particular, Smokeview uses the RTE to account for extinction (absorption plus out-scattering) by the smoke and emission from the fire.

The form of the RTE used by Smokeview to model smoke and fire appearance is identical to that used by the Fire Dynamics Simulator (FDS) to model radiative heat transfer. Smokeview uses an extinction coefficient appropriate for visible light while FDS use one appropriate for infrared wavelengths of light. With the proper extinction coefficient, however, Smokeview can also view smoke at other wavelengths,

simulating a thermal imager, for example. Smokeview solves the RTE assuming a gray gas environment. This is the default solution method for FDS. One other important difference is that Smokeview requires a solution at only one point at a time (any arbitrary point though), the observer's viewpoint, while FDS requires a radiation field, a solution at all points within the solution domain. Approximations are required in order to display smoke and fire at interactive frame rates. The primary approximation is to take advantage of the low albedo character of smoke allowing one to either simplify or eliminate scattering terms in the RTE.

A slice rendering method for solving the RTE, splits the integration path at grid planes within a 3D mesh. A series of partially transparent slices are drawn through the data where each slice is approximately perpendicular to the line of sight. These partially transparent slice planes are then drawn individually and combined by the video hardware to form one image. The spacing of data within one plane and the spacing between planes are parameters that can be specified by the user in order to speed up the visualization.

As the separation distance between slice planes becomes smaller, the computed opacity values are subject to increased round off error due to finite precision arithmetic. In fact, if these planes are sufficiently close, the computed opacities truncate to zero. In this situation other techniques for visualizing smoke such as volume rendering methods are required.

RADIATION TRANSPORT EQUATION

The model used here to visualize smoke is the radiation transport equation (RTE) [3]. This equation uses radiance to represent smoke appearance. Radiance has units of Watts per square meter per unit solid angle $\mathrm{W}/(\mathrm{sr}\cdot\mathrm{m}^2)$. The solid angle accounts for the fact that a light source appears brighter if it emits a given amount of light through a smaller cross-sectional area. Similarly the radiance of a light source does not depend on distance from the observer (unless a participating medium is present to absorb or scatter light) since any increase in distance that would reduce radiance is offset by the light source's reduced cross sectional area. The radiation transport equation discussed in this section models the change in radiance due to these factors.

The radiation transport equation is used to calculate radiance due to one or more light sources within a region possibly containing a participating medium such as smoke [3]. The change in radiance along a ray with direction ω at any one instant and wavelength may be expressed using

$$(\omega\cdot\nabla)\,C(x,\omega) = -\underbrace{\sigma_a(x)C(x,\omega)}_{\text{absorption}} - \underbrace{\sigma_s(x)C(x,\omega)}_{\text{out-scattering}} + \underbrace{\sigma_a(x)C_e(x,\omega)}_{\text{emission}} + \underbrace{\sigma_s(x)\int_{4\pi} p(x,\omega,\omega')C_i(x,\omega')\,\mathrm{d}\omega'}_{\text{in-scattering}} \quad (1)$$

where $C(x,\omega)$ represents the radiance at x along a direction ω. As illustrated in Figure 1, the right hand side of equation (1) is split into four components accounting for absorption, in and out scattering and emission where $\sigma_a(x)$ is the absorption coefficient, $\sigma_s(x)$ is the scattering coefficient, $C_e(x,\omega)$ is the radiance emitted at x along a direction ω and $p(x,\omega,\omega')$ is the fraction of light moving along direction ω' scattered along direction ω. Absorption and out-scattering cause radiance to decrease while emission and in-scattering cause radiance to increase. The radiance terms C, C_e and C_i have units of $\mathrm{W}/(\mathrm{m}^2\cdot\mathrm{sr})$. The coefficients σ_a and σ_s have units of $1/\mathrm{m}$ and specify the time and location

dependent change per unit length to the radiance term to which they are applied.

