Synthesizing Interactive Fires

by
Christopher Harton Perry

B.A., Physics and Astronomy
Amherst College, Amherst MA
May 1992

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES
at the
Massachusetts Institute of Technology
June 1994

Signature of Author ________________________________
Program in Media Arts and Sciences
6 May 1994

Certified by ________________________________
Rosalind W. Picard
Assistant Professor of Media Technology
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ________________________________
Stephen A. Benton
Chairperson
Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Synthesizing Interactive Fires

by
Christopher Harton Perry

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on May 6, 1994
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

Abstract

This thesis presents a model for the computer graphic synthesis of interactive fires. The term *interactive* is used to highlight two features of the proposed model absent from previous fire models: first, the fires *interact* with their environment like real fires by spreading over, charring, and lighting the objects which dictate the shapes and colors of the flames themselves. Second, useful forms of control are provided without a severe computational load to allow the user to *interact* with the model in the shaping of the final synthesis.

Four constraints on model selection are identified in hopes of insuring these two forms of interactivity: the model must be *convincing, controllable, contemporary*, and *computable. Convincing* means that the synthetic fires visually mimic real fires in shape, color, and motion. *Controllable* means that the parameters of the synthesis are adjustable with appropriate physical, semantic, and other "knobs." *Contemporary* means that the model is based on today's graphics paradigm and requires no new modeling, animating, or rendering techniques. *Computable* means that results render at rate which allowing for real-time interaction with the model. These *four C's* create the perspective with which background work is reviewed and the actual fire model derived.

The model meets the four criteria by combining a perceptually-motivated, particle system flame model with a physically-based flame spread model. Both are presented in detail.

Videotaped examples of synthetic results are available from the author.

Thesis Supervisor: Rosalind W. Picard
Title: Assistant Professor of Media Technology, NEC Career Development Professor of Computers and Communications

This work was supported in part by Hewlett Packard and by BT (formerly British Telecom)

Synthesizing Interactive Fires

by
Christopher Harton Perry

The following people served as readers for this thesis:

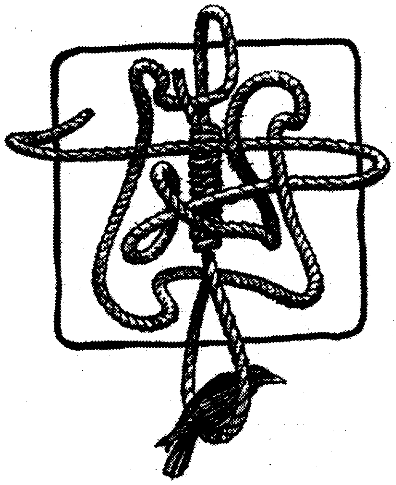
Reader: ______________________________

Neil Gershenfeld
Assistant Professor of Media Technology
Program in Media Arts and Sciences

Reader: ______________________________

Professor F. Kenton Musgrave
Assistant Professor of Engineering and Applied Sciences
Department of Electrical Engineering and Computer Science
George Washington University

Acknowledgements



Thank you, Roz for being a constant source of support, inspiration, and energy over the last two years. I will always remember the pleasure I've had developing these ideas with you.

Thank you to Ali, Stan, Irfan, Wave, Trevor, Linda, and Santina, for advice all along on matters unlimited, submission-time support, and general companionship.

Thank you, Steven, Thad, Bobby, Lee, Aaron, Boback, Alex, Judy, Fang, Tom, Steve, Laurie, Laureen, Andy, and my other compatriots at vismod for watching and commenting on the fires throughout the past year. You have been a terrific group to work with.

Finally, thank you to my family, 28 Stone, the B-dorm crew, and the MIT rugby club, for out-of-lab support, distractions, and constant entertainment.

If Thou be one whose heart the holy forms
Of young imagination have kept pure,
Stranger! henceforth be warned; and know that pride,
Howe'er disguised in its own majesty,
Is littleness; that he who feels contempt
For any living thing, hath faculties
Which he has never used; that thought with him
Is in its infancy. The man whose eye
Is ever on himself doth look on one,
The least of Nature's works, one who might move
The wise man to that scorn which wisdom holds
Unlawful, ever. O be wiser, Thou!
Instructed that true knowledge leads to love;
True dignity abides with him alone
Who, in the silent hour of inward thought,
Can still suspect, and still revere himself,
In lowliness of heart.

from "Lines Left upon a Seat in a Yew-tree," William Wordsworth, 1795 *for Margaret Perry*

and Milton Godshalk

Contents

Chapter 1

Introduction

Fire is everywhere, appearing to us on a range of scales from sparks and candles to forest fires and explosions. It is one of Nature's greatest actors, being at one time powerful, mesmerizing, dangerous, comforting, illuminating and destructive.

Fire appears throughout movies and television, often resulting in expensive construction and subsequent destruction. Certain exotic flames and explosions can be created only under specialized conditions and these can be hard if not impossible to repeat. Ultimately, the laws of physics limit the range of available effects. If it were possible to generate fires without actual combustion and control them without difficulty, the benefits and savings would be great.

This thesis journals an attempt at the visual synthesis of fire on a computer, and this chapter aims to place this challenge in multiple contexts.

1.1 general target: the *four C's*

Synthesizing natural phenomena with computer graphics techniques has been a topic of interest due to recent convergent trends in digital computing and, primarily, the entertainment industry. The various creators of content are imagining "natural" behaviours too difficult, expensive, or impractical to image with ordinary means, and computers with sufficient power are being developed to generate these complex images.

Because of this, models of natural phenomena that are "realistic image" generators *only* may enjoy scattered successes, but through lack of practicality they will fall by the wayside. The truly successful synthesizers of natural phenomena must also be usable by the creative people who need them.

These considerations support models for synthesizing natural phenomena that meet four criteria (the *four C's*). The first of these constraints on new models has been prevalent since the begining of computer graphics; the latter three arise from the dependency on and desire for interactive, human-driven synthesis. Models must be:

Convincing: The visual output of the model must of course mimic the visual nature of the real phenomena. Although there are some markets for non-realistic synthetic imagery (cartoons, etc.), the major drive is towards photorealistic images that are hard to discern from real ones.

Controllable: As much control as possible over the colors, shapes, and motions of the synthetic phenomena should be given to the user, preferably in the form of semantically-labeled "knobs" for ease of use. This criterion aims to prevent models that cover tiny subsets of phenomena while at the same time it supports the human factor interacting with the syntheses.

Contemporary: The model should be defined within the existing paradigm of computer graphics so that the systems used sucessfully for modeling, rendering, and animating traditional objects can also include the new model. This criterion is meant to drive research in the direction of a unified set of complete-scene modeling tools rather than a disconnected set of tools, one for every aspect of a scene.

Computable: The model should run in interactive times so that the large parameter space spanned by the provided controls can be examined. Although hardware breakthroughs will consistently drive computability up, present-day controllability and thus model practicality would be naught if this criterion is not met.

1.2 the *four C's* for fire

This thesis proposes a model for fire synthesis that meets the four C's.

In terms of fire, *convincing* means not only in shape, color, and motion but in overall behaviour. Fires are more than just *flames*: a flame is the light-emitting part of a fire, namely, the glowing part of a candle or the familiar wisps of a bonfire. Real fires are the interaction of three-dimensional flames with the environment: they spread over surfaces, release smoke and heat, grow, emit light, consume fuel, and die. The proposed model is designed to realize these interactions, many of which are demonstrated here[1].

Controllable insures that to be as useful as possible, the shape and the behavior of the fires should be manipulable on a variety of scales. External fields like wind and gravity, for example, have a profound effect on the way fires burn in the real world and must be included. Different objects burn with different rates and colors, and the ability to define

[1] It is not reasonable within the scope of a Master's thesis to demonstrate them all.

and alter the characteristics of materials should be granted. Someone may even want to anthropomorphize a fire and should be able to control the model to a useful degree on levels at which that can be done.

Specifically, in addition to low-level control over individual flame positions and shapes, the following adjustable physical properties have been written into the fire model presented here. Note that these non-orthogonal parameters can be interactively changed with the fires reacting accordingly:

- gravity
- wind
- material flammability
- material geometry (shape and scale)
- illumination
- fire spread
- charring properties

A *contemporary* fire model must exist in a world where light sources and objects already have efficient and functional representations, namely, in the existing paradigm of computer graphics. This means that, for example, the surfaces fires adhere to must be selected from the list of those already in use, namely, polygons, parameterized surfaces, mathematical primitives, etc. Because flames might be considered both light sources and objects in a scene, they must be able, ideally, to fit into both rendering and modeling with no change required in the standard graphics model. The proposed model renders flames with Gouraud-shaded, alpha-blended triangles and spreads them over polygonally-defined solid fuels. Simple point light sources can be driven by the flames for the lighting of scenes.

Computability means that the fire model must provide results in interactive times. The proposed model can render complete fires at approximately 1 Hz, and can achieve faster rates for low-resolution parameter browsing by shutting off particularly expensive detail calculations.

1.3 quantifying success

Determining whether the model is *controllable*, *contemporary*, and *computable* (thus meeting three of the *four C's*) is a relatively easy task. "Knobs" can be counted, graphical elements either exist in one's current graphics system or they don't, and frame rates are easy to measure.

However, the success of any rendered images as *convincing* examples of fire is difficult to measure quantitatively. No metrics have been demonstrated that allow a computer to compare two image sequences and measure similarity in the way a human does[2], plus, comprehensive human visual tests are both expensive and time consuming. As a result, the success of the proposed fire model as a *convincing* one will be evaluated in terms of more qualitative comparisons to real fires, such as what behavioural characteristics are shared between the two and what people exposed to the examples think of them.

1.4 fires defined

Fires arise due to the interaction between a self-supporting exothermic chemical reaction and physical transport processes. An energy-releasing reaction by itself is not enough: a match will ignite but will not burn without a continuous flow of oxygen. On an orbiting spaceship, for example, a match must be continuously moved for steady-state burning to occur since convection currents don't form without gravity[8].

The individual mechanisms occurring in fire (such as chemical reaction, molecular diffusion, and heat conduction) are well understood on their own, but no unifying scientific description emerges that can help manage fire as a single entity. This is most certainly an issue of scope: as Faraday said in the begining of a six-lecture treatise on the candle[3],

> There is not a law under which any part of this universe is governed which does not come into play and is touched upon in these [burning] phenomena.

For example, one who studies the energy release efficiency of various combustible materials will work on the chemical and molecular level, while one interested in preventing forest fires will be concerned with forest-size spreading rates. In each of these cases, the mechanisms

[2] Although the model proposed in this thesis has several parameters which could be fit to live video for comparison and recognition; these will be noted as they are introduced. Further study would be needed to claim that these parameters are sufficient for visual similarity.

inherent in real fires which are not on the scale being studied are extraneous and, moreover, *in the way* of acquiring complete knowledge at the desired scale.

The two largest groups of researchers studying fire are combustion scientists and flame spread scientists, working at such greatly different scales that there is almost no ground for collaboration between them. The former typically investigate new materials and their energy release as fuels. The latter concern themselves with where the burning regions are moving, and study the physical mechanisms of heat transfer from one region to another.

Fire is a different phenomenon to each of these groups because they study it with different goals. Combustion scientists want high levels of predictable performance from a given substance, while spread modelers hope to prevent disasters.

One difficulty in synthesizing fire is in combining these two, namely, treating fire as a visual phenomenon: the fires that people see. This both eases and hinders the application of existing scientific knowledge to the problem at hand. For instance, the limits of human visual acuity argue (advantageously, in terms of deriving a simple model) against requiring molecular or temperature representations since people cannot without aid see heat or individual molecules. However, the fires people see range from the well-behaved candle flame to the roaring bonfire, so the representation must be able to describe these extremes and everything in between. This kind of general model has yet to appear in the scientific community.

Allowing such a broad range of acceptable fires might be interpreted as making believable synthesis easier (try shooting at the side of a barn), but true success demands covering most of the acceptable range and not just one isolated instant. For example, if one could master the simulation of wave fronts when a pebble enters the water, but had no handle on the steady state of the water's surface, one's simulations would be very limited. Since no obvious interpolation scheme between different believable solutions exists, the space of acceptable fire colors and motions is analogous to a mine field: there are countless safe places to stand but to get from one to another without blowing up is the real goal.

1.4.1 flame types: diffusion and pre-mixed

Briefly, there are two types of flames that exist: *diffusion* flames and *pre-mixed* flames. Canonical examples of each are, for the former, standard candle flames, and for the latter, Bunsen burner flames[3]. The two types are distinguished by the fact that in diffusion flames, physical diffusion of oxidant and fuel is what limits the reaction rate, while in pre-mixed flames, it is the rates of the chemical reactions themselves that limit the total reaction rate[6][1]. Visually, they are quite different (see the color plates in [6] for some examples).

