Structural Modeling of Flames for a Production Environment

Arnauld Lamorlette
PDI/DreamWorks

Nick Foster
PDI/DreamWorks

ABSTRACT

In this paper we describe a system for animating flames. Stochastic models of flickering and buoyant diffusion provide realistic local appearance while physics-based wind fields and Kolmogorov noise add controllable motion and scale. Procedural mechanisms are developed for animating all aspects of flame behavior including moving sources, combustion spread, flickering, separation and merging, and interaction with stationary objects. At all stages in the process the emphasis is on total artistic and behavioral control while maintaining interactive animation rates. The final system is suitable for a high volume production pipeline.

CR Categories: I.3.7 (Computer Graphics): Three-Dimensional Graphics and Realism – A nimation; I.3.5 (Computer Graphics): Computational Geometry and Object Modeling – Physically Based Modeling.

Keywords: animation systems, fire, flames, convection, physically-based modeling, wind fields, Kolmogorov spectrum.

1. INTRODUCTION

Numerical simulations of natural phenomena are now routinely used for animation or special effects projects. They are "coming of age" for computer graphics in the sense that numerical components have been added to the base physical models for the sole purpose of controlling and stylizing simulations for visual effect. It is no longer enough that numerical techniques are massaged to make such methods efficient for animation. They should also incorporate techniques for controlling both the look and behavior of the phenomena. Recently, liquids and gases in particular have proved amenable to this approach [Foster and Fedkiw 2001; Fedkiw et al. 2001]. That is, efficient numerical methods have been combined with physics-based control methods to create general animation systems.

It would seem sensible then, to apply this same approach to the modeling and animation of fire. Experience suggests however, that there exists a subtle difference in the requirements of fire animation over phenomena such as fluids. The latter are often required as environment elements. They need to interact with characters and objects in a believable way. Rarely are these elements the main "actor" in a scene. When they are, (the Abyss and The Perfect Storm spring to mind) the physics rules are either obeyed closely, or can clearly be relaxed because the fluid is already acting atypically.

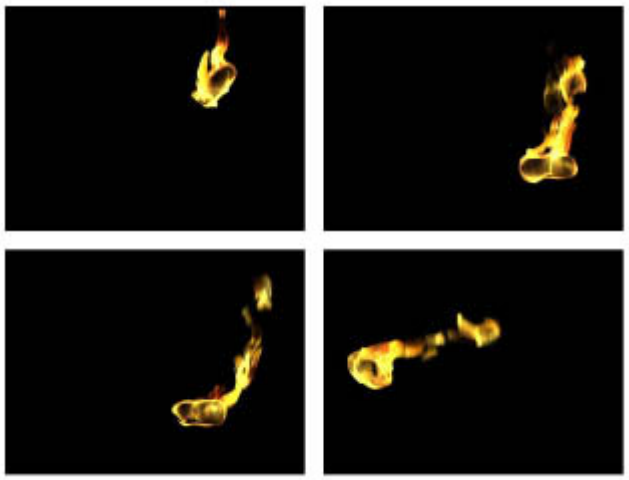

Figure 1: A burning torch is waved through the air.

Fire on the other hand, is a dramatic element that requires the maximum level of control possible while maintaining a believable appearance. We expect fire to look complex and unpredictable, while at the same time having a recognizable structure according to the conditions under which it is burning. That complexity by itself makes direct numerical simulation of fire much less attractive than for other phenomena for three reasons:

- Numerical simulations scale poorly. As the resolution of a 3D simulation increases, computational complexity increases by at least $O(n^3)$ [Foster and Metaxas 1996; Stam and Fiume 1995]. The resolution required to capture the detail in even a relatively small fire makes simulation expensive.
- The number of factors that affect the appearance of fire under different circumstances leads to an inter-dependent simulation parameter space. It's difficult for developers to place intuitive control functions on top of the underlying physical system.
- Fire is chaotic. Small changes in initial conditions cause radically different results. From an animation standpoint it's difficult to iterate towards a desired visual result.

Together, these restrictions make numerical simulation a poor choice as a basis for a large-scale fire animation system, where control and efficiency are as equally important as realism.

This paper presents a different approach to modeling fire. Instead of modeling the physics of combustion, we achieve the trade-off between realism, control, and efficiency by recognizing that many of the visual cues that define fire phenomena are statistical in nature. We then separate these vital cues from other components that we wish to artistically control. This provides us with a number

of distinct structural elements. Some of these elements are physics-based, but in general they need not be. Each structural element has a number of animation parameters associated with it, allowing us to mimic a variety of both real-world and production-world fire conditions.