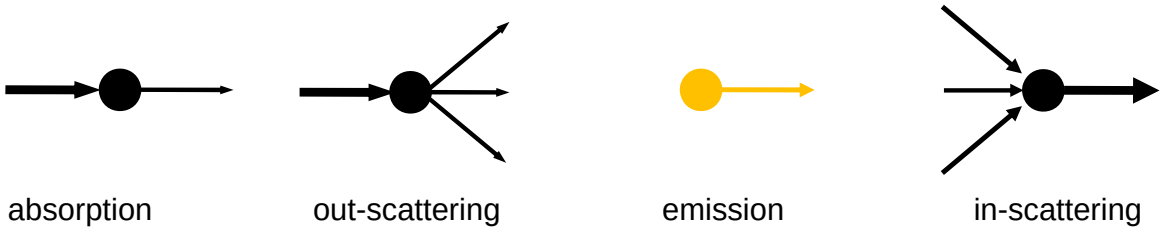


Figure 1: Diagram illustrating components of the radiation transport equation. Absorption and out-scattering terms decrease radiance. Emission and in-scattering terms increase radiance.

Approximating the Radiation Transport Equation

The RTE may be simplified by ignoring the in-scattering term and combining out-scattering and absorption coefficients. The Beer-Lambert law results if the emission term is also neglected.

Equation (1) is then approximated by neglecting the integral term and using $\sigma_t(x) = \sigma_a(x) + \sigma_s(x)$ to obtain

$$\frac{\mathrm{d}C}{\mathrm{d}x}(x) = -\sigma_t(x)C(x) + \sigma_a(x)C_e(x) \tag{2}$$

$$C(x_0) = C_0 \tag{3}$$

This equation has solution

$$C(x_N) = \tau(x_0, x_N)C_0 + \int_{x_0}^{x_N} \tau(x, x_N)\sigma_a(x)C_e(x)\,\mathrm{d}x \tag{4}$$

where $\tau(a, b)$ representing the optical depth between a and b is given by

$$\tau(a, b) = \exp\left(-\int_a^b \sigma_t(s)\,\mathrm{d}s\right) \tag{5}$$

If the emission term is neglected, $\sigma_t(x) = \sigma_t$ is assumed to be constant and L is the path length then this equation simplifies to

$$\frac{C(x_N)}{C_0} = \exp(-\sigma_t L) \tag{6}$$

which is the Beer-Lambert law.

Discretizing the Radiation Transport Equation

The approximate RTE solution given in equation (4) is discretized by converting integrals into a sum. Figure 2 illustrates the terms used to perform these discretizations. The integration path is split into N parts each with length $\Delta x = (x_N - x_0)/N$. The coordinate system is set up so that the initial radiance, C_0, is located at x_0, most distant from the observer and the final radiance, C_N, is located at x_N closest to the observer.

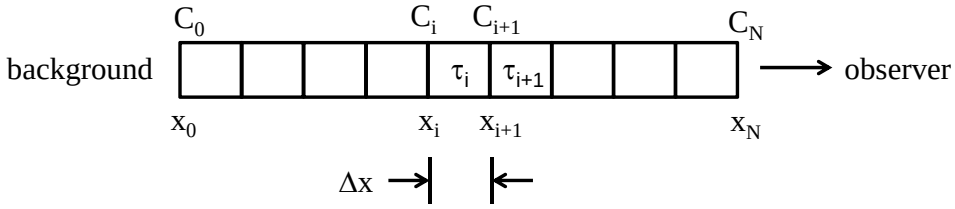


Figure 2: Setup for discretizing the equations used to model radiance within a column of 3D smoke data. The transparency across the interval from x_i to x_{i+1} is τ_i. The transparency across the intervals from x_i to the observer is the product of individual transparencies or $\tau_i \tau_{i+1} \cdots \tau_{N-1}$
.

