Visualization and Modeling of Smoke Transport Over Landscape Scales

Glenn P. Forney
William E. Mell

NISTIR 7428

Visualization and Modeling of Smoke Transport Over Landscape Scales

Glenn P. Forney
William E. Mell
Fire Research Division
Building and Fire Research Laboratory

May 2007

Contents

List of Figures

Visualization and Modeling of Smoke Transport Over Landscape Scales

Glenn P. Forney

William E. Mell

Abstract

Computational tools have been developed at the National Institute of Standards and Technology (NIST) for modeling fire spread and smoke transport. These tools have been adapted to address fire scenarios that occur in the wildland urban interface (WUI) over kilometer-scale distances. These models include the smoke plume transport model ALOFT (A Large Open Fire plume Trajectory model) and WFDS (Wildland-urban interface Fire Dynamics Simulator) for fire spread and smoke transport in the wildland-urban interface. The visualization tool is called Smokeview. In this paper, an overview of the physical basis of the fire spread and smoke transport models will be discussed briefly along with the visualization of characteristic results using Smokeview. A technique will be described for visualizing smoke realistically and indications will be given how Smokeview can be applied to other fire models.

1   Introduction

NIST, the National Institute of Standards and Technology, has developed a suite of validated computational tools for the simulation and visualization of fire spread and smoke transport. These tools have been adapted for addressing fire scenarios that occur over distances on the order of kilometers, in particular, fire scenarios that address the wildland urban interface (WUI) problem. These models include the smoke plume transport model ALOFT (A Large Open fire plume Trajectory model[1]) and the model WFDS (Wildland-urban interface Fire Dynamics Simulator) for fire spread and smoke transport in the wildland-urban interface.[2, 3] The visualization tool is called Smokeview[4, 5]. These tools were developed with an emphasis on ease of use on affordable computer platforms. In this paper, the physical basis of the fire spread and smoke transport models will be discussed briefly along with presentation of characteristic results using Smokeview. Smokeview visualizes data in several ways: by animating two-dimensional slices of gas phase quantities such as temperature or smoke concentration, by animating flow vectors and animating surface conditions such as incident heat flux or burning rates at the forest floor and also by animating isosurfaces showing all places in the simulation scenario where a gas phase quantity takes on a specified value.

Smokeview is capable of rendering smoke and fire realistically using smoke concentrations computed by WFDS. By taking account of smoke properties, visibility can be assessed. The potential use of Smokeview for the visualization of predictions from other smoke transport models will also be discussed.

2   Modeling Fire Spread in the Outdoors

Computer modeling and visualization are important tools for understanding many complex processes. Fire behavior is no exception. Fire models range in complexity from simple algebraic correlations for predicting quantities such as flame heights or flow velocities to moderately complex differential equation based zone fire models for predicting quantities spatially averaged over large regions. Both classes of models work well when used appropriately, but break down for complicated flows or geometries. For such cases, computational fluid dynamics (CFD) techniques are required.

2.1   Computing Fire and Smoke Spread with CFD

The Fire Dynamics Simulator (FDS) has been developed at NIST to simulate the effects of fire and smoke spread[6, 7]. FDS predicts smoke and/or hot air flow movement caused by fire, wind, and other factors by numerically solving the

fundamental equations governing fluid flow, commonly known as the Navier-Stokes equations. The fire model WFDS builds on FDS by adding algorithms needed for solving the wildland interface problem. In particular, algorithms for modelling flame spread on or among the types of materials encountered in the WUI such as grassland, trees and shrubs. An experimental program is also proceeding to determine important material properties required by the WUI models[8].

WFDS is a physics-based fire-atmosphere coupled model that uses a form of CFD known as large eddy simulation (LES) to predict the thermal conditions resulting from a fire. These types of models require significantly more computational resources than the most commonly used fire spread models such as BEHAVE[9] or FARSITE[10], which are semi-empirical or empirical. However, there are a number of fire behavior problems, of increasing relevance, that are outside the scope of empirical and semi-empirical models. Examples are wildland-urban interface fires, assessing how well fuel treatments work to reduce the intensity of wildland fires, and investigating the mechanisms and conditions underlying blow-up fires and fire spread through heterogeneous fuels. WFDS uses approximations to the governing equations of fluid dynamics, combustion, and the thermal degradation of solid fuel. The LES approximation for solving the governing equations is a way of describing the effect of turbulence on the flow field. Turbulence is a phenomena that causes gases to mix over a wide range of length scales making it hard to replicate with even the fastest computers. The combustion model assumes that fuel and oxygen burn readily when mixed. The fire itself is a heat source term in the governing equations, creating buoyant motion that drives the smoke and hot gases throughout the domain of the simulation. The smoke yield is specified for a given fuel type based on measurements.

The downside of a CFD calculation such as WFDS is that depending on the computational resources, it can easily take days to run. As a result, parallelization techniques become important for splitting the work load among multiple computers, thereby speeding up the calculation and allowing results to be generated in a reasonable time.

2.2   Simulation Overview

WFDS, like any CFD model, requires that the scenario of interest be divided into small control volumes called computational cells. The model then computes the density, velocity, temperature, pressure and species concentration of the mass distribution (gas and particulates) in each cell based on the conservation laws of mass, momentum, and energy. WFDS simulates the spread of the fire by calculating the thermal degradation of the vegetative fuels. This is driven by the radiative and convective heat fluxes from the fire and depends on material properties of the vegetation.