Realism is achieved by basing structural state changes on the measured statistical properties of real flames. In addition, we define a set of large-scale procedural models that define locally how a group of flame structures evolve over time. These include physics-based effects such as fuel combustion, diffusion, or convection, as well as pure control mechanisms such as animator-defined time and space curves. Together, the structural flame elements and procedural controls form the basis of a system for modeling and animating a wide range of believable fire effects extremely efficiently in the context of a direction driven animation pipeline.

2. PREVIOUS WORK

Previous work on modeling dynamic flames falls loosely into two categories: direct numerical simulation and visual modeling. Direct simulations have achieved realistic visual results when modeling the rotational motion due to the heat plume around a fire [Chiba et al. 1994; Inakge 1990]. Simulation has also proved efficient at modeling the behavior of smoke given off by a fire [Rushmeier et al. 1995; Stam and Fiume 1993], and the heat plume from an explosion [Yngve et al. 2000]. For the shape and motion of flames however, simulation has not been so useful. Numerical models from Computational Fluid Dynamics have not proven amenable to simplification without significant loss of detail [Drysdale 1998]. Without that simplification they are an expensive option for large-scale animations.

Visual modeling has focused on efficiency and control. Particle systems are the most widely used fire model [Nishita and Dobashi 2001; Stam and Fiume 1995]. Particles can interact with other primitives, are easy to render, and scale linearly (if there are no inter-particle forces). The problem of realism falls solely on the shoulders of the animator though. Force fields and procedural noise can achieve adequate looking large-scale effects due to convection, but it is very difficult to come up with a particle-based model that accurately captures the spatial coherence of real fire. Flame coherence has been modeled directly using chains of connected particles [Beaudoin and Paquet 2001; Reeves 1983]. This retains many of the advantages of particle systems while also allowing animators to treat a flame as a high level structural element.

3. ESSENTIAL MODEL

The system we have developed to use as a general fire animation tool has eight stages. The key to its effectiveness is that many stages can be either directly controlled by an animator, or driven by a physics-based model. The components are as follows:

- Individual flame elements are modeled as parametric space curves. Each curve interpolates a set of points that define the spine of the flame.
- The curves evolve over time according to a combination of physics-based, procedural, and hand-defined wind fields. Physical properties are based on statistical measurements of natural diffusion flames. The curves are frequently re-sampled to ensure continuity, and to provide mechanisms to model flames generated from a moving source.

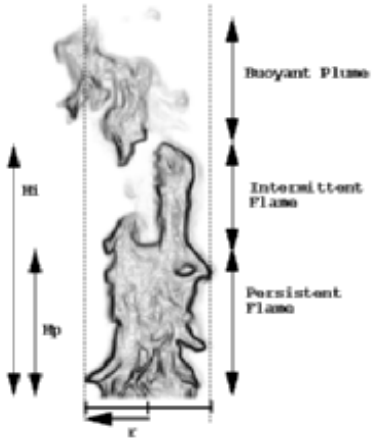

Figure 2: Classic three stage buoyant diffusion flame.

- The curves can break, generating independently evolving flames with a limited lifespan. Engineering observations provide heuristics for both processes.
- A cylindrical profile is used to build an implicit surface representing the oxidization region, i.e. the visible part of the flame. Particles are point sampled close to this region using a volumetric falloff function.
- Procedural noise is applied to the particles in the parameter space of the profile. This noise is animated to follow thermal buoyancy.
- The particles are transformed into the parametric space of the flame's structural curve. A second level of noise, with a Kolmogorov frequency spectrum, provides turbulent detail.
- The particles are rendered using either a volumetric, or a fast painterly method. The color of each particle is adjusted according to color properties of its neighbors, allowing flame elements to visually merge in a realistic way.
- To complete the system, we define a number of procedural controls to govern placement, intensity, lifespan, and evolution in shape, color, size, and behavior of the flames.

The result is a general system for efficiently animating a variety of natural fire effects. The cost of direct simulation is only incurred when it is desired. Otherwise, the system provides complete control over large-scale behavior.

4. STRUCTURAL ELEMENTS

The fire animation system is built up from single flames modeled on a natural diffusion flame (see Figure 2). Observed statistical properties of real flames are used wherever possible to increase the realism of the model.

4.1 Base Curve

The basic structural element of the flame system is an interpolating B-Spline curve. Each curve represents the central spine of a single flame. The flames themselves can merge or split (see Section 4.3) but all of the fire phenomena we model are built from these primitives.