Although pre-mixed flames make their way into the laboratory and jet engines, diffusion flames are what people see most of the time, be they on the candlestick, in the fireplace, or in some disaster on the news. The model proposed in this thesis is designed for diffusion flames.

1.5 motivations

Fire is an integral actor in the arts and entertainment industries, but since synthetic fires have for the most part been too unwieldy or unrealistic, pyrotechnic and hand-drawn methods are the only ways to efficiently bring fire into a production. Without a form of "stunt" fire, the real stuff, like certain human acting talent, tends to be expensive, limited in repertoire, and demanding of much attention.

1.5.1 financial

The total cost per day for a very modest pyrotechnics film shoot is estimated at \$5,520[11], based on a \$1,000 fee for pyrotechnics alone. More advanced substrates, including destroyed vehicles and buildings, run into and beyond the tens of thousands of dollars, not to mention any immeasurable damage that could come to humans. Recent filming in Boston for the film "Blown Away" involved explosively destroying automobiles and a ship; excessive damage fees were incurred when dozens of neigborhood windows were shattered by the force of the shock wave. More difficult effects, like explosions or fires with "behavior," can cost even more.

Some fires and explosions can be filmed in miniature to reduce the extent of the destruction. However, expensive post-production processes (usually by hand) can be necessary to make the tiny fires composite properly with the full-size shots.

These considerations argue that realism and computational efficiency are required for synthetic fires to be financially desirable. To minimize post-production costs, flames must

[3] The standard laboratory burner was invented by Bunsen around 1855.

not only look realistic, they should light a scene properly and automatically alter models as they burn. Synthetic fires must also render in close to interactive times so that extra takes generate little extra cost.

Likewise, if paid employees have to spend inordinate amounts of time understanding and using a fire synthesis system it will not find its way into the budget. This supports a model that can be inserted directly into a scene with these other computer-generated objects and burn realistically with little user intervention. Advantageous control knobs would be ones that allow the characteristics of the fuel and environment to be specified, but without burdensome amounts of detail. The examples mentioned above of physical forms of control, like wind, gravity, object flammability, could perhaps make drastic, yet believable, changes in every flame's shape without need to address every flame individually.

1.5.2 artistic, creative

Even with the gigantic budgets of Hollywood, not everything can be purchased. Spaceships, alien creatures, and dinosaurs are three examples which only exist in photographic media today because visual approximations can be made to look real. Until the advent of computer graphics, these effects were limited to be those which could be built by a modeler or somehow put manually into a piece of film. Models on a computer, however, don't need to be physically sound, they don't require expensive materials to build, and they don't break or take up space. If you can think of it, chances are it can be built on a computer.

Beyond just providing a stunt double for real fire, a major advantage of having synthetic flames would be that one could create fires which don't adhere to physical laws. Certain artistic or hyper-real endeavours, like attaching fire to people, simply cannot be done without obtrusive protective suits. Likewise, anthropomorphic fire which moves and "behaves" would have no place in the real world but might be needed for a particular story.

1.5.3 theoretical

The failure of the traditional sciences to unify a treatment of fire provides a theoretical incentive to work towards fire synthesis. Physics aims to tell us about the world, but it hasn't succeeded at giving a visual level of description for fire. Computer simulations of natural phenomena run into this situation all the time: one is given the science from a textbook, but what is really desired is the *perceptually relevant* portion of the science.

Discovering the laws of perceptual physics, perceptual chemistry, or perceptual biology is an intriguing academic affair, opened up recently because of the needs for synthetic natural phenomena.

1.5.4 personal, spiritual

The Empedoclean quartet of earth, fire, air, and water have held up over the centuries as being a minimum and complete representation of the elements of Nature as perceived by common man. Research in computer graphics has, in the author's opinion, produced stunning, realistic animations of each of these except for fire. Thus the "Nature Toolkit" is only three-quarters done - it is a quest to begin to complete the set.

Chapter 2

Background

Fire synthesis is not a new topic; there is previous work which can be studied and applied. This chapter reviews prior work on flame and fire synthesis in terms of the *four C's*, with emphasis on two categories of techniques: particles and noise. Two approaches to creating new models, numerical and ontological, are also reviewed.

2.1 computer graphics techniques

Since its ultimate goal is the rendered image, perhaps the most relevant research into synthesizing fires as visual phenomena comes from computer graphics. The most well-known methods fall in the two categories of particle systems and noise-based techniques.

2.1.1 particle systems

Particle systems are a tool used for rendering and animating phenomena which can be abstracted into the combined behaviors of many individual elements [21]. These elements (*particles*) are given certain visual characteristics and initial conditions, and then change through time governed by user-defined laws.

The first particle-based animated fire sequence to achieve major recognition was the work of Reeves in 1983 for the movie Star Trek II: The Wrath of Khan (the "Genesis Sequence"). He used two particle systems to describe the propagation of a fire over a spherical planet. One of these systems controlled the spreading of the flames over the surface of the planet, and the other formed the flames themselves[21].

The Genesis Sequence exhibits both the successes and the flaws of the standard particle

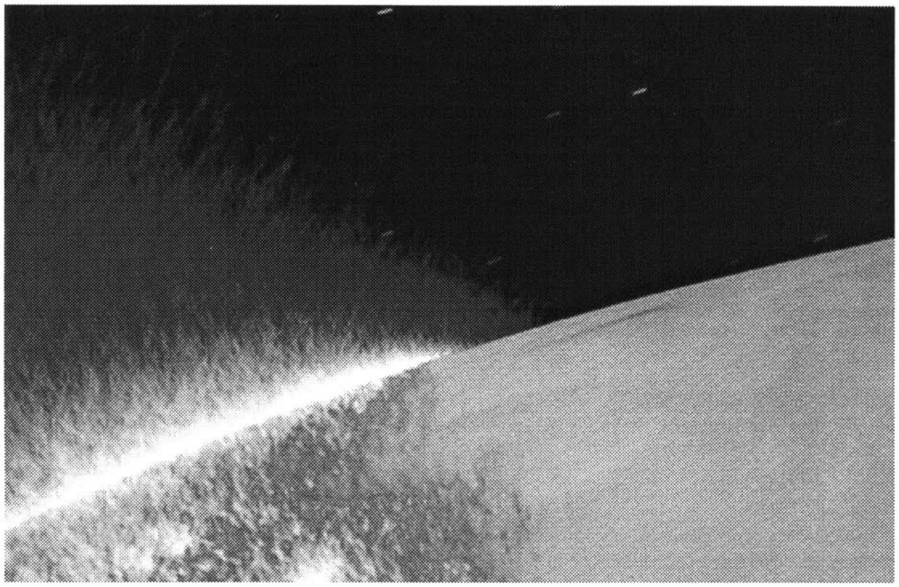


Figure 2-1: A frame from the Genesis Sequence showing a wall of flame synthesized with a particle system. From [21].

system technique. The engulfing of the planet's surface and the motions of the fire as a whole were very convincing, but the pointilistic nature of the flames was clear in the final result and thus it appeared to be some sort of imaginary fire (see Figure 2-1).

The particle system technique of synthesizing fire was also used by Sims[25]. He attempted to replicate some of the more complicated behaviors of real flames: the attraction to the conflagrant object when close to the object, the different temperatures and therefore colors of different regions in a fire, and the flickering of individual flames. Each of these enhancements yielded a better synthesis, and the net result (seen in Sims's "Burning Logo") is much more flamelike than Reeves' original work. However, the particulate nature of the method still shows itself, especially on a single paused frame of the animation where the fire looks more like autumn leaves than like fluid flames (see Figure 2-2). Nonetheless, his work represents the best particulate fire work seen.

In general, the particle system technique has two major advantages. The first is speed (*computability*): traditional particle systems treat each particle independently of the others, yielding an $O(N)$ computational cost where N is the number of particles. The second is that particle systems are easily placed into a scene because each particle is given a world space location and velocity. This makes their response to and their effect on other objects easy to calculate. In terms of *controllability*, when winds or gravity - very important in

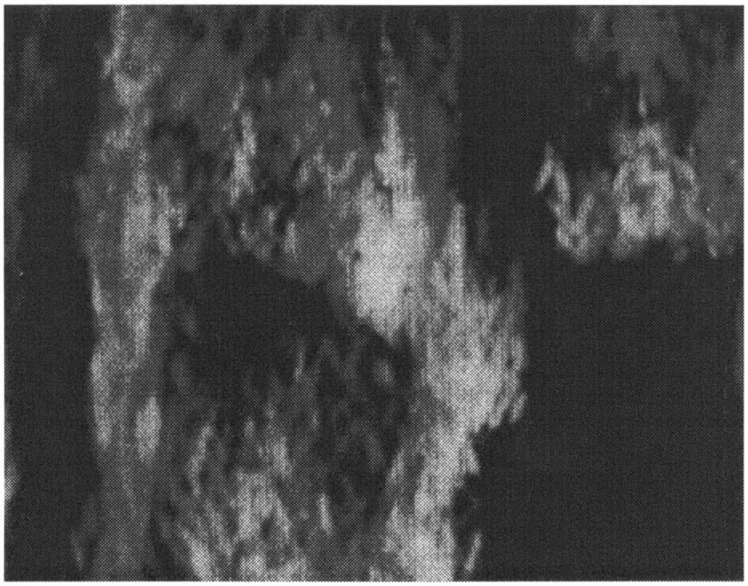


Figure 2-2: A frame from Sims's *Burning Logo* sequence. Note the disconnected nature of the flames. Any vertical bands in the image are a result of poor reproduction. From [25].

the shaping of real flames - are to be added, world-space representations become even more desirable.

The major disadvantage of the particle system technique is the pointilistic, discontinuous nature of the final rendered images (seen in Figures 2-1 and 2-2). This happens because individual particles are traditionally rendered as points or tiny spheres, and since efficiency requires neglecting particle-to-particle interaction, certain particles end up stranded from any cohesive group. For certain "fires" such as sparklers or fireworks, this might be a desired effect, but for common continuous flames, the appearance is not *convincing.*

2.1.2 noise-based techniques

Another computer graphics method used for animating fire is the noise-based solid texturing approach introduced by Perlin[18]. In this technique, band-limited noise and a set of shaping functions are used to vary the color, height, transparency and other characteristics of an object to be rendered.

An especially useful form of the noise is a fractal function formed by summing the absolute value of the original noise at different scales over several octaves. The resultant form has an approximate $1/f$ power spectrum, generalizable to $1/f^{\beta}$, and is somewhat arbitrarily referred to as "turbulence" in computer graphics. As described by Voss[28], this

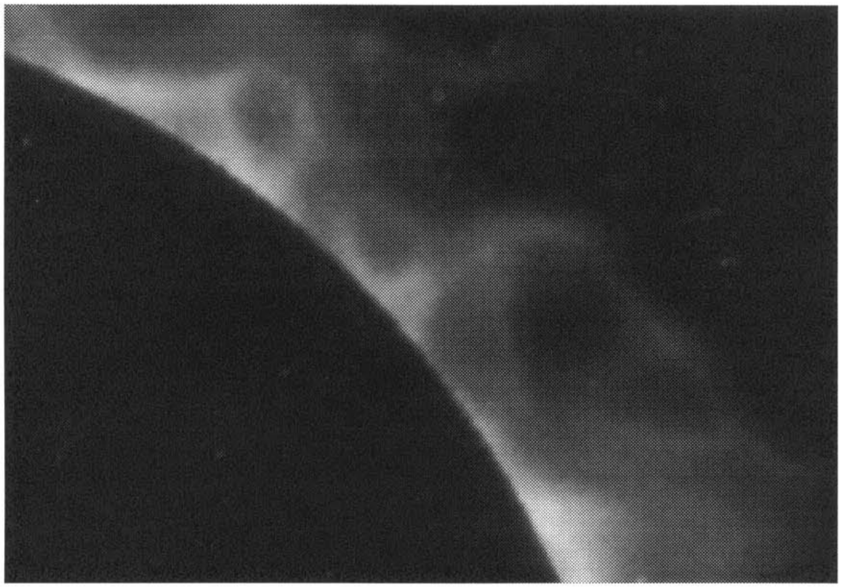


Figure 2-3: A solar corona synthesized with noise-based techniques. From [18].

$1/f$ behavior is manifest in many natural systems.

Perlin synthesized fire by combining a simple treatment of the physics of illumination from flames with the approximation of turbulence[18][19]. He began with the assumption that the heat and, therefore, color of the fire will decrease with distance from the source. He then created a function that mapped distances to colors in such a way that large distances would return black and smaller distances would correspond to a combination of red and yellow. To determine the color at a lattice site, he sent this function the distance to the point in question plus the value of the turbulence at that point. See Figure 2-3 for an example.