The spatial resolution of the simulation depends on the number of cells used to discretize the volume of interest, much like the quality of a digital photograph depends mainly on the number of pixels. The number of cells is ultimately limited by the computing power available and the time available for the calculation. Current computers limit the number of cells to a few million. Many more cells may be used if calculations are run in parallel by splitting up the problem in to parts and solving each part on a different computer. Model users must balance the detail they want to incorporate with run times required to perform the computation. In general, finer grids result in longer calculations, but produce better results. Ultimately, one reaches a point of diminishing returns where the answer becomes insensitive to the increasing resolution of the grid.

The temporal resolution or time step size is determined from the grid resolution and the flow speed. The time step is chosen by WFDS, so that flow does not cross more than one grid cell during a single time step.

Both FDS and Smokeview would not have been possible without the recent advent of high-speed computers for performing computations, fast video cards for visualizing results and the Internet for exchanging information and ideas. These programs also would not have been possible without the research needed to develop the underlying fire models and the techniques needed to implement these models accurately and efficiently.

While simple line plots are adequate for visualizing the results of simple fire models, more sophisticated techniques are needed for interpreting the massive amounts of data generated by CFD models. This is where visualization tools such as Smokeview, the companion to WFDS, become essential.

3   Visualizing Smoke

Smokeview displays smoke allowing quantitative assessment using standard visualization techniques such as animated tracer particles that follow the flow, animated shaded 2D and 3D contours that display flow quantities and animated flow vectors that display flow quantities and direction. Smokeview also visualizes smoke realistically by converting

soot density to smoke opacity, displaying smoke as it would actually appear. These visualization techniques highlight different aspects of the underlying flow phenomena.

Visualization is essential at all stages of the modeling process. It is used before a run to verify the correctness of scenario geometry, (*e.g.* locations and size of simulation features), during a run to monitor the simulation (ensuring boundary flows are behaving as intended) and after the run has been completed to analyze the results.

Smokeview consists of about 70,000 lines of code. Most of it is written in C using standard libraries such as OpenGL[11] and GLUT[12] for graphics; GD (http://www.boutell.com/gd/), libpng (http://www.libpng.org/pub/png/) and libjpeg (http://www.ijg.org) for generating image files; and libzip (http://www.gzip.org/zlib/) for compression. A portion of Smokeview is written in Fortran 90 to input data generated by WFDS. The use of portable libraries allows Smokeview to run on many platforms including Windows, and various versions of Unix such as IRIX (for the SGI), Linux and OS X (for the Macintosh)[1].

Though Smokeview is usually used to visualize the results of WFDS or FDS simulations it is also used by other models. For example, Smokeview is used to visualize simulation results of the zone fire model CFAST[13]. An earlier version of Smokeview was adapted for visualization of the constituents of concrete during its formation (Concreteview[14]) and molecular dynamics simulations (Molecview[15]). The file formats that Smokeview uses for visualizing data are documented in both the Smokeview and FDS users guide[5, 7] enabling other fire models to make use of Smokeview as a visualization post-processor.

3.1 Scientific Visualizations

Figures 1, 2 and 3 show simulations of a crown fire spreading left to right from a virgin, untreated, forest stand into a treated stand. The fuel properties in the untreated stand are based on those measured in the black spruce and Jack pine overstory stands in the Northwest Territory of Canada during the International Crown Fire Modeling Experiments[16].

3.1.1 Particles

WFDS uses particles as modeling elements to account for heat transfer and momentum drag that occurs during tree burning. This is illustrated in Figure 1a. Particles may also be used to visualize the fire and smoke flow. Figure 1b shows a realistic view of the trees.

3.1.2 Slice planes

Smokeview allows animated color shaded contours of calculated gas quantities to be drawn at any horizontal or vertical plane in the simulation. To minimize file output, the user specifies the particular slice planes to be visualized. If disk space is not an issue, then the user may specify the entire 3D volume. Smokeview then allows the user to scroll through the 3D volume of data one slice at a time displaying any horizontal or vertical plane. Figure 2 illustrates temperature contours in a vertical plane through the center of a grassland fire. Temperatures below 100 °C are truncated. Figure 2a shows solid shaded contours while Figure 2b shows a vector plot. The data is colored the same way in both cases. Vector animations as illustrated in Figure 2b are better than regular slice animations at highlighting flow changes, especially in regions where temperatures are uniform.

3.1.3 Isosurfaces

Smokeview uses isosurfaces to identify *where* a specified level of a gas phase quantity occurs rather than *how much* . For example, WFDS uses a mixture fraction model to simulate combustion. In this model, there is a critical or stichometric mixture fraction value, such that regions greater than the critical value are fuel rich and regions less than the critical value are fuel lean. Burning then occurs, according to the model, on the level surface where the mixture fraction equals this stoichimetric value. Therefore, it is of interest to visualize these locations. This is done using animated isosurfaces.

The isosurfaces are generated at each desired time step using a marching cube algorithm[17] modified to remove ambiguities. A decimation procedure is used to reduce the number of resulting triangles by collapsing nodes of triangles with large aspect ratios and re-triangulating. This makes the isosurface look better and also reduces storage