In the first frame of animation for which a particular flame is active, a particle is generated at a fixed point on the burning

surface and released into a wind field. The particle is advanced in the wind field for a frame using an explicit Euler integration method (Runge-Kutta is sufficient and completely stable in this case), and a new particle is generated at the surface. The line between these two points is sampled so that *n* control points are evenly distributed along it. For each additional frame of animation, they convect freely within the wind field.

After convection, an interpolating B-Spline is fitted so that it passes through all the points. This curve is then parametrically re-sampled, generating a new set of *n* points (always keeping the first and last unchanged). This re-sampling ensures that whatever the value of *n*, visual artifacts don't appear in the structure if control points cluster together, or as a side effect of large time steps.

4.2 Flame Evolution

During the animation, the control points for the structural curve evolve within a wind field. This evolution is for providing global shape and behavior. Specific local detail due to fuel consumption and turbulence at the combustion interface is added later.

The motion is built from four main components: convection, diffusion, initial motion, and buoyancy. Initial motion is applied only at the very base of the curve. If the fire-generating surface has a velocity $\mathbf{V}$, the particle, $\mathbf{P}_0$, is given an initial velocity, $-\mathbf{V}$. The flame is then generated in a stationary reference frame. Because flames move with the participating media (i.e., with no specific inertia), this provides an efficient way of taking account of moving sources.

Buoyancy is due to the tendency for hotter (less dense) air to rise. The region around the visible part of the flame is relatively homogeneous in terms of temperature (at least with respect to the ambient temperature). We chose to model buoyancy as a direct linear upward force. The resulting flow is laminar in nature, so we add rotational components in the form of simulated wind fields and a noise field generated from a Kolmogorov spectrum.

The equation of motion for a structural particle, $\mathbf{p}$, is defined by

$$\frac{d\mathbf{x}_p}{dt} = w\left(\mathbf{x}_p, t\right) + d\left(T_p\right) + \mathbf{V}_p + c\left(T_p, t\right) \qquad (1)$$

where, $w(\mathbf{x}_p,t)$, is an arbitrary controlling wind field, $d(T_p)$, the motion due to direct diffusion, $\mathbf{V}_p$, motion due to movement of the source and, $c(T_p, t)$, the motion due to thermal buoyancy. T_p is the temperature of the particle. Diffusion is modeled as random Brownian motion scaled by the temperature T_p. Thermal buoyancy is assumed to be constant over the lifetime of the particle, therefore

$$c\left(T_p,t\right) = \quad g_y\left(T_0 \quad T_p\right)t_p^{\,2} \qquad (2)$$

where is the coefficient of thermal expansion, g_y is the vertical component of gravity, T_0 is the ambient temperature, and t_p is the age of the particle. Equations (1) and (2) allow us to combine accurate simulated velocity fields with ad-hoc control fields without producing visual discontinuities.

4.3 Separation and Flickering

We model flame separation and flickering as a statistical process. The classic regions of a free-standing diffusion flame are illustrated in Figure 2. The intermittent region of the flame is defined over the range $[H_p, H_i]$. From its creation a flame develops until it reaches

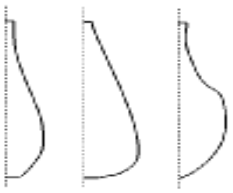

Figure 3: Different normalized flame profiles for a candle flame, torch flame, and camp fire flames respectively**.**

H_i. At that point we periodically test a random number against the probability function,

$$D(h) = \frac{1}{\sqrt{2}\,(H_i \quad H_p)/2}\; e^{\;h \quad h \;\frac{|\mathbf{V}_c|}{f}^2 \Big/ 2\left((H_i \quad H_p)/2\right)^2} dh \qquad (3)$$

to determine whether the flame will flicker or separate at a length h. The frequency *f*, is the approximate breakaway rate in Hz, while $\mathbf{V}_c$ is the average velocity of the structural control points. From observation [Drysdale 1998], $f = (0.50 \quad 0.04)(g_y/2r)^{1/2}$ for circular sources with a radius r. If necessary, $D(h) \quad 1$ for $h > H_{max}$, where H_{max} is some artistically chosen limit on flame length.

Once separation occurs, a region of the structural curve is split off from the top of the flame. This region extends from the top of the flame to a randomly selected point below. We could find no measured data that describes the size distribution of separated flames. In that absence, this point is selected using a normal distribution function with a mean of $H_p+(H_i-H_p)/2$ and standard deviation of $(H_i-H_p)/4$. The control points representing the buoyant region are fitted with a curve, and re-sampled in the same way as for the parent structure (Section 4.1). The number of points split off is not increased back to *n* in this case however. This prevents additional detail appearing in the buoyancy region of the flame where no additional fuel could be added. During its lifetime, the structural particles that make up the separated flame follow Eq (1) as before.