The optical depth, $\tau(a, b)$, defined in equation (5) is discretized using a sum after defining sample points $s_j = x_0 + j\Delta s$ for $j = 0$ to N with spacing $\Delta s = (x_N - x_0)/N$ to obtain

$$\tau_i^{N-1} = \tau(x_i, x_N) = \exp\left(-\int_{x_i}^{x_N} \sigma_t(s)\, \mathrm{d}s\right) \approx \exp\left(-\sum_{j=i}^{N-1} \sigma_t(s_j)\Delta s\right) \tag{7}$$

$$= \prod_{j=i}^{N-1} \exp\left(-\sigma_t(s_j)\Delta s\right) = \prod_{j=i}^{N-1} \tau_j \tag{8}$$

where $\tau_j = \exp\left(-\sigma_t(s_j)\Delta s\right)$ represents the transparency over one discretization interval. For $i = N - 1$ to 0, the optical depth τ_i^{N-1} may be computed recursively using

$$\tau_i^{N-1} = \tau_{i+1}^{N-1} \tau_i \tag{9}$$

where the recursion is initiated with $\tau_N^{N-1} = 1$. Substituting $1 - \alpha_i^{N-1} = \tau_i^{N-1}$ and $1 - \alpha_i = \tau_i$ into equation (9) gives

$$1 - \alpha_i^{N-1} = (1 - \alpha_{i+1}^{N-1})(1 - \alpha_i) = 1 - \alpha_{i+1}^{N-1} - \alpha_i + \alpha_{i+1}^{N-1}\alpha_i \tag{10}$$

which simplifies to

$$\alpha_i^{N-1} = \alpha_{i+1}^{N-1} + (1 - \alpha_{i+1}^{N-1})\alpha_i \tag{11}$$

Similarly, the radiance given by the RTE solution $C(x_N)$ in equation (4) may be discretized to obtain

$$C_N = \tau_0^{N-1} C_0 + \sum_{i=0}^{N-1} \tau_i^{N-1} \sigma_{a,i} C_{e,i} \Delta x \tag{12}$$

What is seen on the screen is the term C_N. If the expression $\sigma_{a,i} C_{e,i} \Delta x$ in equation (12) is interpreted as the emitted color of the fire or heated gas at a location i and C_0 is interpreted as the color of the light source *behind* the smoke, then equation (12) restated in words gives the observed color as a weighted average of source and emitted colors where each weight is the optical depth from the observer to the corresponding emitted color location. These emitted colors can be determined from a blackbody temperature curve or from a colormap meant to show variations in temperature in terms of color.

IMPLEMENTATION

The algorithm described here for visualizing smoke and fire realistically consists of placing planes within the solution domain and determining opacity and color at various locations within these planes using data generated by a fire model such as FDS. The video card then is used when drawing smokeview to form a solution by combining colors and opacities in the currently drawn plane with those already accumulated in a screen buffer. The original Smokeview algorithm placed planes exactly where data was recorded by FDS, either parallel to the XY, XZ or YZ axes planes or diagonally to two of these planes. The algorithm described below places planes in more general locations. Interpolation is then used within a 3D data set to obtain data values at locations required by the algorithm. This increased flexibility allows one to more easily reduce the amount of data accessed to draw smoke. Planes are placed perpendicular to the line of sight and are equally spaced though not necessarily at the same spacing as used by the underlying FDS grid. This results in faster visualizations with the caveat that reduced data may result in less realistic visualizations. A brief overview of the algorithm is given below.

1. Given spacing parameters parallel and perpendicular to the line of sight, place equally spaced planes through the data, each plane oriented perpendicular to the line of sight. This is illustrated in Figure 3. Plane locations change as the scene is moved. By placing planes only where smoke is located, as illustrated in Figure 4, faster visualizations result. If the graphics processing unit (GPU) is used to visualize smoke then data is passed to the GPU along with vertices of the polygon of the intersected plane. If the central processing unit (CPU) is used to visualize smoke then the polygon needs to be triangulated (the GPU triangulates data within any triangle it draws).