---

[1]Any mention of commercial products is for information only; it does not imply recommendation or endorsement by NIST.
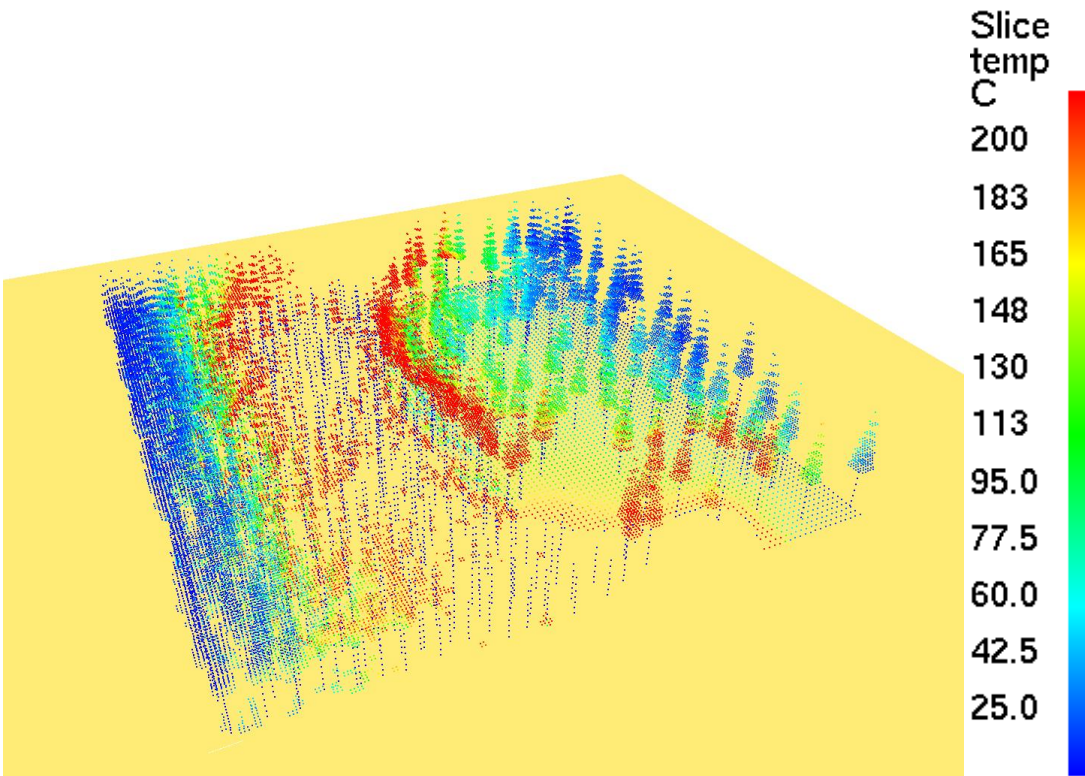
a) Trees drawn and modeled using particles. Particles release heat. Particles also drag or slow down air flowing past.
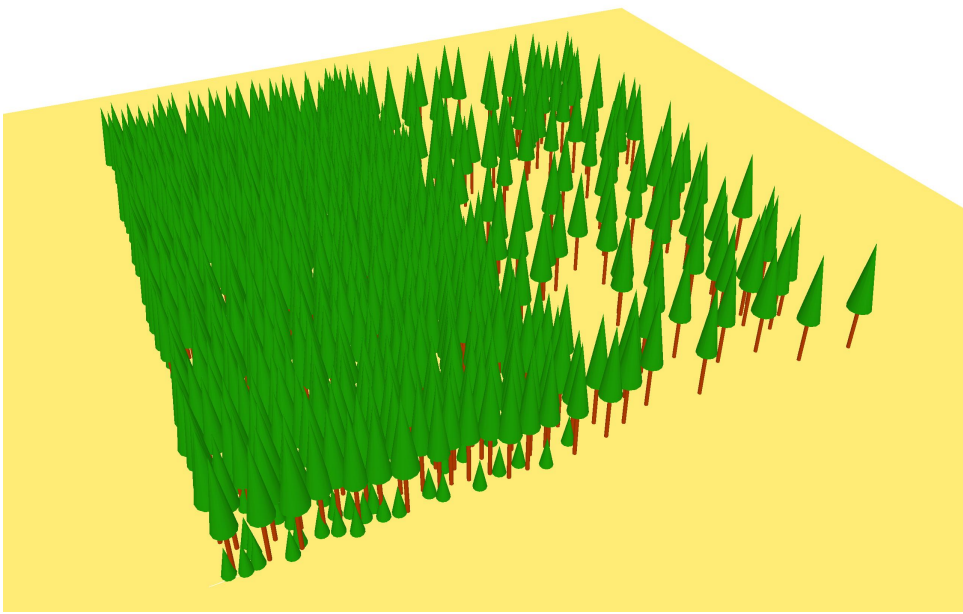
b) Trees drawn realistically.

Figure 1: Particle and realistic view of a crown fire simulation burning with wind from the left at 4.0 m/s (8.9 mph). Colors in the particle view represent temperature. Tree section is 50 m on a side.

a) Shaded temperature contours.   b) Colored flow vectors.