Without modeling the combustion process there's no accurate way to determine the fuel content of the breakaway flame. Therefore there's no good model for how long it will remain visible. Each separate flame is given a life-span of Ai^3, where *i* is a uniform random variable in the range [0,1], and *A* is a length scale ranging from $1/24^{th}$ second for small flames up to 2 seconds for a large pool fire. The cube ensures that most breakaway flames are short lived. There's no reason why the buoyant flame can't separate again, although in general the entrainment region quickly dominates as there's little airborne fuel in a buoyant flame except for oxygen starved fires.

4.4 Flame Profiles

With the global structure and behavior of a flame defined, we now concentrate on the visible shape of the flame itself. A flame is a combustion region between the fuel source and an oxidizing agent. The region itself is not clearly defined, so we require a model that is essentially volumetric.

One such model treats each segment (the line between two adjacent control points) of the structural curve as a source for a potential field [Beaudoin and Paquet 2001]. The total field for a

flame is the summation of the fields for each of its segments. The flame is rendered volumetrically using different color gradients for various iso-contours of the potential field. This method gives good results for small fires like candle flames, and handles flame merging efficiently. Stylization is difficult however, requiring contraction functions for each desired flame shape. In addition, turbulent noise has to be built directly into the potential function as there is no subsequent transformation stage before rendering.

To retain complete control over basic shape, we represent a flame using a rotationally symmetric surface based on a simple two-dimensional profile. The profile is taken from a standard library and depends on the scale of flame effect that we wish to model (see Figure 3 for examples). We have had good results from hand drawn profiles, as well as those derived from photographs.

We then define the light density of the visible part of the flame as

$$d(\mathbf{x}) = \frac{I}{1 + \left| \mathbf{x}_{f(\mathbf{x})}^{\ 2} \quad \mathbf{x}^2 \right|} \tag{4}$$

where $\mathbf{x}_{f(\mathbf{x})}$ is the closest point to $\mathbf{x}$ on the parametric surface defined by rotating the profile, and I is the density of combustion at the surface (normalized to one for this work). Equation (4) defines a simple volumetric density function for our flame.

In order to transform and displace this starting shape into an organic and realistic looking flame, we need to transform Eq (4) onto the flame's structural curve. The density function is first point sampled volumetrically using a Monte Carlo method. The point samples do not survive from frame to frame. The sampling allows us to deform the density function without having to integrate it. It is important though, that the rendering method employed (see Section 5) be independent of sample density, so that the flame does not appear pointillistic. Once the density function has been approximated as particles, we displace and transform them to simulate the chaotic process of flame formation.

4.5 Local Detail

Two levels of structural fluctuation are applied to the point samples that define the visible part of the flame. Both animate in time and space. The first directly affects overall shape by displacing particles from their original sample positions, while the second simulates air turbulence. We define buoyancy noise, which represents the combustion fluctuation at the base of the flame. It propagates up the flame profile according to the velocities of the nearest structural particles (Section 4.1). There is no real physical data to go on here, but a noise function that visually looks good for this is Flow Noise [Perlin and Neyret 2001]. The rate of rotation of the linear noise vectors over time is inversely proportional to the diameter of the flame source, i.e., large flames lead to quicker vector rotation.

The participating medium also causes the flame to distort over time. At this scale, a vector field created using a Kolmogorov spectrum exhibits small-scale turbulence and provides visual realism. Kolmogorov noise is relatively cheap to calculate and can be generated on a per-frame basis (see [Stam and Fiume 1993] for some applications of the Kolmogorov spectrum).

Each sample particle is therefore:

- Displaced away from its initial position according to the Flow Noise value that has propagated up the profile.

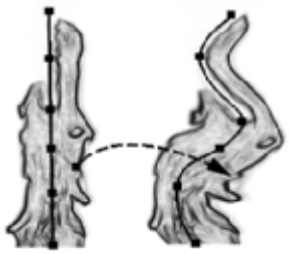

Figure 4: A cylindrical coordinate system is used to map point samples from the profile space to the space of the deformed structural curve.

- Transformed into structural curve space according to the straightforward mapping shown in Figure 4.
- Displaced a second time using a vector field generated from a Kolmogorov spectrum.

After these transformations, the particles are tested against the transformed profiles of their parent flame's neighbors. If a particle is inside a neighboring flame and outside the region defined by Eq (4), then that particle is not rendered at all. This gives the appearance that individual flames can merge. The Kolmogorov field and global wind field $w(\mathbf{x}_p,t)$ ensure that merged flames behave in a locally similar fashion even though there is nothing in the explicit model to account for merging. The particles are then in their final positions, ready for rendering.