2. Each data plane must be triangulated when smoke is drawn by the CPU. The data plane, represented as a polygon, is divided into a series of equally sized triangles (except for those triangles that border the polygon edge). To do this, a 2D coordinate system is constructed for the planar polygon. As illustrated in Figure 5, let $\mathbf{u}$ be a vector corresponding to the longest polygonal side. Let $\mathbf{v}$ be a vector corresponding to the side clockwise (from the point of view of the observer) from $\mathbf{u}$. Orthogonal unit vectors, $\mathbf{s}$ and $\mathbf{t}$, in the plane of the polygon are formed using

$$\begin{aligned} \mathbf{s} &= \mathbf{u}/||\mathbf{u}||, \\ \mathbf{t} &= (\mathbf{s} \times \hat{\mathbf{v}}) \times \hat{\mathbf{v}} \end{aligned}$$

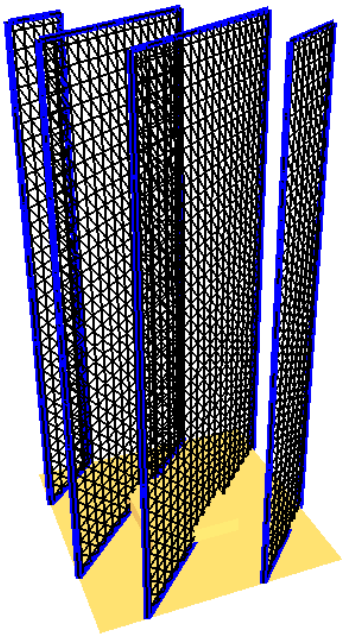


Figure 3: Planes are placed within the entire solution domain so that they are equally spaced and are oriented perpendicular to the line of sight. In this example, the solution domain is rotated and the number of planes is reduced to make them more visible.

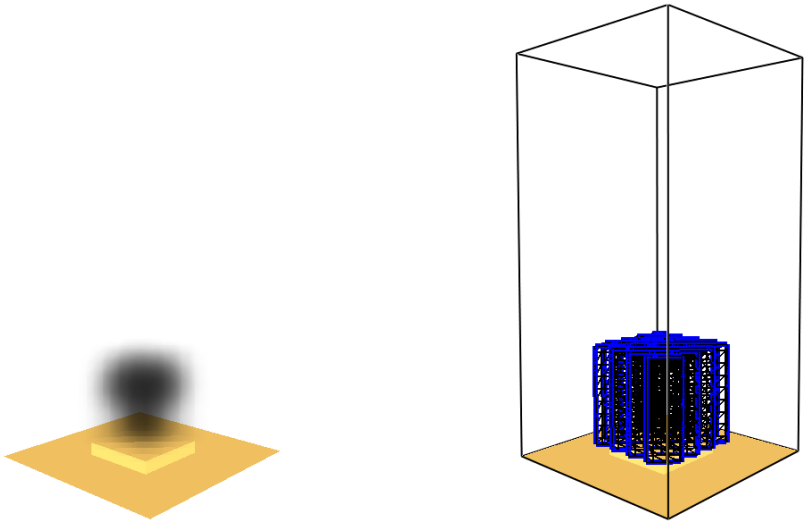
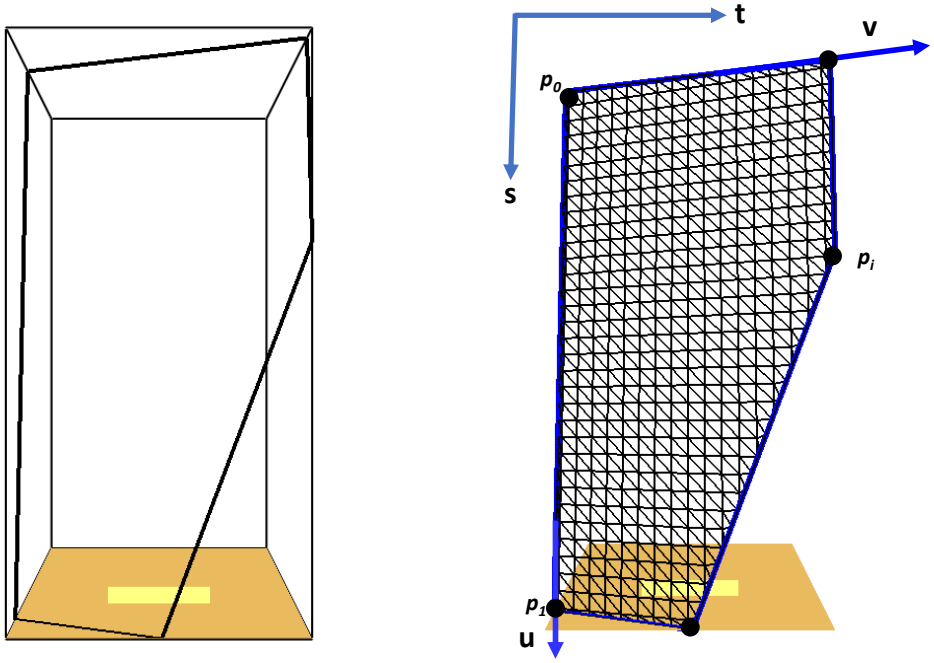