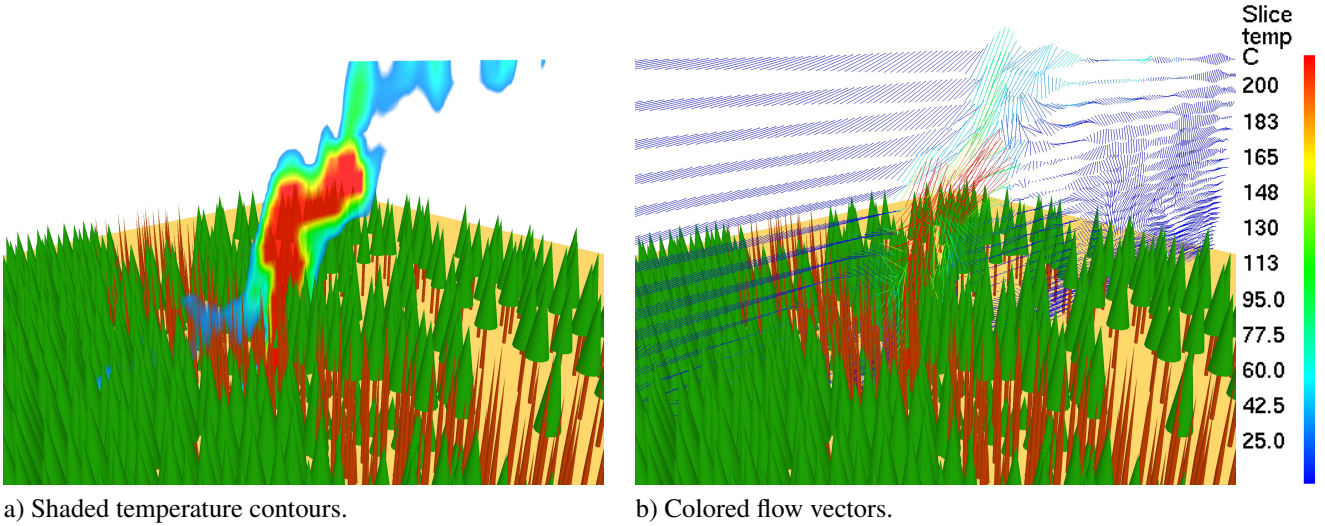Figure 2: Snapshot of shaded temperature contours and flow vectors through the center of a crown fire simulation with wind velocity boundary condition (from the left) of 4.0 m/s (8.9 mph). Region below 50 °C (122 °F) is hidden in the contour plot. Tree section is 50 m on a side.

requirements. Figure 3 illustrates the use of isosurfaces for visualizing the stoichimetric mixture fraction at the same time and view point as seen in Figure 2.

3.2 Realistic Visualization

Visualizing smoke realistically is challenging for three reasons. First, the storage requirements for describing smoke throughout the simulation scene at every time step can easily exceed the disk size capacities of present 32 bit operating systems which would typically be 2 GB. Second, the computation required both by the CPU and the video card to display each frame can easily exceed 0.1 s, the time corresponding to a 10 frame/s display rate. Finally, the physics required to describe smoke and its interaction with itself and surrounding light sources is complex and computationally intensive. Approximations and simplifications are required.

Smoke visualization techniques described previously, such as the use of tracer particles or shaded 2D contours are useful for analyzing data quantitatively, but are not suitable for applications where realism is required. Some examples of such applications are using Smokeview as a virtual fire fighter trainer or using Smokeview to examine the obscuration effects of smoke on an outdoor environment.

The approach used by Smokeview for visualizing smoke realistically is similar to that taken in Fedkiw *et. al.*[18] except that interactions with smoke and light are not considered (only the effects of smoke obscuration are visualized). The video hardware is exploited to perform a simple obscuration calculation by using OpenGL to display a series of partially transparent parallel planes. The planes are oriented to be perpendicular to the viewer's line of sight. The transparencies are computed based on physics using data derived from a WFDS calculation. Vertices in each plane are colored black or dark grey based on the estimated smoke albedo. The vertices are also assigned an OpenGL α opaqueness parameter. The assigned value depends on the optical smoke thickness, with 0.0 used for completely transparent smoke and 1.0 for completely opaque. Figure 4 gives a flowchart describing the process FDS uses to pre-compute smoke data and Smokeview uses update and draw the data.

3.2.1 Computing Opaqueness

The α values are pre-computed by WFDS using Beer's law[19]

$$\alpha = 1 - \exp(-ks\Delta x) \qquad (1)$$

for a particular view direction (down the x axis) where Δx is this distance between two nodes, k is the soot mass extinction coefficient and s be the soot density. Beer's law is an empirical relationship relating light absorption to the
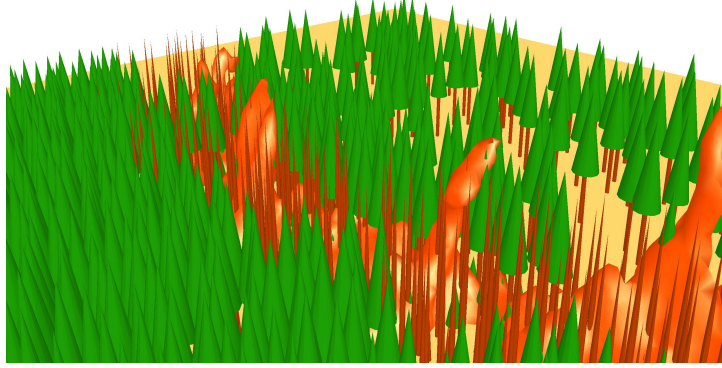
Figure 3: Snapshot of an isosurface for of stoichimetric mixture fraction, a reasonable surrogate for visualizing flames.

material properties of the media the light is traveling through, in this case soot or smoke. Smokeview currently does not consider light scattering effects with smoke.

3.2.2 Adjusting Opaqueness

The absorption parameter, α, needs to be adjusted when the view direction is not aligned along the axis orthogonal to the viewing planes (as in Figure 5), the distance between adjacent smoke planes changes, or viewing planes are skipped.

Ten million exponential operations per second are required to display smoke with corrected α's at 10 frames per second if the simulation has grid dimensions of $100 \times 100 \times 100$. Recent advances in CPU and video hardware makes these types of visualizations possible. These corrections may also be performed in the video card (GPU), resulting in increased display rates because the GPU performs the corrections simultaneously at all or many of the grid nodes rather than one at a time as the CPU would.