5. RENDERING

5.1 Particle Color

Apparent flame color depends on the types of fuel and oxidizer being consumed together with the temperature of the combustion zone (hotter regions are bluer). Instead of trying to calculate the color of the flames that we want to model, we find a reference photograph and map the picture onto the two-dimensional profile used for flame shape (Figure 3). Particles take their base color directly from this mapping. Obviously, any image could be used, giving complete control over the color of the flame. This mapping does not determine the intensity of light from the particle, just its base color.

5.2 Incandescence

The flames transmit energy towards the camera and into the environment. We assume that the visible light transmitted is proportional to the heat energy given out by the flame. This is approximated by

$$E = \frac{1}{2} k \quad 0.3m \ H_c \frac{2A_f r^3}{h} \tag{5}$$

where $m = |\mathbf{v}_c| r^2$ is the upward mass flow, H_c the heat of combustion of the fuel (e.g., 15 $MJkg^{-1}$ for wood, 48 $MJkg^{-1}$ for gasoline), A_f is the surface area of the fuel (M^2), r is the radius, and k is a scale factor for visual control.

Each particle then has its incandescence at the camera calculated relative to

$$E_i = \frac{d(\mathbf{x}_i)}{n4 \ \left((h/2)^2 + l^2 \right)} \tag{6}$$

where l is the distance from the center of the flame to the camera, and n is the total number of particle samples. The k, in Eq (5) can be automatically adjusted to compensate for the energy loss due to the $d(\mathbf{x}_i)$ term in Eq (6).

Equation (5) also approximates the total energy given out to the environment. For global illumination purposes, an emitting sphere at the center of each flame segment with a radius of h/3, gives reasonable looking lighting.

5.3 Image Creation

Ideally, the deformed flames should be volume rendered directly. If the potential field approach is used, relatively fast methods are available to do this [Brodie and Wood 2001; Rushmeier et al. 1995]. For the system described however, volume rendering requires density samples to be projected through two inverse transformations to hit the density function, Eq (4). This is elegant, but limits us to using noise functions that can be integrated quickly. Instead, we note that due to the coherence in the global and local noise fields, sample points that are close together will tend to remain close together under transformation. Therefore, if we sample the untransformed profile with sufficient density to eliminate aliasing, then the transformed sample points should also not exhibit aliasing. The transforms themselves are inexpensive per particle and so total particle count isn't a huge factor in terms of efficiency. We calculate the approximate cross-sectional area of the flame as it would appear from the camera, then super-sample the profile with sufficient density for complete coverage.

The images in this paper were created using a sampling rate of around ten particles per pixel. This gives a range of between around a thousand and fifty thousand particles per flame. The intensity of each particle is calculated using Eq (6) so the total energy of the flame remains constant. The opacity associated with each particle is more difficult to calculate however. Real flames are highly transparent. They are so bright in relation to their surrounding objects that only extremely bright objects are visible through them. Cinema screens and video monitors cannot achieve contrast levels high enough to give that effect unless the background is relatively dark.

To our knowledge there's no engineering (or photographic) model of apparent opacity in incandescent fluids. So, with no justification other than observation we calculate the opacity of a particle as being proportional to the relative brightness between it and what is behind it. This is by no means a physically correct approach but it appears to work well in practice. The particles are then motion blurred using the instantaneous velocities given by Eq (1).

6. PROCEDURAL CONTROL

Separating the fire system into a series of independent components was driven by the desire to have complete control over the look and behavior of flame animations. Animators have direct control over the basic shape and color of the flames (Sections 4.4 and 5.1), as well as the scale, flickering, and separation behavior (Section 4.3). In addition, global motion is dictated by the summation of different kinds of wind field. Wind fields, whether procedural [Rudolf and Raczkowski 2000; Yoshida and Nishita 2000] or simulated [Miyazaki et al. 2001] are well understood and effective as a control mechanism. The open nature of the system easily allows for many other procedural controls, three examples of which follow.

6.1 Object Interaction

Interaction between the structural flame elements and a stationary object requires a wind field that flows around a representation of that object. We approximate the object as a series of solid voxels and simulate the flow of hot gas around it using the method based on the Navier-Stokes equations described in [Foster and Metaxas 1997]. This wind field is used in Eq (1) to convect the structural control points. The only difference is that after their positions are modified they are tested against the object volume and moved outside if there's a conflict. Any ambiguity in the direction the point should be moved is resolved using the adjacency of the points themselves.