Sakas and Westermann[23] enhanced Perlin's "rescale and add" method capable of forming $1/f^\beta$ turbulence and altered it to more closely approximate real turbulence which they recognized was $1/f^\beta$ only at frequencies in the *inertial subrange* of the turbulence spectrum. They also put in hooks to allow for control of certain characteristics of the turbulence field, taking into account the smallest eddy sizes, a unidirectional wind, and different fractal dimensions along the various axes of the lattice. Figure 2-4 shows example flames synthesized with their technique.

Noise-based techniques were combined with some creative shaping by Peachey[17] to construct torches for the popular Listerine Knight animations. He used sinusoidal splines to shape the edges of the flame, and used the perturbed color ramp indexing method

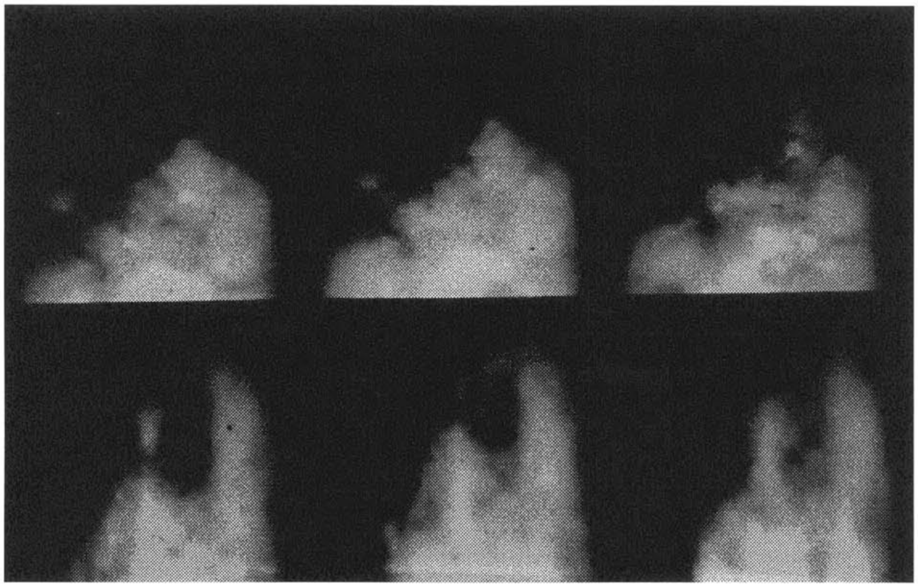


Figure 2-4: Noise-based flames from Sakas and Westermann [23] showing some different styles of flame that can be synthesized by varying the scales of the noise axes.

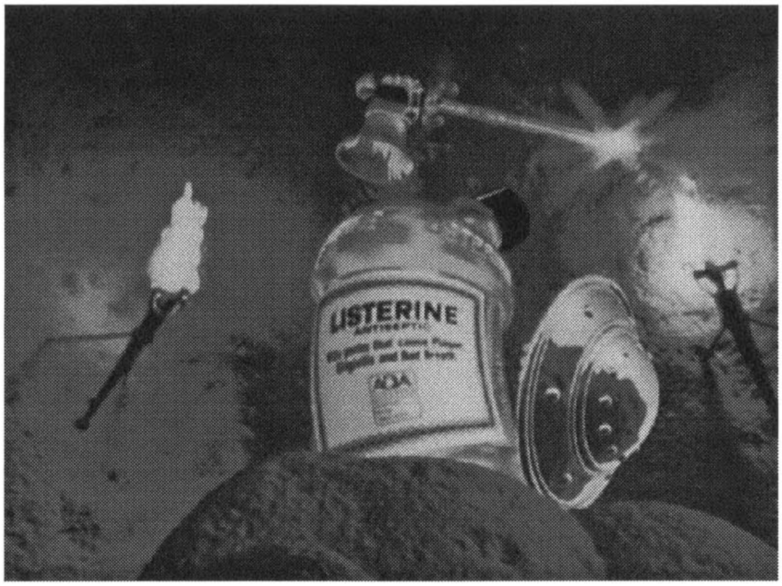


Figure 2-5: A frame from one of the Listerine Knight animations showing spline-shaped, noise-based torches. From [2].

described above to color the interior (see Figure 2-5).

In general, noise-based techniques are successful at exactly what particle systems fail at, namely, forming and maintaining spatially continuous flames. Additionally, the $1/f$ nature of the turbulence function provides extremely *convincing* motions through time.

The major disadvantages of using noise-based methods for fire synthesis are its computational inefficiency and its awkwardness to use in scenes[1]. Computation speeds are steadily increasing, but three-dimensional volume rendering presently remains time-prohibitive. Even if these types of rendering techniques become available in interactive times the second drawback would continue to exist, namely, the difficulty of using and *controlling* these "noise spaces" in scenes with other objects.

To elaborate, imagine a fire that begins as a small flame on the visible surface of an object and spreads to engulf the entire object. The region of space containing the flames would have to be mapped to a three-dimensional noise field, warped in such a way to attach the flames to the objects at their bases, but also allow them to interact in the surrounding environment. Additionally, the noise-based approach is not good at capturing the whole range of scales at which visual fire can appear: Peachey's use of shaping functions is an example of the extra effort needed to create a relatively small-scale noise-based fire.

2.1.3 other computer graphic technique(s)

Inakage [9] has taken a more physical look at the processes that form real flames. Combining a model for the emission and transmission of light in the regions near combustion with volume rendering techniques, he has rendered convincing examples of both diffusion (candle) and pre-mixed (Bunsen burner) flames. He also provides physical forms of control over the flames with adjustable model characteristics such as the ratio of oxidant to fuel, flame temperature, and velocities of both the oxidant and the fuel.

Unfortunately, this level of physically-based volume rendering is computationally prohibitive and no clear methods for animation or extending the approach to large fires have been presented. This is especially apparent in that the forms of control are useful for small flames but would not be the desired "knobs" for bigger fires, where ratios and velocities might be exchanged for more global burning parameters (see Section 3.5).

[1] Note how noise-based techniques are practically the foil of particle techniques!

2.2 numerical methods

Numerical techniques have been proved time and time again as successful ways to simulate Nature on a machine, so their potential to handle fire merits examination here. For example, a small bit of physics can have quite magical results, as Newton's second law and Hooke's law can turn an aliased, flat-shaded cube into a bouncing chunk of Jello[10].

The level of represention desired in this thesis requires both convincing flames and convincing behaviours as these flames spread over objects. Therefore, a complete numerical solution would have to yield both the intricate colors in the flames and the three-dimensional transport of flames to other parts of the environment.

Based on the research to date, such a numerical model of a fire is unfeasable. Even just two-dimensional numerical flame simulations are extremely unwieldy, taking hundreds of supercomputer hours to complete[26]. In the words of Fernandez-Pello[5],

> The formulation of a rigourous mathematical model of the flame spread process would consist of the conservation equations for the reacting gas phase coupled at the interface to the condensed phase conservation equations through the appropriate boundary conditions. This would require the solution of a system of coupled, two-dimensional, elliptic, nonlinear partial differential equations that would include variable material properties, appropriated gas phase chamical kinetics and solid phase pyrolysis mechanisms. The solution of this full problem, even after considerable simplifications, is very difficult.

2.3 the ontogenetic approach

Fire synthesis is just like other problems in synthesizing Nature: scientific models exist that try to explain the phenomenon in question, but these models aren't easily applicable to making computer generated images.

Ontogenetic[2] is a term used to describe an engineering rather than a scientific approach to modeling, where one seeks a convincing visual effect rather than accurate physical simulations. To quote Musgrave[16],

[2] From Webster's Collegiate Dictionary, 10th Ed.: "ontogenetic: ... 2: based on visible morphological characters." The term was first used to describe graphics modeling techniques by F. Kenton Musgrave.

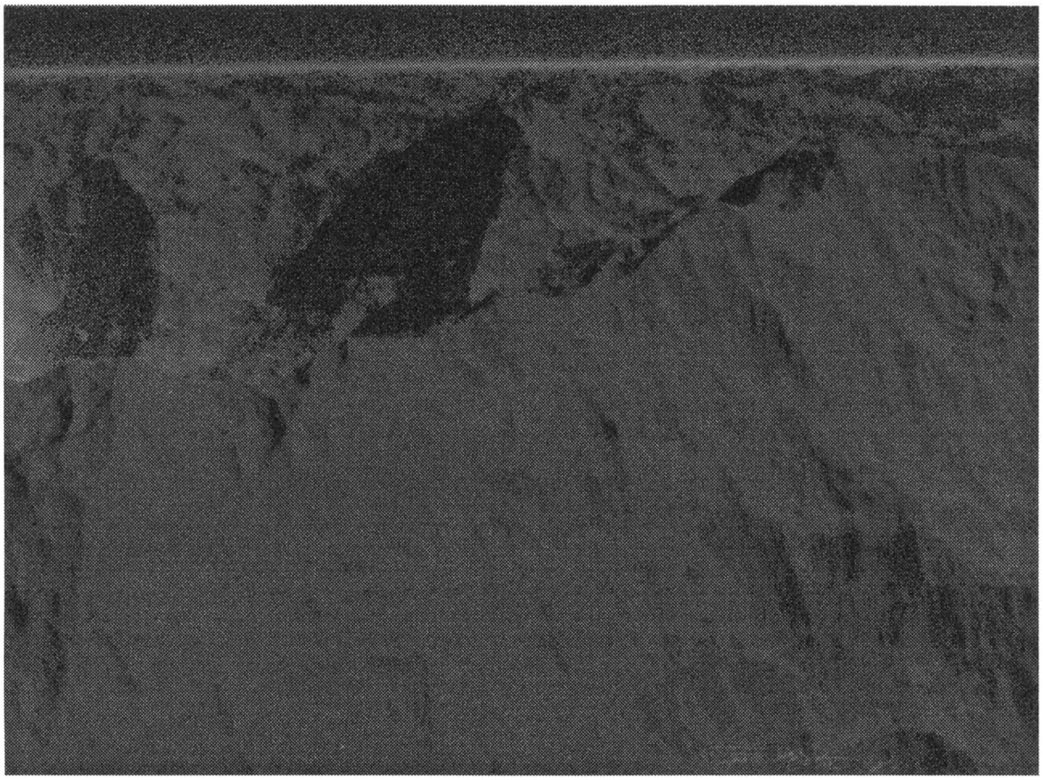


Figure 2-6: "Slick Rock" – an example of synthetic landscape imagery based on ontogenetical models. From [2].

> The underlying idea is that, in the field of image synthesis, it is a legitimate engineering strategy to construct models based on subjective morphological (or other) *resemblance*, as opposed to, for instance, precise (e.g., mathematical) veracity, as is a goal in scientific models.

The advantage of this approach is that computationally attractive models are considered *just as good* as the existing scientific models if they produce perceptually realistic results. In the realm of efficient and realistic visual fire simulation, this is exactly what we seek.

The computational intractability of the current governing equations for fire will force some form of approximation of them to be made, if they are to be used at all. The ontogenetic paradigm supports studying these equations for what their effect is visually; as other research in the synthesis of natural phenomena can show, this can be quite a successful tactic. Figure 2-6 is an example of an ontogenetically modeled mountainscape.

A recent work in simulating gaseous phenomena [27] is a good example. In this paper, turbulent wind fields are modelled as the sum of two individual fields: a deterministic large-scale field and a stochastic small-scale field. This separation allows them to compute the fields and the evolution of the densities in reasonable amounts of time. Physically, the dichotomy is unrealistic, but as their synthesis shows, this is not the case perceptually.[3]

The synthetic Jello example mentioned earlier can teach the same kind of lesson: a few governing equations can provide enough details for people to accept the artificial as realistic.

A full-blown numerical solution for fire would be applying all we've learned about the chemistry and physics of the situation to a simulation that probably doesn't require it, however, if certain parts of the scientific models provide a lot of realism at computationally effective rates they should be exploited.

[3] Unrealistic, perhaps, but inspired by physics: Reynolds's original turbulence theory does regard the turbulent stream as a superposition of two motions, one translative and one random[23].

Chapter 3

Model Derivation

Leontes Still methinks
There is an air comes from her. What fine chisel
Could ever yet cut breath? Let no man mock me,
For I will kiss her.
– **The Winter's Tale**, V:iii

This chapter discusses the hypothesis behind, and selection specifics of, a fire synthesis model that is visually convincing, contemporary, controllable, and computable, thus meeting the requirements defined in Chapter 1 for interactive fires.