The α obscurations are pre-computed using the distance Δx between adjacent planes along the x-axis. The adjusted $\hat{\alpha}$ expressed in terms of $\Delta \hat{x}$ is given by

$$\hat{\alpha} = 1 - \exp(-ks\Delta\hat{x}) \tag{2}$$

where $\Delta \hat{x}$ is the distance between planes along the line of site. Equations (1) and (2) may be used to solve for $\hat{\alpha}$ in terms of α to obtain

$$\hat{\alpha} = 1 - (1-\alpha)^{\Delta\hat{x}/\Delta x} \tag{3}$$

after noting that

$$1 - \hat{\alpha} = \exp(-ks\Delta\hat{x}) = \exp(-ks\Delta x)^{\Delta\hat{x}/\Delta x} = (1-\alpha)^{\Delta\hat{x}/\Delta x}$$

The computation of equation (3) is expensive because the exponential is computed at each grid node for every time step. In addition, numerical cancellation may occur for small α leading to loss of significant digits. Both problems may be solved by expanding equation (3) in a Taylor series and keeping only the first few terms:

$$\hat{\alpha} \approx \alpha r - \frac{\alpha^2}{2} r(r-1) + \frac{\alpha^3}{6} r(r-1)(r-2)$$

where $r = \sec(\theta) = \Delta\hat{x}/\Delta x = ||x_p - x_e||/n \cdot (x_p - x_e)$, n is the unit vector normal to the current plane being drawn, θ is the angle between the view direction and n, x_e is the observers position and x_p is the vertex being drawn (along the view direction). These terms are illustrated in Figure 5.
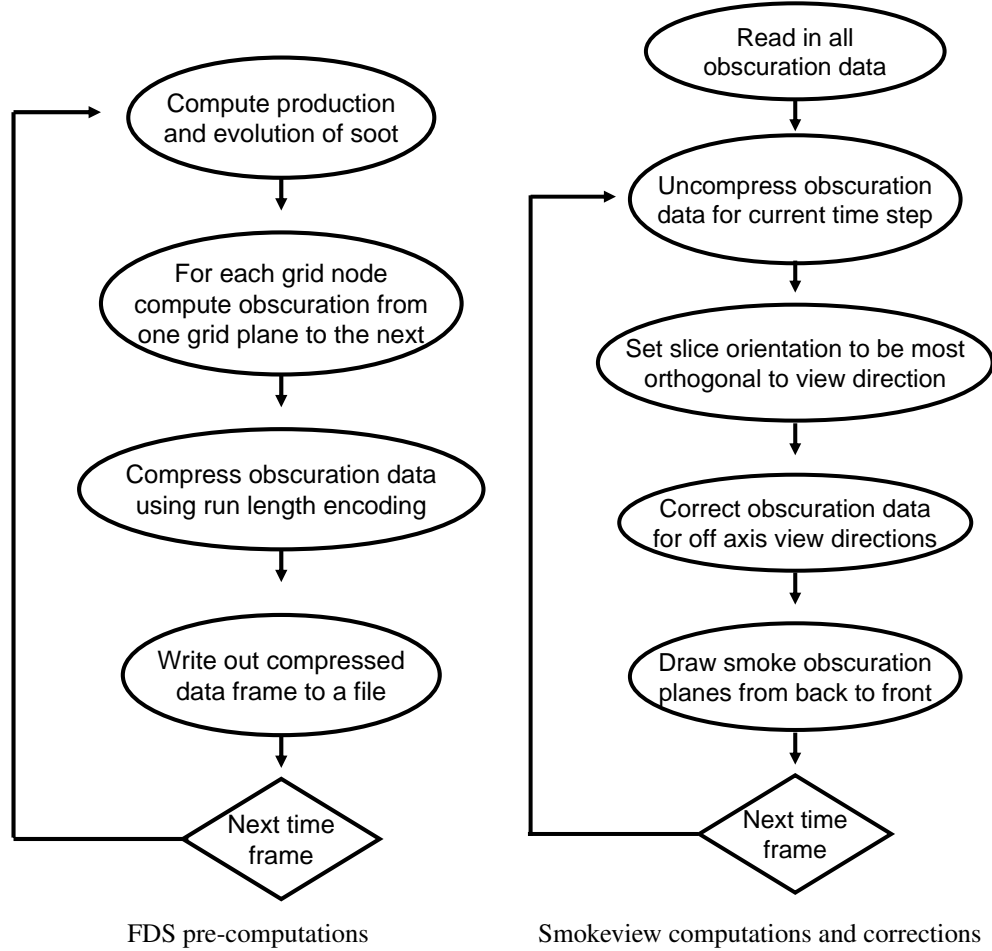
Figure 4: Flowchart of FDS and Smokeview computations for visualization smoke realistically. FDS computes obscuration parameters from one grid plane to the next. Smokeview corrects these parameters to account for non-axis aligned view directions.

When planes are skipped, equation (3) may be simplified. In particular, when every 2nd plane is skipped, $\Delta \hat{x} / \Delta x = 2$, so that equation (3) simplified to

$$\hat{\alpha} = 1 - (1-\alpha)^2 = 2\alpha - \alpha^2$$

The video hardware uses α values contained in the smoke planes to obscure the background much like a camera uses a neutral density filter to darken a scene. Extending the analogy, Smokeview uses one neutral density filter (numerically) for each plane of smoke data. On a node by node basis then, each smoke plane obscures the current image stored in the OpenGL back buffer by the amount $(1 - \alpha)$ to form a new back buffer image. A simplistic description of one step of this process is given by

$$\text{new buffer image} = (1 - \alpha) \times \text{old buffer image}$$

This process is repeated for each smoke plane. Figure 6 illustrates this process for smoke and fire spread over a large grassland fire simulated by WFDS. The fuel properties are based on those measured in grassland plots during experiments conducted in Australia. A further description of the fuels and simulations is in Mell *et. al.*[3].