The sample particles used for rendering are more difficult to deal with in a realistic way. Generally, the wind field naturally carries the flame away from the object, but in cases where the transformed flame profile remains partly inside we simply make a depth test against the object during rendering and ignore the internal particles. This makes the energy (Eq (6)) of flame segments inaccurate close to an object. However, the object itself is usually brightly lit by nearby flames and this is not noticeable. Figure 7 shows a rendered sequence of images involving interaction between flames and simple objects. The wind field environment had a resolution of 40x40x40 cells, which proved to be sufficient to achieve flow that avoids the objects in a believable way.

6.2 Flame Spread

Many factors influence how flames spread over an object, or jump between objects. Spread rate models rapidly become complex to take into account fuel and oxidizer concentrations, properties of the combusting material, atmospheric conditions, angle of attack and so on. For simplified behavior, we use the procedural model from [Perry and Picard 1994]. The velocity of the flame front can be described by,

$$\mathbf{v}_s = \frac{T_f}{L}\left(\quad + h \sin \quad \right) \tag{7}$$

where is the relative orientation between the flame and the unburned surface, h is the height of the flame, T_f is the temperature of the flame, L is the thermal thickness of the burning material, and is the thermal conductivity of air. Values of and L for different materials are available from tables but it is straightforward to adjust them to get a desired speed.

For complete animation control, we use a gray-scale image to determine precisely when flames become active on a surface. The image is mapped onto the surface of the burning object, and the value of the image samples correspond directly with flame activation (or spreading) times. Similar maps control both the length of time each flame burns as well as its overall intensity, diameter and height.

6.3 Smoke Generation

Two measures define the capability of a flame to produce smoke. The first is the "smoke point". This is the minimum laminar flame height at which smoke first escapes from the flame tip. The second is "smoke yield". This is a measure of the volume of smoke produced and it correlates closely with the radiation emitted from a diffusion flame. We procedurally generate smoke as particles above the tip of each flame. The smoke point above the tip is given simply by h, and the density of particles (or density associated with each individual particle) is the radiation intensity (Eq (5))

multiplied by an arbitrary scale factor. Once generated, the smoke is introduced into a gas simulator [Foster and Metaxas 1997] with an initial velocity of $\mathbf{Vp}_{n-1}$ (from the upper control point) and a temperature set to maintain the upward buoyancy force specified by Eq (2).

7. RESULTS

All the images shown in this paper were generated using the system described. Figure 1 shows a sequence of four images from an animation of a lit torch being swung through the air. The flames are generated as if the torch is standing still, but with an initial base velocity as described in Section 4.2. The control points are influenced by thermal buoyancy and Kolmogorov noise. There are 5 flame structures, each rendered using 9000 sample particles. The animation as a whole took 2.7 seconds per frame on a Pentium III 700MHz processor. That broke down as 0.2 seconds for the dynamic simulation, 0.5 seconds for rendering and 2.0 seconds for B-Spline fitting and particle sorting.

A more stylized example is the dragon breath animation shown in Figure 5. Here, flames are generated at the dragon's mouth with an initial velocity and long lifespan. These flames split multiple times using the technique outlined in Section 4.3. Again, five flames are used initially, generating over eighty freely moving structures by the end of the sequence. There are a total of 1.5 million sample particles distributed according to the size of the flames relative to the camera. Dynamic simulation took 3 seconds, rendering 30 seconds, and B-Spline fitting 1 minute per frame. The possibility of individual flames merging (Section 4.5) was disabled for artistic preference. A pre-calculated wind field based on random forces gives some good internal rotation to the fireball but overall we feel there's still a low level of turbulence missing from the animation. Results could be improved by tracking the sample particles over time through the Kolmogorov noise, although this would incur more cost during rendering to prevent undersampling.

Figures 6 and 7 show a selection of images from the large format version of the movie Shrek and interaction between flames and stationary objects respectively. The animation system was used for all the fire required by that production. Each component of the system has between three and six parameters all related to visual behavior (flicker rate and average lifespan for example). While the system actually has more parameters than a corresponding direct numerical simulation, they are fairly intuitive, and mutually independent. In our experience, aside from learning the simulation tools used for wind field creation, an animator can be productive with the system within a week of first using it.

8. CONCLUSION

We have presented a system for modeling diffusion flames. The main focus has been on efficiency, fast animation turnaround, and complete control over visual appearance and behavior. With these goals in mind, the system has been built to take advantage of recent advances in direct simulation, as well as proven techniques for generating and controlling wind fields. These methods are combined with a novel approach for representing the structure of a flame and a procedural animation methodology, to produce a comprehensive animation tool for high volume throughput that can dial between realistic and stylistic results.