Figure 4: To improve visualization performance, especially at the beginning of the simulation when a fire is typically small, planes are placed only where smoke and fire is located resulting in faster visualizations (since data does not need to be obtained or drawn where it would not be visible).

where $\hat{\mathbf{v}} = \mathbf{v}/||\mathbf{v}||$.



a) intersection of a plane and a solution mesh b) triangulated polygon

Figure 5: Intersection of a plane perpendicular to the line of sight and the solution domain. This results in a polygon which is triangulated using a 2D coordinate system represented by vectors **s** and **t** located in the plane of this polygon. Similar polygons uniformly spaced and perpendicular to the line of site are also generated and triangulated whenever the scene is moved.

3. For a spacing parameter Δ, the polygon is then triangulated in terms of vectors **s** and **t** by first forming vertices $v_{ij} = x_0 + i\mathbf{s} + j\mathbf{t}$ for i and j so that v_{ij} covers the polygon (and the region just next to it) and next forming triangles $(v_{ij}, v_{i+1,j}, v_{i+1,j+1}), (v_{ij}, v_{i+1,j+1}, v_{i,j+1})$ from these vertices. Triangles with all vertices outside the polygon are neglected. If one or two vertices are outside the polygon they are moved to the closest point on the polygon.

The slice orientation is chosen to be the one most perpendicular to the viewer's line of sight. The opacity at each vertex is computed using the distance between adjacent planes and soot density data computed by the fire model. Opacities are adjusted if the distance between planes is different than the spacing parameter used to compute the original opacity. Opacity data is computed and compressed using run length encoding as a preprocessing step and decompressed as each frame is displayed.

Summarizing, a slice rendering algorithm for visualizing smoke consists of splitting the RTE across individual slice planes within a single mesh. Figure 6 shows an example illustrating a various number of slice planes used to visualize smoke and fire. The 3D computational domain is partitioned into a series of 2D slices. The RTE is then solved on each slice. Each slice solution only accounts for conditions between adjacent slices. The individual partially transparent slice solutions are then combined using video hardware to form the final image. Slices become more transparent as the come closer together.

Problems can then occur resulting in poor visualizations because of numerical round off error. In this case, different solution techniques are required such as volume rendering which integrates the entire RTE at once rather than one slice at a time.