The visualization is performed by displaying a series of partially transparent planes. For illustration, these planes are made more conspicuous (in Figure 6a) by skipping smoke planes (displaying every third plane) and orienting them
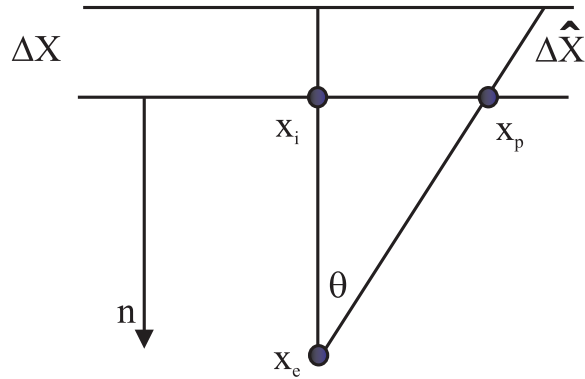
Figure 5: Diagram illustrating adjustment needed to opaqueness parameter, α, for non axis aligned views. The α value along the ray containing the $\hat{x}$ segment needs to be larger to account for the longer path length.

along the 'x' axis. Figure 6b shows the visualization as it normally appears with all slice planes shown and oriented along a plane most perpendicular to the view direction.

3.2.3 Compression

The opacity parameters are computed at each node for all time steps. As noted earlier, the space required to store these values can easily exceed the 2 GB file size limit found on 32 bit operating systems such as Linux or Windows. Compression techniques are required to reduce storage requirements.

Compression for this application occurs in two steps. First, a four to one compression level is achieved by using Beer's law to convert soot density, a four byte quantity, to opacity, a one byte quantity. Video cards presently use only one byte to represent opacity. Next, the sequence of opacity values are compressed using run-length encoding. Run length encoding works as follows.

1. Represent four or more consecutive identical characters as #*nc* where # is a special character denoting the beginning of a repeated sequence, *n* is the number of repeats and *c* is the character repeated. *n* can be up to 254 (255 is used to represent the *special* character).

2. Represent characters not repeated four or more times as is.

Run length encoding was chosen because it provides a reasonably good compression ratio, is simple to code and most importantly can be decompressed efficiently. This last property is important for any compression scheme chosen because it is a rate limiting step in the process that Smokeview uses to display smoke data. The CPU time required to compute the smoke flow can easily exceed one minute of CPU time per outputted time step, so extra time used to produce a more compact file is affordable. However, each data frame is decompressed *on the fly* so a compression format that can be rapidly decompressed is critical.

Smokezip is a software program developed as a companion to Smokeview and FDS to compress FDS daa files. A second compression scheme is used by Smokezip to compress FDS files even more compactly. Smokezip uses the ZLIB compression library available at (http://zlib.net/).

3.2.4 Smoke plane orientation

Smoke opacity data computed as described in previous sections is stored in a 3D array. This array corresponds to the solution domain as setup in an FDS input file (or some other model). Smoke planes are drawn in Smokeview through this data. The orientation is chosen to be most perpendicular to the viewers line of sight. A plane orientation exactly perpendicular to the view direction could be drawn if one is willing to pay the added CPU cost of interpolating opacity values between grid nodes.