3.1 the bi-partite hypothesis: flames + spread = fires

Although *fires* and *flames* have been equated in graphics, there is an understood difference between them in the physical sciences. From [14],

> A *fire* is a set of physical and chemical phenomena, which include [sic] combustion, fluid flows, and pyrolysis or evaporation. When the combustion occurs in the gas phase, the luminous part of the gas is called the *flame*.

As the above definition of fire implies, physically-based fire synthesis should consider not only the luminous, combustive gases but also the processes which transport materials to and from the combustion region (fluid flows and pyrolysis). These aspects of real fires are what lead to non-isotropic spread and a self-supporting burning process.

The particle system and noise-based approaches described in the previous chapter synthesize fires with a single flame model. Both succeed to a point at achieving the complex colors and motions apparent in real flames, but practically no investigation has been done to link these fires to the things they are burning (with the exceptions being the ad hoc techniques of [21] and [13]). Thus, they are really only *flame* models.

This thesis presents a non-physical flame model and an independent, physically-based fire spread model, and then a method of combining the two to form fires. Although the separation of combustion from flow and evaporation is inherently non-physical, the success of this thesis is based on the hypothesis that the division will work perceptually. This "bi-partite" hypothesis comes as much from a desire for simplicity (Occam's razor) as from the examples set by the spread modelling community who succeed in accurately modeling spread without treatment of combustion[30] (also [4]):

> Theoreticians often attempt to include all potentially important phenomena in their models of fire spread, believing that they cannot properly describe the process if something that contributes is neglected. Nevertheless, there is merit to the opposite view which holds that the best avenue for developing understanding is to neglect all but essential phenomena and to study thoroughly limiting cases in which different phenomena are controlling.

3.2 flame model: an enhanced particle system

The flames are modeled using a modified particle system technique. Modifications are made to combat the pointillistic problem mentioned in the previous chapter which, in the context of the goal of this thesis, is the primary recognized disadvantage with such methods. Noise-based techniques, on the other hand, remain computationally expensive, and, more prohibitively, resist facile control and placement in three-dimensions.

Flames can be rendered with fewer particles and without the discrete artifacts of [21] and [24] by adding dynamic geometries to the particles. It is important to note that this is a treatment of one of the "infinite degrees of freedom" forseen in [21] and not a totally new type of modeling primitive; thus, the term "particle" will still be used to describe the atomic elements of the flames.

3.2.1 particles with dynamic geometries

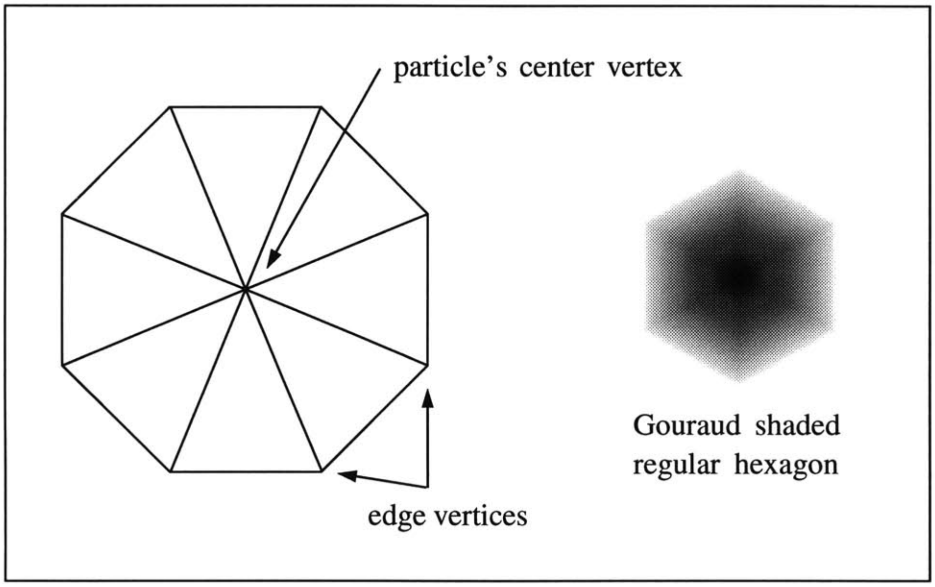


Figure 3-1: Diagram of a particle with the added geometry. The center and edge vertices can have different RGBA values so Gouraud shading the triangles yields fuzzy-edged structures rather than sharp points.

Each particle is modified to consist of a set of non-overlapping coplanar triangles which all share one common vertex: the world coordinate position of the particle itself (see Figure 3-1). The vertices of the individual triangles are defined relative to this point and are allowed to change in color (RGBA) and position only. Each vertex other than the center point belongs to exactly two triangles, thus the center is surrounded by triangles without gaps between them. This extra geometry is added to expand the range of rendering options for the particle without a huge increase in complexity.

By shading the triangles and allowing for transparency, the particles can be efficiently rendered as continuous-looking, fuzzy-edged shapes. Gouraud shading was used in this implementation to keep the shading model *contemporary* (see Figure 3-1). In the case of a regular N-gon with many sides, this shape approximates a circle. When non-opaque particles overlap, their colors can be blended in one of many ways to achieve desired results.

As with traditional particle properties like position and velocity, the RGBA values and relative positions of the added vertices are allowed to vary with time. Thus the particles can be made to squash and stretch (varying position), change in color (varying RGB), fade in or out (varying transparency), and in general take on arbitrary forms with planar topology.

We hide the two-dimensionality of the particles by always rotating so that they face the viewer before rendering. The particles are therefore reduced from being truly three-

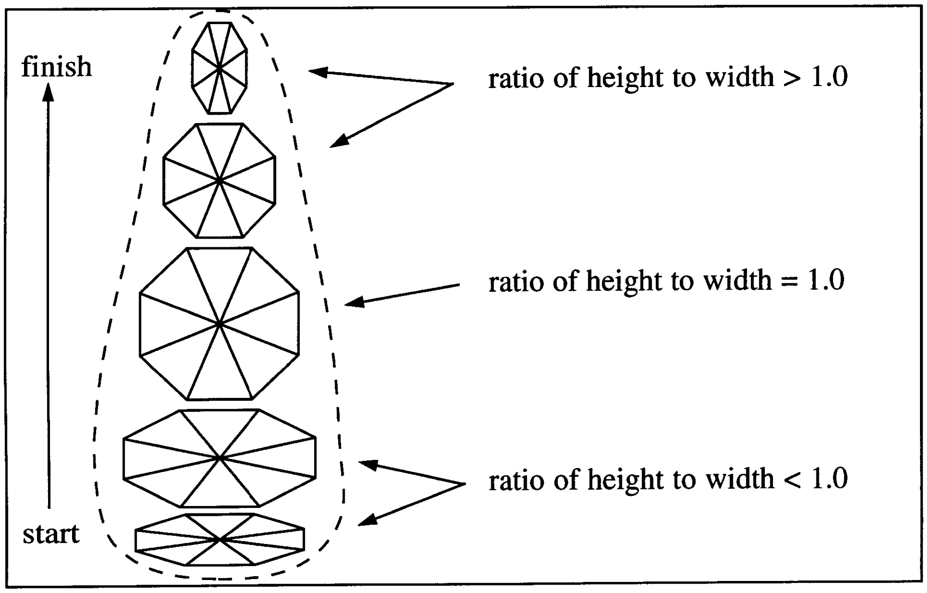

finish
start
ratio of height to width > 1.0
ratio of height to width = 1.0
ratio of height to width < 1.0

Figure 3-2: Approximating a candle flame with the rendered trajectory of a single particle. The ratio of the particle's height to its width changes in this example from less than 1 at the start to greater than 1 at the end, yielding the general shape of a candle flame. Note that the RGBA value of the center vertex as well as those of the edge vertices can vary along the trajectory as well.

dimensional entities to being spherically symmetric three-dimensional "blobs," that is, their rotational degrees of freedom are removed.

Although not used in this formulation, a four-by-four transformation matrix could be associated with each particle to re-introduce the rotational degrees of freedom. The particle would first be rotated to face the viewer, then the attached matrix could be used to transform the added vertices of the particle. To impart non-isotropic shape to the two-dimensional geometry, this matrix could depend on the current position of the particle relative to the world space coordinate system, and one could even set up a noise model to generate these matrices. The triangular dynamic geometry will then be rendered accordingly.

3.2.2 constructing flames with particles - the *long exposure* method

The smallest flame considered in this thesis is the standard diffusion flame: a candle. This is the atomic element from which larger fires will be built in conjunction with the spreading model (via the bi-partite hypothesis).

We model a single flame as the complete trajectory of a single particle, similar to the way a blade of grass was rendered in [22] (see Figure 3-2 for a conceptual sketch of this

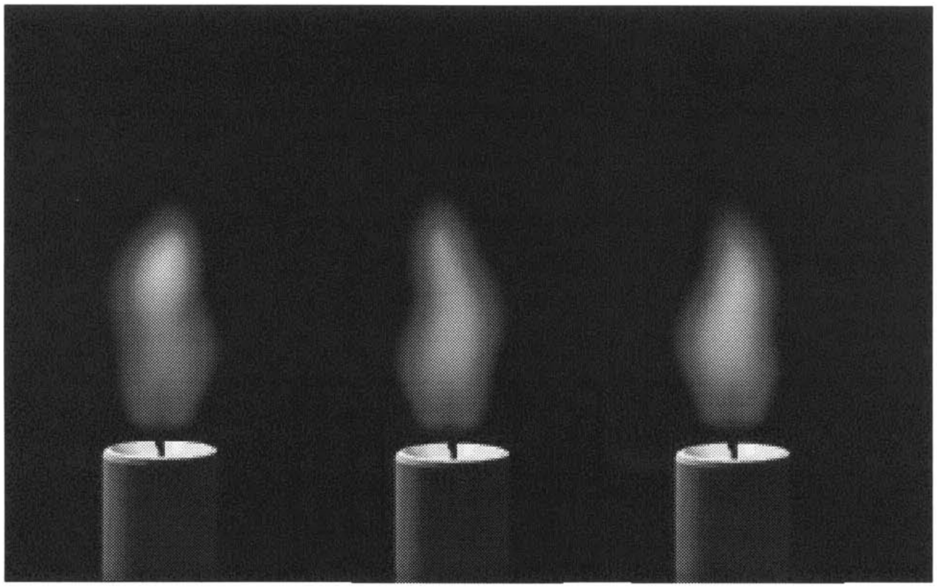


Figure 3-3: Candle flames. Each is the 25-step trajectory of a single, 20-triangle particle. The candlesticks were added for effect.

process). The positions and RGBA values of the added geometry are made to vary over time with the position and velocity of the particle, and as a result the open-shutter image of a single particle takes on candle-like shapes and colors (see Figure 3-3).

The function used to alter the geometries of the particles in Figure 3-3 is based on examination of the shapes of actual flames from pictures in the literature ([1] [6]), contained on the "Pyromania" CD-ROM of digitized fire footage[11], and from observation. We use a sinusoidal function based on the current "age" of the given particle, where age zero corresponds to the base of the flame. In Figure 3-3, the maximum age was 24 because the flames were the 25 step trajectories of a single particle.[1] More complex functions could clearly be devised for greater control over the shaping.

It is important to note the significance of the type of blending used when particles overlap. The triangles that compose an individual particle can have only a linear RGBA gradient due to the Gouraud shading technique, thus if a Painter-like method of blending were used (where new items completely overwrite old ones in the framebuffer), the color profiles across a given scanline would be piecewise linear.

Figure 3-4 shows color profiles across both a real and synthetic flame. The kind of

[1] In Figure 3-3, the vertices of the particle were placed on the unit circle, and they were all scaled by a given factor in each frame. The function used for the scale factor in the x direction (the width) was $\text{xfac} = 0.67 + 0.4 * \sin(\pi * \text{age}/24)$. The scale used for y was linear and increasing, with slope 0.032.