Computing Opacity

Consider a ray traveling from the background to the observer through intervening smoke. Light is absorbed or scattered by the smoke as the ray passes through each slice plane. Emission from the flame or hot smoke is implemented by coloring the smoke. Scattering is implemented using the total mass extinction coefficient. Light losses are assumed to be from both absorption and scattering. Obscuration or opacity is computed along each ray one grid plane at a time, using the Beer-Lambert law as follows. The $\alpha = 1 - \tau$ values are pre-computed by FDS using the Beer-Lambert law [3]. The α parameter represents an opacity, 0.0, for completely transparent, and 1.0 for completely opaque and is given by

$$\alpha = 1 - \exp(-\sigma_t \Delta x) \tag{13}$$

for the view direction down the x axis where Δx is the distance between two grid planes and σ_t is the total mass extinction coefficient. The α parameter in equation (13) is used by Smokeview to blend the smoke plane currently being drawn with the background. For a different plane spacing, $\Delta\hat{x}$, the opacity is adjusted using $\hat{\alpha} = 1 - (1 - \alpha)^{\Delta\hat{x}/\Delta x}$.

Computing Color

Smokeview visualizes smoke and fire by drawing a series of triangles in equally spaced parallel planes. Color for these triangles are assigned by mapping temperature or HRRPUV (heat release per unit volume) values to color such as with a color map illustrated in Figure 7. Transparency for these triangles is assigned using soot density, the greater the soot density, the more opaque the triangle.

The example color map in Figure 7 is split into two parts. The left half is used for coloring non-burning regions, the right half is used for coloring burning regions. An HRRPUV cutoff value denoted $\text{hrrpuv}_{\text{cutoff}}$ is used to distinguish these two regions. If an HRRPUV value is below the cutoff, smoke is drawn using colors from the left half of the color map, while if an HRRPUV value is greater than the cutoff, fire is drawn using colors from the right half of the color map. The color map is defined as a table of 256 red, green, blue color triplets. A formula giving a color index for a given HRRPUV value is given by

$$\text{color index} = \begin{cases} 127\dfrac{\text{hrr}}{\text{hrr}_{\text{cutoff}}} & 0 \le \text{hrr} \le \text{hrr}_{\text{cutoff}} \\ 127 + 128\dfrac{\text{hrr}-\text{hrr}_{\text{cutoff}}}{\text{hrr}_{\text{max}}-\text{hrr}_{\text{cutoff}}} & \text{hrr}_{\text{cutoff}} \le \text{hrr} \le \text{hrr}_{\text{max}} \end{cases} \tag{14}$$

A second method for coloring fire is illustrated in Figure 8. Instead of blending colors found along the line of sight it uses the color corresponding to the maximum temperature found along that path. This is implemented in Smokeview using a feature of the open graphics library (OpenGL)[5] that replaces the

a) 6 planes

9 planes

a) 12 planes

15 planes

Figure 6: Four images showing increasing number of planes used to visualize smoke and fire.

Figure 7: Example colormap used for converting temperature or HRRPUV values to color.

color in the background only if the currently drawn color has a greater value (in terms of the red, green or blue color components). Though the visualizations look more realistic it is not as flexible since the background has to be black (zero red, green blue color values) in order for the maximum replacement feature to work.