Figure 7 illustrates this process showing three view directions and the corresponding smoke plane orientations that would be used. Off-axis viewing is minimized by selecting the view planes orientation that minimizes the angle
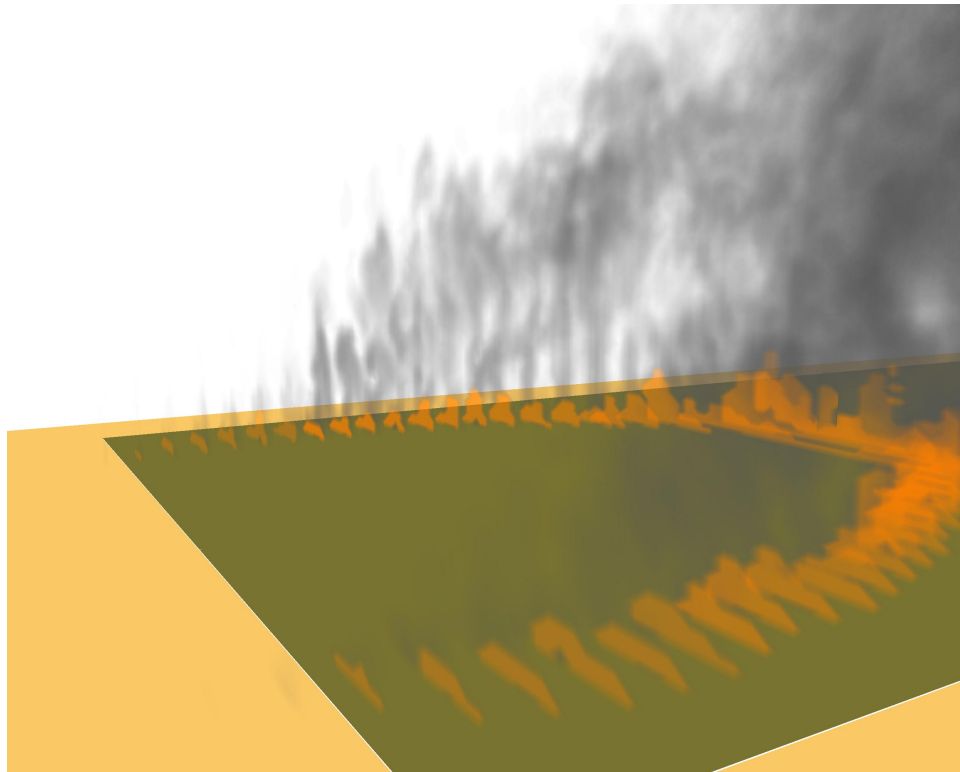
a) slices skipped and oriented along 'x' directions
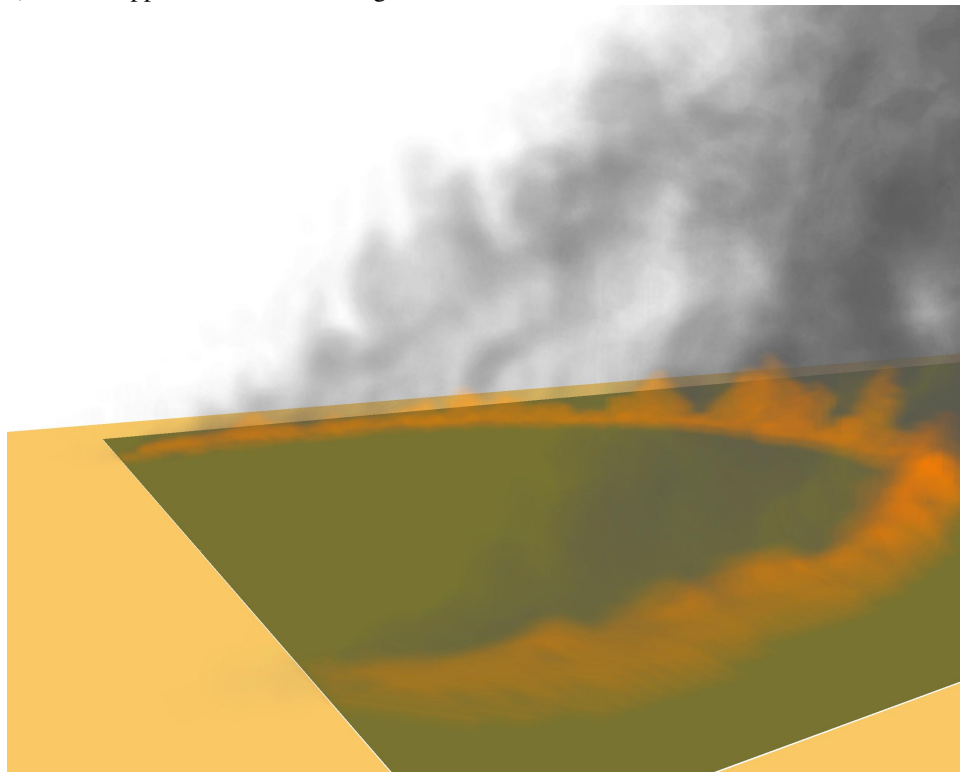
b) all slices shown and oriented towards viewer

Figure 6: Realistic visualization of a large grass fire simulated using WFDS. Planes in the top image are drawn to be conspicuous by skipping 2 out of every 3 planes and by aligning planes along the x axis. All planes in the bottom image are displayed (none are skipped) and they are aligned to be closest to perpendicular of all possible plane orientations.
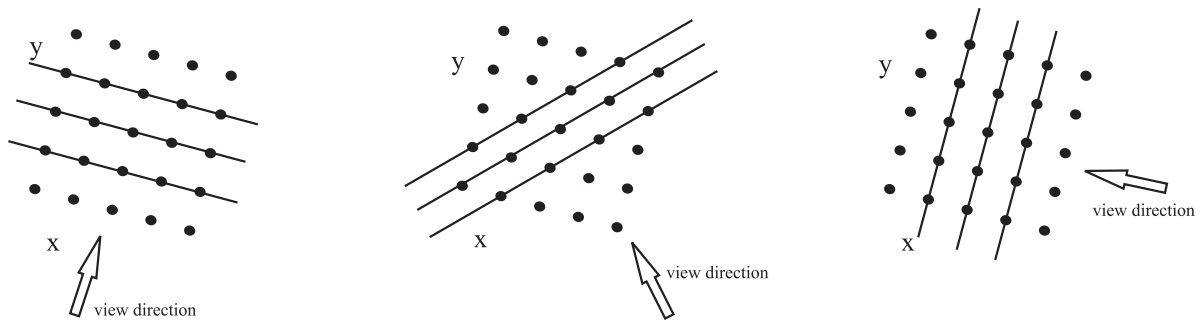
a) smoke planes parallel to *y* axis   b) smoke planes parallel to $y = x$ axis)   c) smoke planes parallel to *x* axis

Figure 7: View of smoke planes from above. Smoke Planes are oriented so that are "most perpendicular" to the line of sight
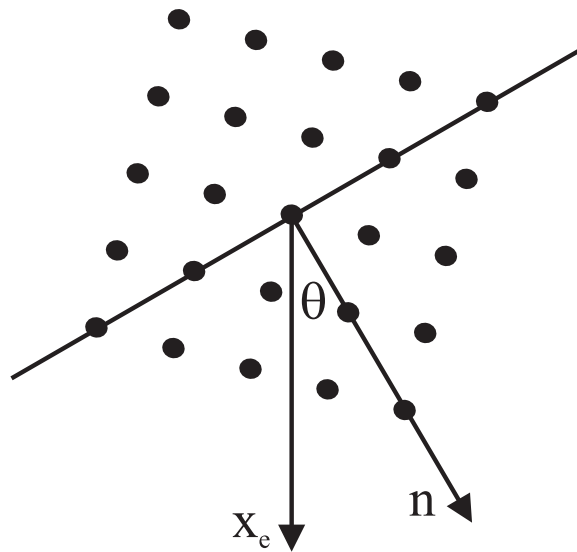

Figure 8: Diagram illustrating the angle between the line of sight and smoke plane normal vector. View planes are chosen to minimize this angle.

between the planes normal direction and the view direction. This angle, θ, is illustrated in Figure 8, and is given by

$$\cos(\theta) = \frac{n \cdot v_e}{||n|| \, ||v_e||}$$

where n is normal vector for the candidate smoke plane, and v_e is the view direction vector. In OpenGL, the view direction vector, v_e, is computed by simply obtaining the modelview matrix, M and multiplying it by the vector, $(0,0,1)^T$ or equivalently the third row of M.