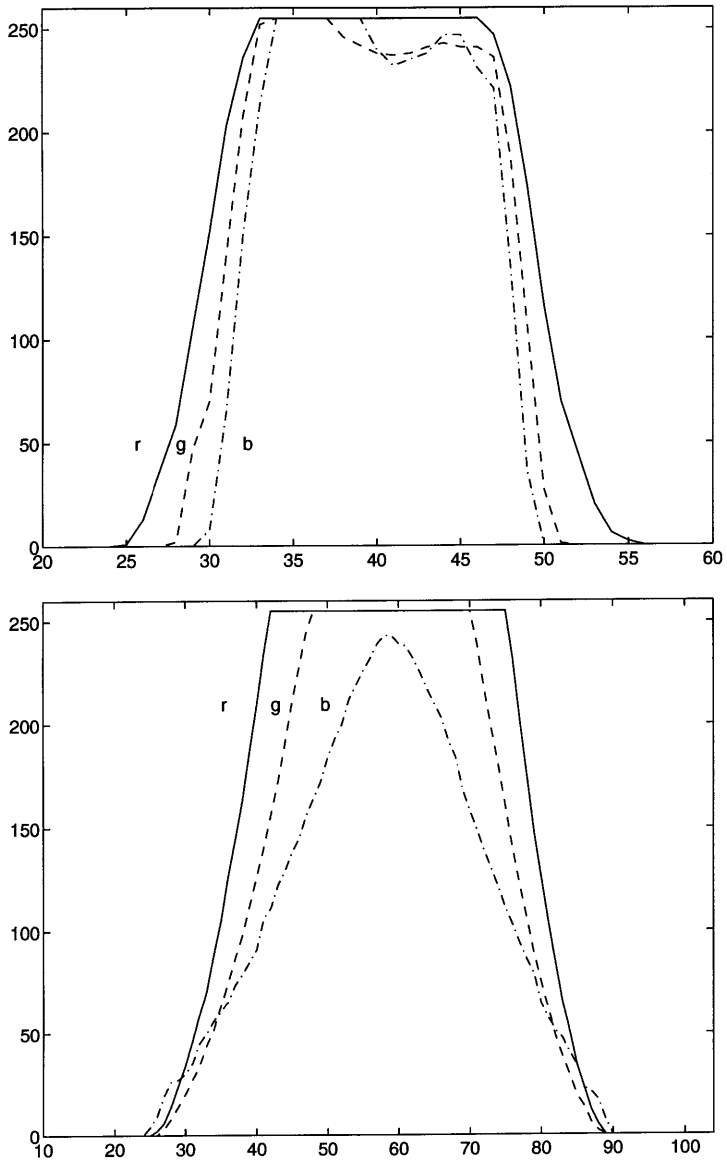


Figure 3-4: Plots of RGB values (y axis) versus position (x axis, in pixels) for real (top) and synthetic (bottom) flame images. These color profiles show some of the differences (such as asymmetry and edge variation) between actual flames and those rendered with the fuzzy particle model. See Section 3.2.2.

blending used in the synthetic examples shown (Figure 3-3) accumulates contributions to each RGBA component until saturation, and as a result shares the color characteristic with the real flame that $R > G > B$ along the edges.

The synthetic profile differs from the real, however, in two important ways. The first is that in the synthetic case, the outer edge of the flame marks the begining of the ramp up from zero for all three components (RGB), while in the real case they begin at different places. The second difference is that since the fuzzy particles all overlap on the specific flame trajectory, the color profiles will not contain asymmetric features like those observed in the real profile of Figure 3-4. This characteristic of the flame model will be addressed in Section 4.1

3.2.3 flames in the wind

The examples in Figure 3-3 show particle system flames with a heretofore unexplained perceptual feature: significant bending in their shape around a generally vertical orientation. This is the result of placing the flames in a wind field[2].

Recall that to render a flame using the long exposure method we start a particle at the desired base of the flame ($\vec{p}(0)$) with a certain velocity ($\vec{v_p}$) and allow it to move through a certain number of steps, accumulating color in the framebuffer while doing so.

Using a (Eulerian) finite difference implementation of Newton's second law we can determine the trajectory of the particle given a force field ($\vec{F} = m\vec{a}$). This force field is the three-dimensional wind field $\vec{u}_\infty(\vec{p}, t)$ in which the flames exist.

In the absence of external winds, candles and other flames burn "upwards," or, more specifically, in the direction opposite gravity. This is because the release of energy in the combustion region forms convection currents: the neighboring air rises after being heated by the flame, and cooler air from the surroundings flows in to replace it. Thus the idea of zero external wind is inappropriate around a flame which exists in both an atmosphere and a gravitational field.

We describe the external wind field $\vec{u}_\infty(\vec{p}, t)$ at a point $\vec{p}$ and time t as the vector sum of two parts, one due to the convection currents and the other independent of them:

$$\vec{u}_\infty(\vec{p}, t) = -k_g\vec{g} + \vec{w}(\vec{p}, t)$$

[2] which will be shown to also provide proper motion of flames when the object they are burning on moves.

where $\vec{g}$ is gravity (assumed not to vary with time or position) and $\vec{w}(\vec{p},t)$ is the convection-independent wind. The coefficient k_g is varied to scale the contribution of $\vec{g}$ to $\vec{u}_\infty(\vec{p},t)$[3].

It is often helpful to divide $\vec{w}(\vec{p},t)$ into large-scale and small-scale components, such that global behaviors can be defined with one function and more turbulent, local interactions with another. One example is if you want a generally vortex-shaped wind field to be perturbed by turbulence; it is easy to define the vortex portion as one function and the turbulent disturbances as another. Therefore the expression we use for $\vec{w}(\vec{p},t)$ is

$$\vec{w}(\vec{p},t) = \vec{w}_L(\vec{p},t) + \vec{w}_S(\vec{p},t)$$

where $\vec{w}_L(\vec{p},t)$ and $\vec{w}_S(\vec{p},t)$ define the large-scale and small-scale portions of the wind field, respectively. This paradigm has been used successfully in modeling turbulent wind fields for gaseous phenomena [27], and it represents a separation of local and global forces seen in reaction/diffusion systems [31] and random field models [20], to name a few.

The final expression for the wind and therefore the force field is thus

$$\vec{u}_\infty(\vec{p},t) = -k_g\vec{g} + \vec{w}_L(\vec{p},t) + \vec{w}_S(\vec{p},t).$$

Normalizing the mass of the particles and plugging into $\vec{F} = m\vec{a}$ gives

$$\vec{u}_\infty(\vec{p},t) = \vec{a}.$$

In terms of the update equation for the velocity $\vec{v_p}$ we have

$$\vec{v_p}(t+1) - \vec{v_p}(t) = \vec{u}_\infty(\vec{p},t) = -k_g\vec{g} + \vec{w}_L(\vec{p},t) + \vec{w}_S(\vec{p},t).$$

If $\vec{w}(\vec{p},t) = \vec{w}_L(\vec{p},t) + \vec{w}_S(\vec{p},t) = 0$, flames will bend only in the direction countering gravity and will exhibit no temporal dynamics. Realistic “flickering” or “wavering” of individual flames through time can be accomplished by giving $\vec{w}(\vec{p},t)$ nonzero values.

The candle examples in Figure 3-3 use $\vec{g} = -\hat{\jmath}$, $\vec{w}_L(\vec{p},t) = 0$, and a small scale wind field

$$\vec{w}_S(\vec{p},t) = ((\vec{p}(t) - \vec{p}(0)) \odot \hat{\jmath})C\vec{N}(\vec{p},t) \tag{3.1}$$

[3] in this work, k_g has been equal to 1.0. Note that if $k_g = 0$ and $\vec{w}(\vec{p},t) = 0$, the flames will not take any shape - equivalent to the zero gravity case.

where $\vec{N}(\vec{p}, t)$ is a function that returns the gradient of a band-limited noisefield as a 3-vector[4], C is a constant[5], and $\odot$ denotes taking the inner product. The coefficient $((\vec{p}(t) - \vec{p}(0)) \odot \hat{j})C$ increases in magnitude with the height of the particle, thus this function perturbs the position of the particle increasingly with distance from the base of the flame. This yields motion analogous to the motion of a flag, being fixed at one end and free to move at the other.

$\vec{N}(\vec{p}, t)$ is indicated as a function of time and position to represent the standard technique of "moving through the noise field" in time. In this thesis, $\vec{N}(\vec{p}, t)$ corresponds to the following functon:

$$\vec{N}(\vec{p}, t) = \vec{N}(\vec{p}(t) - k_s T \hat{j})$$

that is, we move vertically downward through the noise field as simulation time (T) increases, scaled by the "sampling" factor k_s[6]. This gives the illusion of the perturbations moving upwards with the flame, and corresponds physically to disturbances in the convection current moving upwards with the current. Note that k_s could be estimated by observing the frequencies of the motion of actual flames, although this oversteps the bounds of the Master's work pursued here.

Although the finite difference scheme and use of terms like "position" and "velocity" imply and describe a physically-based model (albeit the physics of projectiles and not flames), the flame model is not meant to be physical and in fact a very non-physical technique was employed at times to keep the particle system flames from looking like sequences of discrete fuzzy blobs. At each time step, the velocity of the particle was normalized in magnitude ($\vec{v_p}(t+1) = \vec{v_p}(t+1)/|\vec{v_p}(t+1)|$), thus keeping the step distance between renderings constant. This form of fixing the magnitude of the velocity was not used for the flames in Figure 3-3, but it was a required step in the simulations of fires described later where the wind fields became more intense.

For a physical interpretation of these non-physical techniques, consider a linked chain of underwater sausages, or tubular clown-like balloons filled with Helium, held at one end and allowed to swing freely at the other in a wind. They can bend in the wind (or water)

[4] like Perlin's noise function from [18], thus this flame model is a hybrid of noise-based and particle techniques.

[5] a value that gives pleasing motions is 0.04

[6] typical values of k_s will be discussed in Section 3.5.

and even “double back” but their primary tendency is upward. Even this, however, isn’t an exact physical comparison; this portion of the flame model is heuristic by necessity, not physically based, and by no means the only one of its kind that would give decent results.

3.2.4 light emission

Because of the world-space definition of the particulate flames, a light source can be easily associated with individual flames by making its position $\vec{l}$ be a function of the particle positions $\vec{p}$. The mean of $\vec{p}$ over the frame lifetime of a particle is one such function; of course, others should be examined.

3.2.5 two other examined (and rejected) models

It is worth noting that two variations on the dynamic particle technique were investigated and rejected. The first was a more traditional use of particle systems in that multiple, independent, fuzzy particles were used to construct the flame. Like [21] and [25], the base of the flame would be considered a source that would jettison particles in the direction of the top of the flame with slightly varying initial conditions. This method proved to be unsatisfactory primarily because of flickering that came about when an old particle would die or a new particle would be born: since the contributions of the various blobby particles are summed to form the final rendered image, the discontinuous appearance or disappearance of a particle tended to catch, and distract, the eye.

The second technique attempted to shape the dynamic geometries in such a way that the individual particles appear as entire flames and not as circular, fuzzy, blobs (for example, the dashed line in Figure 3-2 would represent the contour of the particle’s added geometry). Thus one particle could be used without requiring the long-exposure rendering scheme used for the flames in Figure 3-3. Because of the linear nature of Gouraud shading, however, multiple, accumulated, renderings of the same particle were required to get saturated central regions of the flames and tapered edges.

The largest drawback with this second technique was that it did not provide for arbitrary shaping of flames under different environmental conditions. It is not uncommon in real fires for flames to bend into U-like or, in general, concave shapes, and it is clear by looking at the model that these would be impossible to render using the multiple-triangles-with-one-shared-vertex geometry proposed above.

3.3 spreading model: how fires move over objects

Fires spread because heat is transferred from the burning to the non-burning regions of the combustible fuel, eventually causing them to erupt into flames. The boundary between these regions is called the *surface of fire inception* (or the *inception boundary*) [30], and to properly model spread we must determine how fast this boundary travels into the unburnt regions.

3.3.1 physics of flame spread

Let q be the heat flux rate (in Joules per unit area per unit time) through the boundary. The spreading rate we are interested in, v_s, is a one-dimensional quantity[7], namely, the speed (in unit distance per unit time) the boundary travels in the direction of its normal (see Figure 3-5). We can relate q to v_s by considering energy conservation across the boundary. On the burning side, in a time t a total of $qt\Delta A$ Joules travel through an area ΔA of the boundary. On the unburnt side, if the fuel density is ρ and the difference in thermal enthalpy between the fuel at ignition temperature and its original temperature is Δh (in Joules per unit mass), then the amount of energy needed for the region ΔA to move a distance $v_s t$ is $\Delta A v_s t \rho \Delta h$. Equating these two expressions and cancelling terms yields

$$q = \rho v_s \Delta h. \tag{3.2}$$

This equation is sometimes referred to as the fundamental equation of fire spread [30]. Informally, it tells us that in order to determine the spread rate v_s, we must know the form of the heat flux rate, q.

The true form of q is complicated and difficult to solve [5]; however, the experimental results from the community studying flame spread can aid in approximating it. They have identified three modes of heat transfer which are dominant among all the contributions to q.

Before describing these modes in detail it is necessary to describe the different regimes of spread. There are two kinds of spread recognized: *concurrent* or wind-assisted spread and *opposed* spread. In the former, the spreading direction v_s has a positive component in the

[7] note the difference between $\vec{v_p}$, the vector-valued particle velocity and v_s, the spread control point speed.