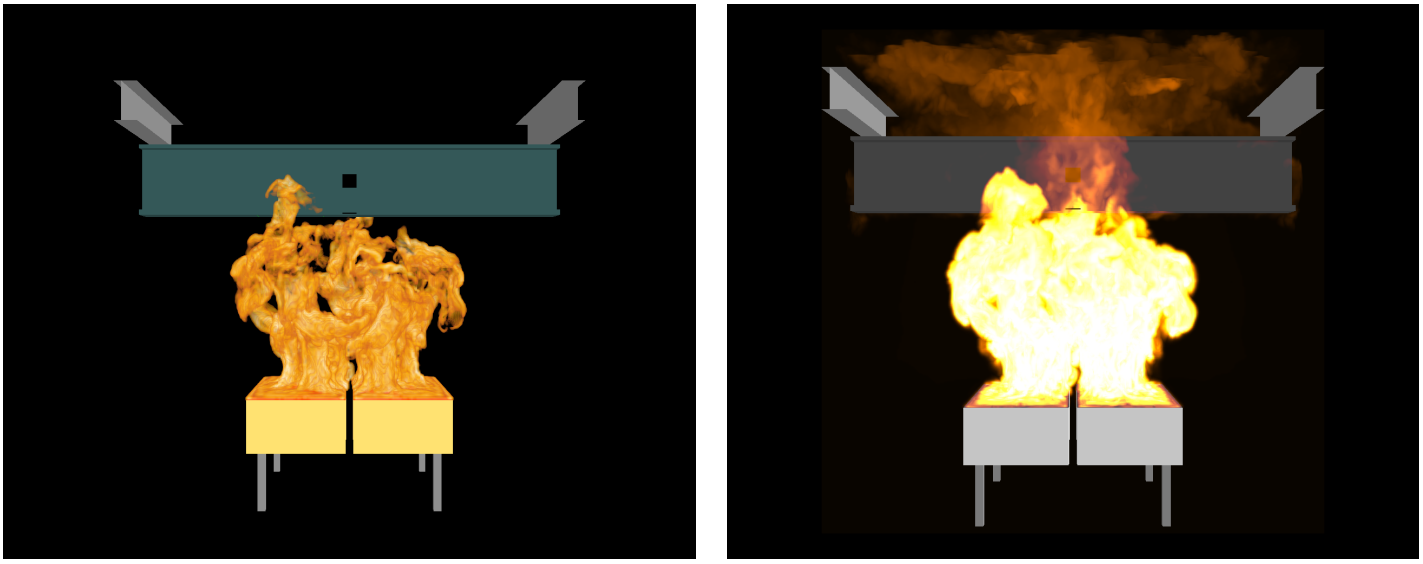


Figure 8: The image on the left was generated by blending colors found along each line of sight using opacities derived from smoke density. The image on the right was generated by using the color corresponding to the maximum temperature found along each line of sight.

SUMMARY

This note describes how Smokeview uses a simplified form of the radiative transport equation to display smoke and fire. This equation is solved using either the CPU or the GPU along with the video card by drawing a series of slices. Each slice is triangulated and drawn using opacities derived from Beer's law a simplified form of the radiative transport equation. The slices are colored using color maps. These algorithms may be improved in several ways. The integration of the radiative transport equation can have problems due to the limited numerical precision (8 bits) used to represent smoke opacity. One solution to this as illustrated in Figure 8 is to use choose maximum values rather than blending values. A better solution is to use data with more precision to perform the integration. Slice coloring may be made more quantitative by relating temperature to color using physics based rather than an assumed color map.

REFERENCES

[1] G.P. Forney. *Smokeview, A Tool for Visualizing Fire Dynamics Simulation Data, Volume I: User's Guide*. National Institute of Standards and Technology, Gaithersburg, Maryland, USA, sixth edition, May 2013. 1

[2] G.P. Forney. *Smokeview, A Tool for Visualizing Fire Dynamics Simulation Data, Volume II: Technical Reference Guide*. National Institute of Standards and Technology, Gaithersburg, Maryland, USA, sixth edition, May 2013. 1

[3] Robert Siegel and John R. Howell. *Thermal Radiation Heat Transfer*. Taylor & Francis, Inc., New York, NY, 4th edition, 2001. 1, 2, 8

[4] Marc Levoy. Display of surfaces from volume data. *IEEE Comput. Graph. Appl.*, 8(3):29–37, 1988. 1

[5] Dave Shreiner, Graham Sellers, John M. Kessenich, and Bill M. Licea-Kane. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3*. Addison-Wesley Professional, 8th edition, 2013. 8