4   Summary

Smokeview is used to visualize data simulated by WFDS, the NIST wildland fire dynamics simulator. Smokeview uses several techniques to visualize data, some scientific and some realistic. The realistic technique uses the video hardware found on modern computers to convert soot densities computed by WFDS to opacities displayed on the computer screen. The local values of obscuration are computed by WFDS for all grid nodes using soot density, grid spacing, and the soot mass extinction coefficient appropriate for visible light and the fuel being burned. This is a one time computation. Smokeview, on the other hand, integrates these local values along the line of sight each time the view position or direction changes and for each frame of data.

Smokeview is a scientific visualization tool that has been adapted to display data for several types of applications and it may be adapted to display data simulated by fire models used to solve other aspects of the WUI problem.

5   Additional Information

Fire on the Web, http://fire.nist.gov, contains resources related to fires including images and movies of real fires along with measured heat release data, software for modeling fire flow and links to NIST publications related to fire. Information on the fire modeling software FDS and Smokeview may be found at http://fire.nist.gov/fds . Further information about the wildland urban interface adaptation of FDS (WFDS) may be found at http://www2.bfrl.nist.gov/userpages/wmell/public.html.

References

[1] W D. Walton, K. B. McGrattan, and J. V. Mullin. ALOFT-PC: A Smoke Plume Trajectory Model for Personal Computers. Technical Report NIST SP 995, National Institute of Standards and Technology, Gaithersburg, Maryland, March 2003.

[2] David D. Evans, Ronald G. Rehm, and Elisa S. Baker. Physics-Based Modeling for WUI Fire Spread - Simplified Model Algorithm for Ignition Structures by Burning Vegetation. NISTIR 7179, 2004.

[3] William Mell, Mary Ann Jenkins, Jim Gould, and Phil Cheney. A physics-based approach to modeling grassland fires. *International Journal of Wildland Fire*, 16(1):1–22, 2007.

[4] G. P. Forney, D. Madrzykowski, K. B. McGrattan, and L. Sheppard. Understanding fire and smoke flow through modeling and visualization. *Computer Graphics and Applications*, 23(4):6–13, 2003.

[5] G.P. Forney and K.B. McGrattan. User's Guide for Smokeview Version 4 - A Tool for Visualizing Fire Dynamics Simulation Data. NIST Special Publication 1017, National Institute of Standards and Technology, Gaithersburg, Maryland, August 2004.

[6] K.B. McGrattan (editor). Fire Dynamics Simulator (Version 4), Technical Reference Guide. NIST Special Publication 1018, National Institute of Standards and Technology, Gaithersburg, Maryland, July 2004.

[7] K.B. McGrattan and G.P. Forney. Fire Dynamics Simulator (Version 4), User's Guide. NIST Special Publication 1019, National Institute of Standards and Technology, Gaithersburg, Maryland, July 2004.

[8] Samuel L. Manzello, Thomas G. Cleary, John R. Shields, and Jian Yang. On the ignition of fuel beds by firebrands. *Fire and Materials*, 30:77–87, 2006.

[9] P. L. Andrews and C D. Bevins. BEHAVE Fire Modeling System: Redesign and Expansion. *Fire Management Notes*, 59:16–19, 1999.

[10] M. A. Finney and P. L. Andrews. A Program for Fire Growth Simuluation. *Fire Management Notes*, 59:13–15, 1999.

[11] OpenGL Architecture Review Board, Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis. *OpenGL Programming Guide - The Official Guide to Learning OpenGL, Version 2*. Addison-Wesley, Stoughton, Massachussets, 5 edition, 2005.

[12] Mark J. Kilgard. *OpenGL Programming for the X Window System*. Addison-Wesley Developers Press, Reading, Massachussets, 1996.

[13] W. W. Jones, R. D. Peacock, G. P. Forney, and P. A. Reneke. CFAST, Consolidated Model of Fire Growth and Smoke Transport (Version 5. technical reference guide. NIST Special Publication 1030, National Institute of Standards and Technology, Gaithersburg, Maryland, October 2004.

[14] D. P. Bentz and G. P. Forney. User's Guide to the NIST Virtual Cement and Concrete Testing Laboratory, Version 1.0. Technical Report NISTIR 6583, National Institute of Standards and Technology, Gaithersburg, Maryland, November 2000.

[15] S. I. Stoliarov, P. R. Westmoreland, M. R. Nyden, and G. P. Forney. A reactive dynamics model of thermal decomposition in polymers: I. Poly (methyl methacrylate). *Polymer*, 44(3):883–894, 2003.

[16] M. Alexander, C. N. Stefner, J.A. Mason, B. J. Stocks, J. R. Hartely, M. A. Maffey, B. M. Wotton, S. W. Taylor, N. Lavoie, and G. N. Dalrympte. Characterizing the Jack pine - black spruce fuel complex of the international crown fire modeling experiments (ICFME). NOR-X 393, Canadian Forest Service, Northern Forestry Centre, 2004.

[17] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169. ACM Press, 1987.

[18] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 15–22. ACM Press / ACM SIGGRAPH, 2001.

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