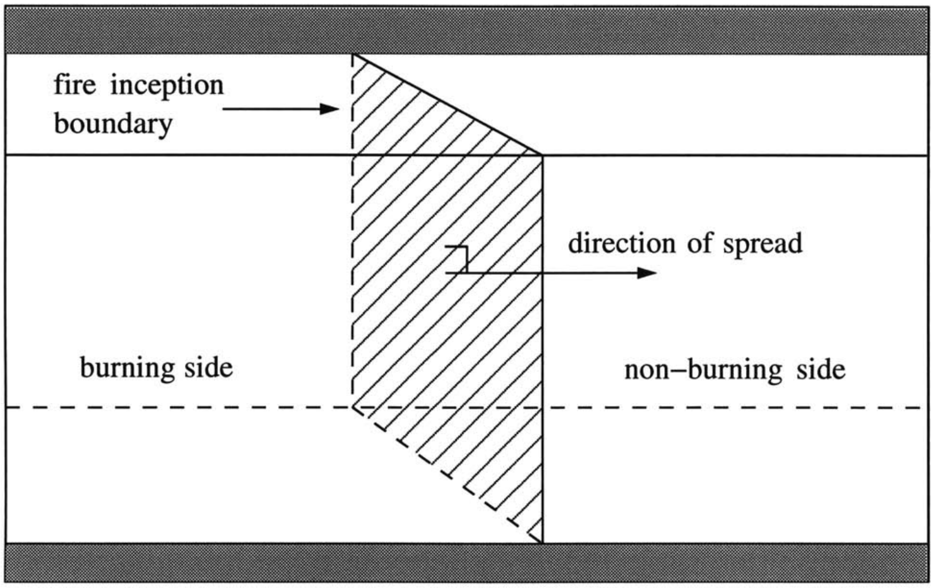


Figure 3-5: Cross-section through the fuel. The surface of fire inception moves through the unburnt fuel in the direction of its normal. Heat flows through the boundary from the left to the right and determines the spread rate v_s.

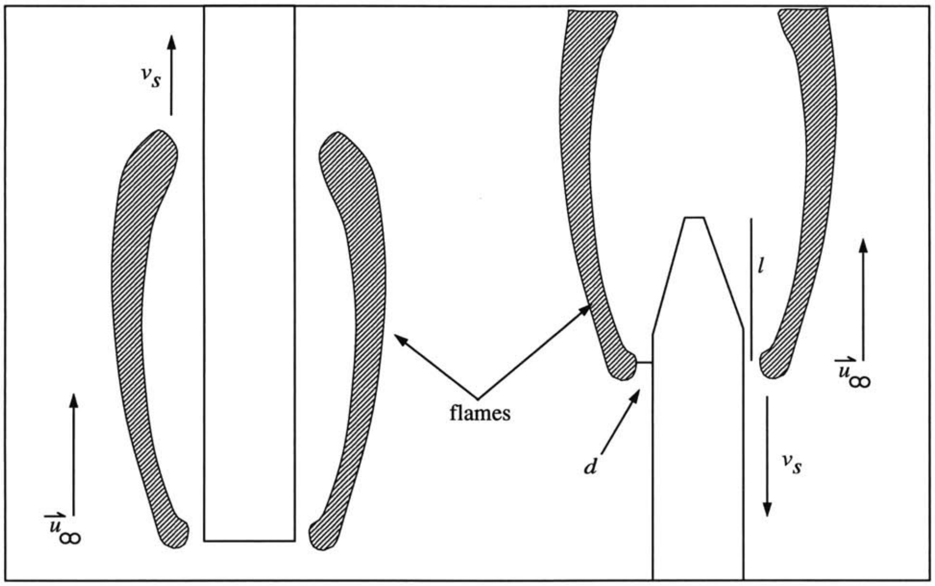


Figure 3-6: Concurrent and opposed spread. In concurrent spread (left), the flames travel in the same direction as the external wind $\vec{u}_\infty$. In opposed spread, they travel in the opposite direction. Note how in the concurrent case the flames lie close to the unburnt fuel; this is why this kind of spread is faster and more treacherous than opposed spread.

direction of the external wind (called $\vec{u}_\infty$), while in the latter it is negative (see Figure 3-6). Gravity induces an upwards wind in the form of convection currents around the flames, so in the absence of any external wind $\vec{u}_\infty$ is purely a function of gravity. A fire started in the middle of a vertical wall with only gravity-induced winds will therefore consist of both types of spread, as the upward evolution will be concurrent and the downward will be opposed.

Fuels are also divided into two categories: thermally thin and thermally thick. Thermally thin fuels are those (like a sheet of paper) whose thickness is such that the temperature gradient across them is negligible, i.e., the temperature is constant through the fuel. Thermally thick fuels, however, do maintain a temperature gradient and the thickness affects the rate of spread.

A given mode of heat transfer has been shown to dominate each of the four categories of spread (these being concurrent and opposed flow on both thermally thin and thick fuels, see [30] and [4]). Since a given fire could contain a combination of these, they will all be described.

opposed flow over thermally thin fuels

Opposed flow over thermally thin fuels is dominated by heat conduction through the gas. For a sheet of thickness L and width w, the expression Lwq gives the amount of energy per second needed for ignition. The heat is assumed to come from a flame located a normal distance d from the surface of the fuel over an average distance l in the tangential direction upstream from the surface of flame inception. Therefore,

$$Lwq = lw\lambda_g(T_f - T_i)/d$$

is a reasonable approximation of this quantity (λ_g is the thermal conductivity of the gas, T_f is the flame temperature, and T_i is the temperature of the unburnt fuel). The velocities of the gas near the inception boundary need to be small for this mechanism to function, thus l and d will be the same order of magnitude and

$$q_o = \lambda_g(T_f - T_i)/L \tag{3.3}$$

where the subscript o is to denote opposed flow. This is equation 6 from [30].

opposed flow over thermally thick fuels

Opposed flow over thermally thick fuels[8] is dominated by heat conduction through the solid. The expression for simple conduction (3.3) can be used, except with λ_g replaced by λ_s, the thermal conductivity of the solid, and L treated specially for more accuracy[9] [30]. This is included for completeness: In this work we have only considered thermally thin fuels.

concurrent flow over all fuels

Concurrent flow over fuels of all thickness is dominated by radiation from the flames. The flames stretch in the flowing air currents to lie over unburnt regions of the solid fuel and they radiate heat to its surface. This mode explains the rapid upwards spread we see when we light a match and hold it upside down, for example. An approximate model treats the flames basically as linear in shape with an orientation angle Θ_f relative to the surface and a length h_f. The expression for q_c (concurrent flux rate) is

$$q_c = \epsilon_f \sigma_b T_f^4 h_f \sin \Theta_f / L$$

where ϵ_f is the emissivity of the flame and σ_b is Boltzmann's constant (see equation 7 in [30]).

For a given fuel, q can be approximated using a combination of these major modes, and v_s can be determined from the fundamental equation of fire spread. We have approximated q by summing the weighted contributions of the opposed and concurrent spread mechanisms, using the expression[10]

$$q = A_o q_o + A_c q_c.$$

The coefficients A_o and A_c depend on the orientation of the spreading direction with respect to the external wind $\vec{u}_\infty$ such that $A_o = 0$ when the flow is concurrent and $A_c = 0$ when the flow is opposed. Since horizontal flow is considered to be the same as opposed flow there are situations where both the concurrent and opposed mechanisms transfer heat to the unburnt

[8] Polymethylmethacrylate, or PMMA, is the standard thermally thick test fuel used.

[9] For our purposes, the relative difference between concurrent and opposed rates of spread is the key visual feature, making the thin fuel approximation of (3.3) valid; that is, L is constant.

[10] for simulations run in this thesis, T_f, T_i, and L were held constant and the non-varying value used for q_o was 0.09. The value for q_c was 0.13. These values have been scaled for visually agreeable synthesis - consult tabulated values for more physical results.

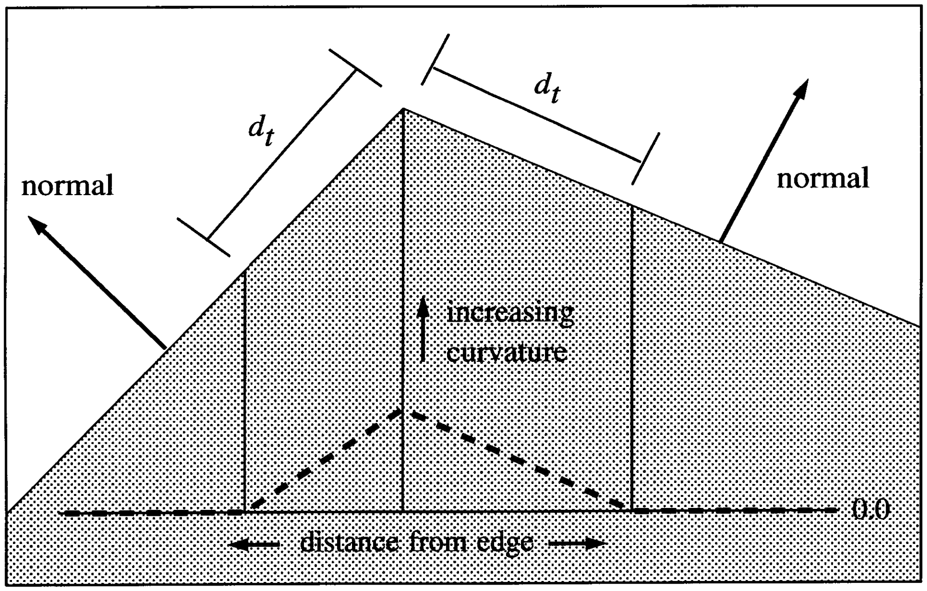


Figure 3-7: The estimation of surface curvature, shown on a cross-section through a polygonally-defined surface. Curvature is considered zero when further than d_t from an edge, and ramps up to a maximum value (which is a function of the angle between the normals, Θ) as the distance to the edge goes to zero.

fuel and both A_o and A_c are non-zero. The values used in this work are $A_o = 1.0$ (the opposed mechanism always contributes), and A_c is computed as the dot product of the normalized spreading direction vector and the normalized projection of $\vec{u}_\infty$ in the plane of the polygon, with negative values clipped to zero (i.e., the flow is not concurrent).

One factor missing from this approximation to q is the effect of surface curvature on the spread rate. Fire burns preferentially where more oxygen is exposed, and in the case of a burning object this will be on external edges and corners. Likewise, when the amount of oxygen is reduced, flames will spread more slowly. This happens on internal edges where the solid angle of exposed air is less than 2π. The curvature is estimated using the normal of the current polygon and the normal of the adjacent polygon: if the angle Θ between the normals is zero, the curvature is zero.

Because our surfaces are defined by polygons there is local flatness everywhere except when precisely on an edge. To give the illusion that the polygons represent curved surfaces, for points within a predefined threshold distance d_t of an edge the curvature value at the edge is approximated by $1 - d/d_t$ where d is the perpendicular distance to the edge from the current point, and then the curvature is used to scale the velocity[11]. This gives a linear

[11] a typical d_t value used is 0.1 times the average edge length of the object's polygons.

ramp up to the true curvature value for points approaching the edges.

To include curvature into the expression for q, we have scaled q by the coefficient $(1 - d/d_t)\Theta/\pi$. Other functions of the surface curvature may also be used, but this simple approximation gives visually satisfactory behavior with real-time performance.

The fundamental equation (3.2) gives our final expression for the spread rate:

$$v_s = (1 - d/d_t)\frac{\Theta(A_o q_o + A_c q_c)}{K} \tag{3.4}$$

where $K = \pi\rho\Delta h$ formally, but represents a factor that sets the rate of the spreading and is best tuned visually[12].

3.3.2 implementation of spread model

To try and preserve *computability*, we have chosen to represent the inception boundary with individual *spread control points*, such that the polyline connecting the points is an approximation to what would be the true curve. The points are given in counterclockwise orientation such that the burning region is to the left and the nonburning region is to the right of the line. In our implementation, the control points are stored in a linked list.

When begining a spread simulation, a predefined number of control points[13] are placed at the ignition site on a polygon and given evenly spaced radial directions out from that point in the plane of the polygon. Each point evolves through time based on the value of v_s calculated from (3.4) and the spreading direction at that point. To avoid undersampling the surface, when two points exceed a threshold distance d_s from each other a new point is inserted between them with its own heading[14].

The inception boundary must be confined to move on the surface of an object for the real magic of burning to take place. We have chosen to implement the spreading equations on polygonally defined objects since they are so often used in computer graphics; however, they have not been proven to be the ideal representation for this form of spreading.

We pre-compute the matrices needed to rotate one polygon's normal to the normal of each of its neighbors and store these with each polygon[15]. If a spread control point will

[12] the value used here was $K = \pi/2$.

[13] Eight or more points were used in the experiments here. With too few the discrete nature of the boundary can be seen early in the simulation.