9. REFERENCES

BEAUDOIN, P. AND PAQUET, S. 2001. Realistic and Controllable Fire Simulation. In *Proceedings of Graphics Interface 2001*, 159-166.

BRODIE, K. and WOOD, J. 2001. Recent Advances in Volume Visualization, *Computer Graphics Forum 20* , 2, 125-148.

CHIBA, N., OHKAWA, S., MURAOKA, K., MIURA, M. 1994. Two-dimensional Visual Simulation of Flames, Smoke and the Spread of Fire, *Journal of Visualization and Computer Animation 5*, 1, 37-54.

DRYSDALE, D. 1998. *An Introduction to Fire Dynamics (2nd Ed.)*, John Wiley and Sons, ISBN 0 471 97290 8.

FOSTER, N. AND METAXAS, D. 1997. Modeling the Motion of a Hot, Turbulent Gas, In *Proceedings of ACM SIGGRAPH 1997*, Annual Conference Series, ACM, 181-188.

FOSTER, N. AND METAXAS, D. 1996. *Realistic Animation of Liquids, Graphical Models and Image Processing 58*, 5, 471-483.

FOSTER, N. AND FEDKIW, R. 2001. Practical Animation of Liquids, In *Proceedings of ACM SIGGRAPH 2001*, Annual Conference Series, ACM, 23-30.

FEDKIW, R., STAM, J., JENSEN, H. W. 2001. Visual Simulation of Smoke, In *Proceedings of ACM SIGGRAPH 2001*, Annual Conference Series, ACM, 15-22.

INAKGE, M. 1990. A Simple Model of Flames. In *Proceedings of Computer Graphics International 1990*, Springer-Verlag, 71-81.

NISHITA, T. AND DOBASHI, Y. 2001. Modeling and Rendering of Various Natural Phenomena Consisting of Particles, *In Proceedings of Computer Graphics International 2001*, 149-156.

MIYAZAKI, R., YOSHIDA, S., DOBASHI, Y., NISHITA, T. 2001. A Method for Modeling Clouds based on Atmospheric Fluid Dynamics, *In Proceedings of the 9th Pacific Conference*, 363-372.

PERLIN, K. AND NEYRET, F. 2001. Flow Noise, *ACM SIGGRAPH Technical Sketches and Applications*, 187.

PERRY, C. AND PICARD, R. 1994. Synthesizing Flames and their Spreading, In *Proceedings of 5th Eurographics Workshop on Animation and Simulation*, 56-66.

REEVES, W. T. 1983. Particle Systems –A Technique for Modeling a Class of Fuzzy Objects, *ACM Transactions on Graphics 2*, 2, 91-108.

RUDOLF, M. AND RACZKOWSKI, J. 2000. Modeling the Motion of Dense Smoke in the Wind Field, *Computer Graphics Forum 19*, 3.

RUSHMEIER, H., HAMINS, A., CHOI, M. Y. 1995. Volume Rendering of Pool Fire Data, *IEEE Computer Graphics & Applications 15*, 4, 62-67.

STAM, J. 1999. Stable Fluids, In *Proceedings of ACM SIGGRAPH 1999*, Annual Conference Series, ACM, 121-128.

STAM, J. AND FIUME, E. 1995. Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes, In *Proceedings of ACM SIGGRAPH 1995*, Annual Conference Series, ACM, 125-136.

STAM, J. AND FIUME, E. 1993. Turbulent Wind Fields for Gaseous Phenomena, In *proceedings of ACM SIGGRAPH 1993*, Annual Conference Series, ACM, 369-376.

YNGVE, G., O'BRIEN, J., HODGINS, J. 2000. Animating Explosions, In *Proceedings of ACM SIGGRAPH 2000*, Annual Conference Series, ACM, 29-36.

YOSHIDA, S. AND NISHITA, T. 2000. Modeling of Smoke Flow Taking Obstacles into Account, In *Proceedings of 8th Pacific Conference on Computer Graphics and Applications*, 135-145.