[14] see Section 3.5 for typical values of d_s.

[15] Note that Θ in equation 3.4 is just the angle of this rotation.

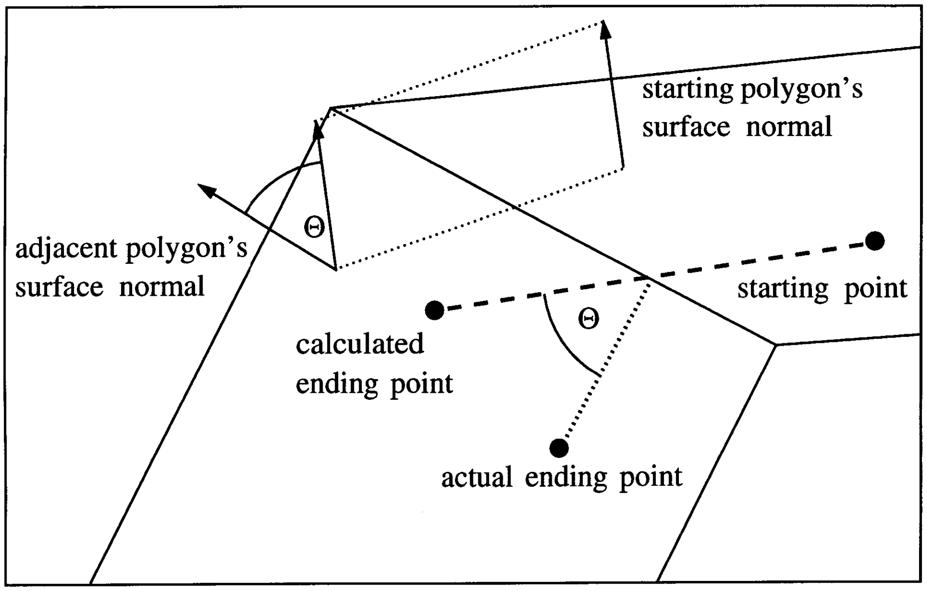

starting polygon's surface normal
adjacent polygon's surface normal
Θ
starting point
calculated ending point
Θ
actual ending point

Figure 3-8: Keeping the spreading on the polygonal model. When the calculated future position of a spread control point lies outside of the current polygon, we split the movement into two steps at the point of intersection with the edge. The portion of the segment outside the polygon is rotated into the neighboring polygon using a pre-stored matrix, forcing the spread to remain on the object.

cross an edge in a given iteration, its heading vector is rotated using the prestored matrix so that it lies in the plane of the new polygon (see Figure 3-8). If no polygon shares that edge, the spread point is terminated at the point of intersection with the edge.

On topologically non-planar surfaces, we must be careful to prevent the boundary from wrapping around and crossing itself, or we risk burning regions already designated as burnt ("double burning"). This can be accomplished by marking all burnt places, but this is computationally cumbersome as it requires a finer representation (Nature finds it easy to do in parallel). Alternatively, one can test the segment formed by the current and new position of a given control point for intersection with the rest of the boundary, which itself is just a set of line segments. In certain cases where double burning is not an issue, such as with topologically planar surfaces that undergo little to no world-motion, one can eliminate the intersection test and increase the speed of the simulation. Figure 3-9 shows the segments which must be tested for two arbitrary control points to properly prevent double burning. We have used the three-dimensional segment intersection algorithm described in [7].

The evolution of the spreading boundary can be visualized by drawing a connected line or curve between the control points. This system runs in real time on a 100 MHz SGI

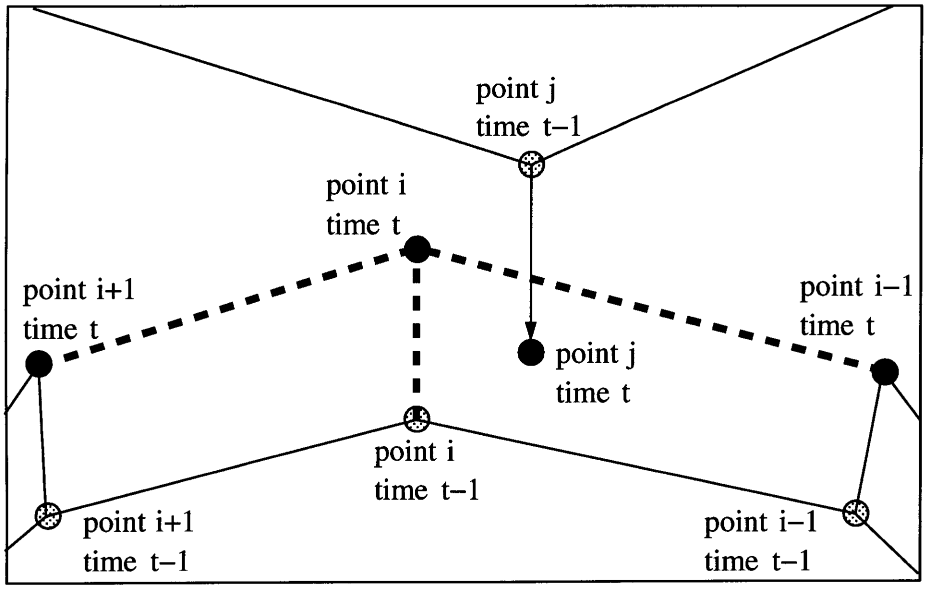


Figure 3-9: Spreading into regions that have already been burnt can be prevented by testing the boundary for intersections with itself. For point i, the dashed segments in the diagram must be tested for intersection with the segment defined by point j's location at time t and point j's location at time $t-1$ for point j to be allowed to move. Since we must loop through all the control points, we need only test two segments per point, not all three.

Indigo2 Extreme for a reasonable new-point threshold distance d_s.

3.3.3 other uses for spreading models

It is interesting to note that with a different governing equation for v_s, this model can be used to spread anything over the surfaces of polygonally-defined objects. Spills, for instance, are just the evolution of a spreading boundary under the force of gravity. This force would be relatively easy to combine with capillary forces to model absorption of dyes on fabrics as well.

3.4 fire combinatorics: combining flame and spread

Success of the bi-partite hypothesis is based on successful solution of three distinct subproblems, the last of which is being able to combine the first (flame) and the second (spreading) models in such a way as to produce fires.

Simply, we combine the models by treating each spread control point as a potential source of flame particles. As the boundary evolves over time, each spread control point records the total linear distance it has traveled and "drops" a new flame particle after

Figure 3-10: An example of real flames from the Pyromania! CD-ROM[11].

a particular distance has been traveled, resetting its distance counter to zero when this happens. By adjusting the minimum drop distance d_d (less than the linear diameter of a single flame particle, typically), different densities of flames can be achieved. The position of the spread control point becomes the origin of a new flame particle, which will exist for a certain duration based on the flammability of the fuel being burned. Thus, we have independent control over both flame density ρ_f and flame lifetime l_f, making it easy for the model to incorporate changes due to the geometry and flammability of the material on fire.

The flame particles on the surface of the polygonally defined objects are assigned initial velocities $\vec{v_p}$ (not to be confused with spread control point velocities v_s) parallel to the outward facing normal of the polygon they are rooted on. This is done primarily so that the flames lie on the exteriors of objects where they will be shaped in the wind field, but the choice of the normal as the initial particle velocity is otherwise arbitrary.

Once flames are placed on the object and given an initial velocity $\vec{v_p}$ they will appear as they should in the three-dimensional world since the flame model already handles wind fields in world coordinates. If the burning object happens to move in the world, it creates a large-scale wind field in the direction opposite the motion that can be easily realized by setting $\vec{w}_L(\vec{p}, t)$ equal to the negative of the motion vector.

Flames in larger-than-candle fires, while they tend to remain distinct, often bend towards and away from other flames and often seem to combine with other flames (see Figure 3-

10). This perceived "communication" can be achieved without $O(n^2)$ particle-to-particle interaction calculations by varying the scale of the input $\vec{p}$ to the noise field $\vec{N}(\vec{p}, t)$ in equation 3.1 ($\vec{N}(\vec{p}, t) = \vec{N}(k_p\vec{p}, t)$). Because of the interpolated nature of the noise, if the positions $\vec{p}$ of particles are scaled by k_p to bring the linear dimension of the conflagrant object to on the order of one lattice site in the noise space, there will be continuous transitions in $\vec{w}_S(\vec{p}, t)$ across any flames that are attached to that object. In the limiting case where $k_p = 0$, $\vec{w}_S(\vec{p}, t)$ will have degenerated into a constant field that varies with time and not with position[16].

3.4.1 another visual cue: surface charring

Certain cues which aren't directly part of the spread or the flame model can be added to make fires more realistic; these become a part of a combinatorical bag-of-tricks that can be fairly easily implemented within the model. The one addressed here is char; see Chapter 4 for others.

The location of an individual flame particle on the surface of a conflagrant object tells us more than just where to place the flame. After the fuel is extinguished, for example, this location is just where one expects to see the blackening effects of char.

The example shown in Figure 3-11 places a circular, fuzzy, translucent black particle at the flame location once the flame is extinct. This geometry blends upon rendering with the underlying object and gives the illusion of char on the object. If a texture map were used with the object, it could be similarly changed to show char.

3.4.2 notes on parallelization

Nature controls fires in real time due to massive parallelization of the computations involved; it is interesting to note how both parts of the bi-partite model lend themselves well to parallelization thanks to the computational triviality of updating each individual element. Determination of the value of the noise-based wind field can be done on an individual flame basis, thus each flame could inhabit a processor and compute its own trajectory independent of the others. Spreading calculations could also occur in parallel as each control point moves without knowledge of the others, with a single-neighbor communication required in checking

[16] typical values for k_p will be discussed in Section 3.5.

Figure 3-11: Example of charring the surface. Fuzzy black particles are placed on the object surface at the position where each flame particle extinguishes. Note how the texture of the surface can be seen through the translucent char.

name	variable	section	low value	high value
global time scale	K	3.3.1	0.004	0.4
drop distance	d_d	3.4	0.01	1.0 (distance units)
spread control point separation	d_s	3.3.2	0.01	0.2 (distance units)
inverse interflame cohesion	k_p	3.4	0.1	10.0
flame noisyness	C	3.2.3	0.0	0.04
flame sampling	k_s	3.2.3	0.0	1.0
curvature distance	d_t	3.3.1	0.01	0.5 (distance units)
flame lifetime	l_f	3.4	1	1000 (frames)

Table 3.1: This table shows names, corresponding variables and sections in the text, and value ranges for some of the more useful widgets used to interact with the model.

for double burn.

3.5 controlling the synthesis

This chapter introduces a long list of variables and constants that shape the results of the fire synthesis model. The specific values mentioned are guidelines only - the real power of writing *controllability* and *computability* into the model is in exploring these parameter spaces interactively.

3.5.1 sample parameter values

We have implemented the fire model in C using a mixed-model combination of the SGI graphics library (GL) and X/Motif. Motif was selected because of its ability to easily create and manage value-changing widgets. Table 3.1 lists the labels, variable names, and value ranges of some of the Motif "scales" (sliders) that were found to be useful in exploring the dynamic range of the model. It is in NO WAY meant to be an exhaustive list.

The ranges of the parameters in Table 3.1 measured in "distance units" were chosen for polygonal models of a particular size (the edge length of a given polygon in a model was typically about 0.5 units). Because many graphics objects are defined in arbitrarily-scaled "object coordinate systems," one is advised to scale these ranges to the proper typical edge length of a model's polygons.

3.5.2 a typical interaction with the system: playing with fire

Imagine that your goal is to burn a graphical object the size of a building. Typically, you would either start with a polygonal model or you would make one; suppose in this case that the model you choose is a unit cube. You start a fire somewhere on the side of the cube using the default settings for the various model parameters, and you find that these don't quite give the slow, controlled spread you are imagining. One technique would be to change the physical model parameters of fire temperature, etc., however, it is probably easier to simply reduce the global time scale K.

Once the spread seems to move at a nice rate you begin to focus on the flames themselves. The default settings tend to place very few flames on the cube and you find that your resulting fire looks unfortunately like a bunch of individual candles. Decreasing the drop distance d_d or decreasing the spread control point separation d_s will increase the density of the flames on the surface and you adjust these until the desired effect is achieved.

Now you find that there are plenty of flames, but the ones on the left side of the cube and the ones on the right side tend to move together in a very correlated fashion. This is generally symptomatic of having an inverse flame cohesion k_p that is too low for the desired fire scale. Flames in real fires tend to show more "cohesion" when the fires are small - a burning matchbook, for instance, will not appear to be composed of individual flames. The best solution is to raise k_p until the world coordinates of the points on the cube are scaled agreeably to the inherent scale of the noise field.