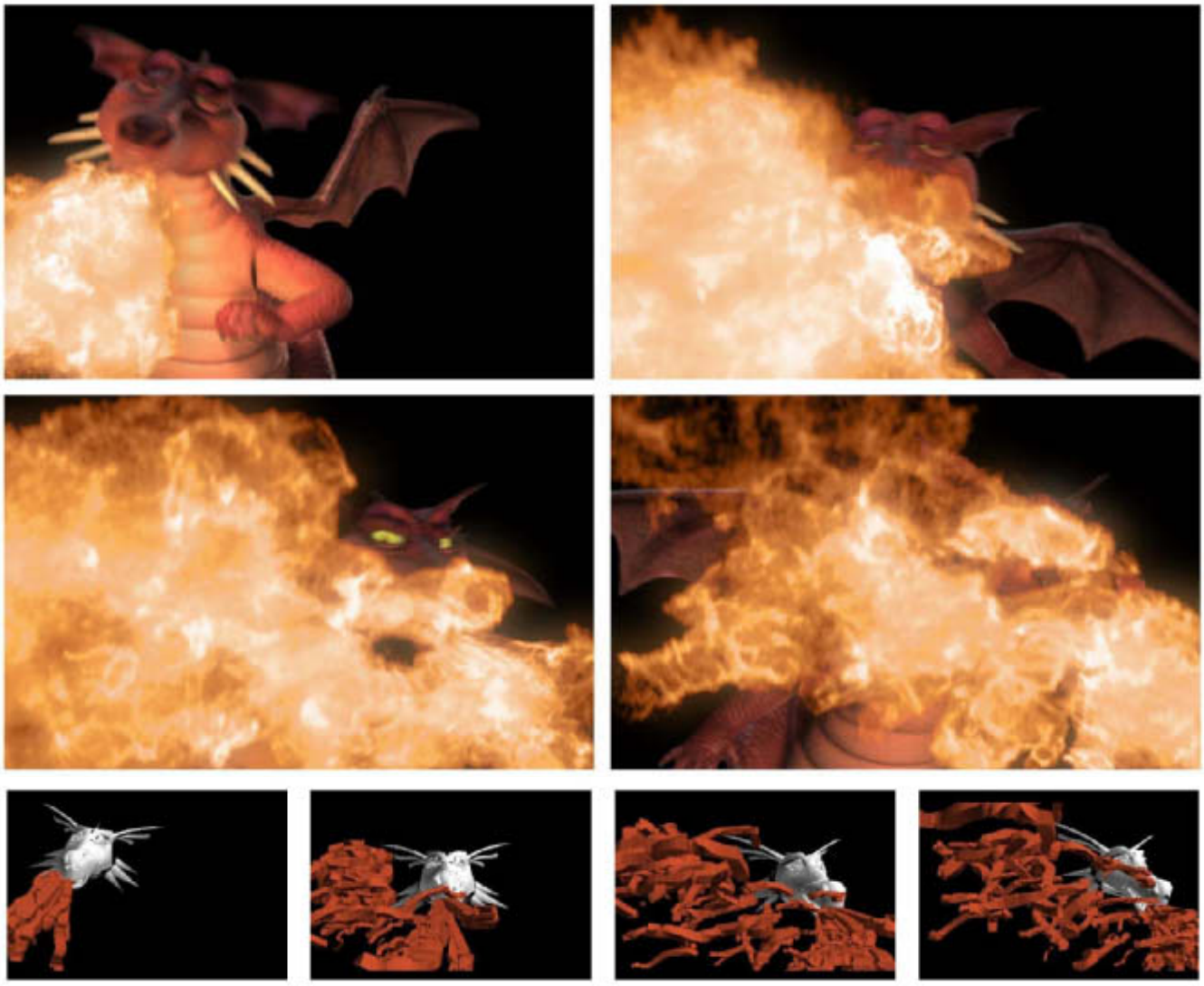

Figure 5: A sequence of images showing fully rendered stylized dragon's breath and the structural curves used to model it.

Figure 6: The fire system in use on a 3D animated feature film.

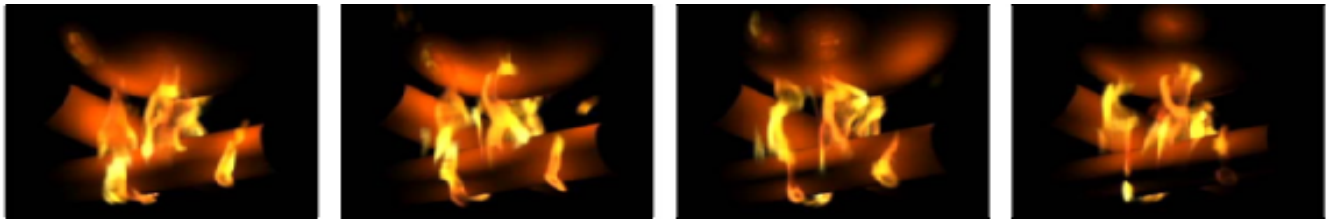

Figure 7: Interaction between flames and simple stationary objects. A wind field simulation resolution of 40x40x40 was used.