The flames now seem to be flickering "too much," that is, you find that there is no perceivable memory between frames - the frame to frame images of a given flame are uncorrelated. Most likely either the flame noisyness C is too high, magnifying so small perturbations in the shape from the noise field, or the flame sampling parameter k_s is too high, causing world space to map to entirely different regions of the noise field each frame. You experiment with reducing both until you get flames you like.

The current parameters now could be used to burn other objects, and you decide to add a garage to your cube-shaped building. Simply add the new polygons to the original model, and the fire behaves accordingly.

This process of diagnosing problems with the synthesis and finding easy solutions among a large parameter space is made possible by the many forms of control and the rapid computation guaranteed with a contemporary model. This kind of interactivity leads to rapid and repeatable solutions to synthesis problems.

Chapter 4

Further Work and Conclusion

The model proposed has successfully met the four C's as outlined in the first chapter, however, the doors are open for future enhancements in all aspects of the model discussed. Six are presented here for consideration.

4.1 enhancing the appearance of the particle model

The proposed flame model tends to give all flames a fuzzy contour. The dynamic geometries that cause this phenomena were added, as explained, to reduce the discrete, pointilistic look of traditional particle models. Thus there are two conflicting problems: we would like flames to have a cohesive nature within each flame, but we would like their edges to be crisper than the fuzzy particles allow.

This is to some degree taken care of when multiple particles overlap in the rendering of a pixel. Since the blending will stop at saturation, multiple particles rendered on top of each other keep a soft edge but ramp up quicker to a saturated value. However, one should not rely on or require having multiple particles. Zooming in on a flame should not be equivalent to zooming in on a single particle which, upon zooming, would grow dimmer. This scale invariance is a drawback of the particle model.

One could imagine a more fractal implementation of the fuzzy particle model for flames where particles of multiple scales are composited to form the final rendered flame. The largest particles might be on the scale of those presented in Chapter 3, but smaller particles allowed to vary off the center trajectory of the particle could be used to improve the synthesis. With such a formulation, zooming in on the flames would serve to enlarge the apparent scale of these smaller particles and even smaller particles would fill in the newly exposed small scales. Convincing asymmetries like that seen in the color profile of the real flame in Figure 3-4 could also arise from such a model.

In Section 3.2.5 we described other geometries that were examined in hopes of increasing interflame cohesiveness while maintaining nice edges, but these in no way were an exhaustive set. Two layers of triangles (an outer and an inner), for example, would provide some extra rendering capabilities by allowing for two linear ramps to be specified. Clearly, if one did not want to be limited to triangles, more complex and versatile colors could be available.

4.2 noise-based improvements to the flame model

It was noted in the background section that noise-based techniques have generated flames with the interflame cohesiveness and edge characteristics recognized here as being absent from (or at least hard to do with) the fuzzy particle model[18][17]. These techniques can also provide flames that disconnect from the tips of individual flames as they move away from the burning object - an effect that can be seen in the real example of Figure 3-10 but cannot be synthesized without modifications to the current particle model. It is worth examining techniques that could bring these methods into the realm of spreading.

The biggest problem with the noise-based techniques is that they don't map well to world space (that is, they are hard to get to naturally interact with their environment). Simple coordinate transformations can place a noise volume in some volume of world space, however, adhering such a volume to objects would be extremely difficult. Plus, as stated, volume rendering is at present computationally inefficient without specialized parallel hardware.

The proposed flame particle model exists easily in world space; two ideas are examined that could possibly bring noise-based enhancements to the rendered flames without abandoning this *convenient* and *contemporary* aspect of the fire model.

noise-based coloring of fuzzy particles

One potential improvement to the fuzzy particle flame model is to color the flames based on a noise-perturbed index into a color map just like the technique used for the solar corona in [18]. Currently, the color of the flames changes deterministically as the particle moves along its trajectory; the inclusion of noise-based coloring may give desired edge effects and disconnections that the current model cannot synthesize.

mapping noise-based flames to dynamic polygons

The particle flame model does not appear to be weakened by the requirement of using two-dimensional particles and forcing them to always face the image plane. This is because the particle spline is really defined in three-dimensions: if we rotate around a flame with a bend in its trajectory we will see this as a three-dimensional bend, even though the particles are just two-dimensional.

This lesson, applied to bringing noise-based flames into the world of spreading, implies that we might get by with two-dimensional noise-based flames (the noise-based flames of [18] and [17] are two-dimensional anyway) if we can arrange to have them reveal three-dimensional structure upon different views.

One method might be to model flames with a texture-mapped, dynamic set of polygons. To generate the geometry, the particle infrastructure would remain partially intact: the idea is that the trajectory splines which come from moving a particle through the wind field would be used to control the shape of a new set of polygons instead of fuzzy particles.

Figure 4-1 shows how a set of polygons would be erected about this spline. This new geometry would then be texturemapped with a two-dimensional, noise-based flame that is newly generated every frame. Since a two-dimensional slice through the three-dimensional noise field is all that would be used, careful bookkeeping of orientations in three-space would be necessary to extract the proper slice from the noise for a given view. These dynamic "textureflames" could then be placed on objects with the proposed spreading model.

Preliminary tests show that seamless mapping is possible onto a dynamic geometry formed around a particle trajectory. A good noise-based model of a single flame is be needed (like the one in [29]) rather than a corona-like model [18] because the texture must be transparent and black by the edges of the polygons. If it isn't, the edges of the dynamic polygonal structure that is the backbone of each flame would be visible.

The generation of new texturemaps in each frame for each flame is a time-intensive process and would probably be limited to final renderings. With efficient texture mapping hardware like the Silicon Graphics Reality Engine, a single texturemap can be used repeatedly in motion tests to watch the behaviours of the flames in the adjustable wind.

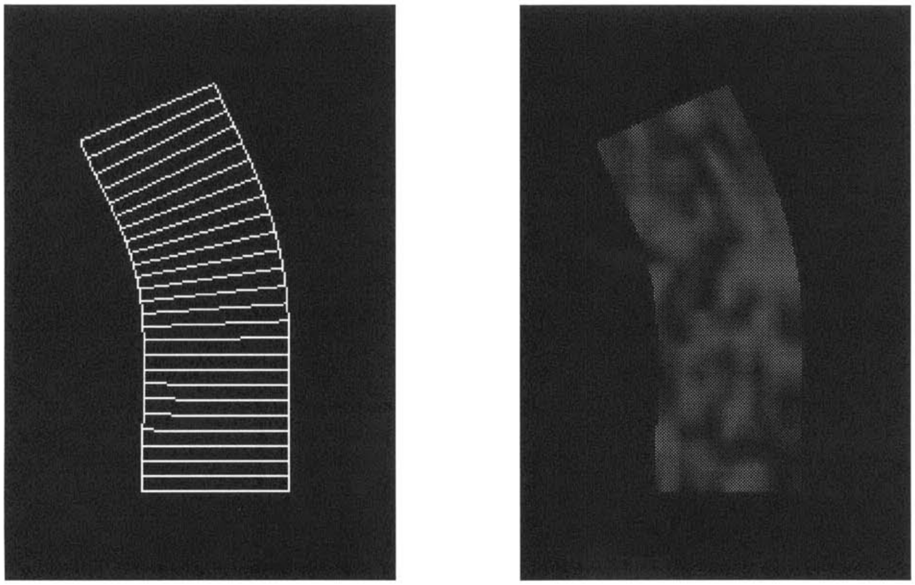


Figure 4-1: A proposed new model for flames tries to use the advantages of both the particle and noise-based models. A particle model like the one described in this thesis gives a control spline, around which a set of polygons is erected (left). These polygons can then be texture-mapped with a two-dimensional, noise-based texture.

4.3 proper determination of geodesics in spread model

Determining which segments to test for intersection is a non-trivial problem when spreading is allowed to occur on arbitrarily defined polygonal surfaces. If two neighboring control points happen to lie in different polygons, for example, segments must be tested against the *geodesic* that connects those two points[1]. In fact, for an arbitrary polyhedral surface, it is geodesics that must be tested for intersections with geodesics: to see if a given point has crossed the spreading boundary, the geodesic connecting its current position to its previous position must be tested against the geodesics connecting all the other spread control points. The problem of determining geodesics on polygonally defined surfaces has received attention in the past[12][15], and has been shown to run at $O(n^2 \log n)$, where n is the number of edges of the surface.

The unique nature of the fire model in this thesis can be used to speed up geodesic determination, however. For example, from the moment of ignition of a fire we know the geodesics between all spread control points (by definition, for they start in the same place), and if we can accurately identify geodesics in the next time step we should, by induction, be able to continue this process through the simulation.

As alluded to above, there are two different kinds of geodesics which are relevant. The first is the geodesic between a point and its previous position on the surface (this will be called an interpoint geodesic) and the second is that between neighboring control points (intrapoint geodesic).

Since the process of taking a given control point through to the next time step involves bending the velocity of the point about shared edges of adjacent polygons, we can build up our interpoint geodesics as we go by remembering the intersection points. This requires no extra calculations than those already being used.

Intrapoint (and interpoint, for that matter) geodesics can be determined more efficiently than $O(n^2 \log n)$ if the underlying surface is assumed to be relatively "flat." "Flatness" is perhaps most easily defined as the presence of a one-to-one mapping from the polygonal surface to the plane where the control points of interest are located. More specifically, label the individual control points c at a given time with index i, such that increasing

[1] the *geodesic* between two points is the shortest length path between the points that lies in the defined surface.

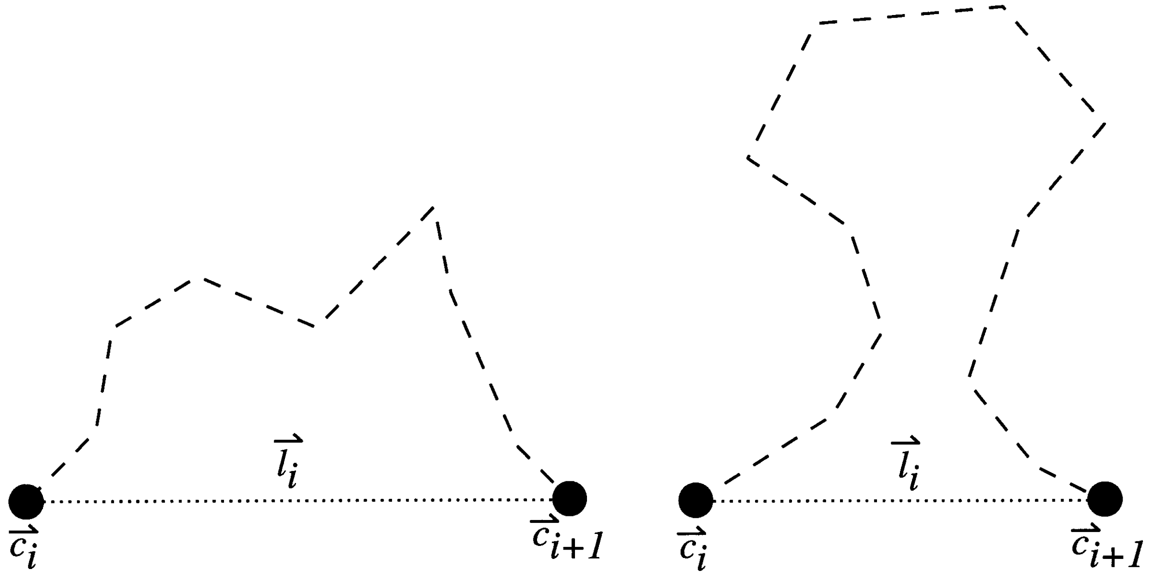


Figure 4-2: Examples in two dimensions of flat (left) and non-flat (right) connections between two spread control points ($\vec{c}_i$ and $\vec{c}_{i+1}$). Note that every vertical path from line $\vec{l}_i$ to the dashed polyline on the left intersects the polyline only once, whereas this is not the case on the right.

i corresponds to travelling along the spread boundary in the counterclockwise direction (unburnt region on your right).

Consider control point $\vec{c}_i$ (see Figure 4-2). The line $\vec{l}_i$ connecting point $\vec{c}_i$ and point $\vec{c}_{i+1}$ is given by

$$\vec{l}_i = \lambda\vec{c}_i + (1 - \lambda)\vec{c}_{i+1}$$

for $\lambda \in [0, 1]$. If the polygonal surface (or, in two dimensions, the polyline) defined as a function of $\vec{l}_i$ between these points takes on unique values for the given range of λ, then the surface between the points is said to be locally flat. Note that "local" is defined relative to a scale determined by $|\vec{l}_i|$.

If local flatness is guaranteed, then the geodesic between points $\vec{c}_i$ and $\vec{c}_{i+1}$ can be determined by projecting the actual, unconfined, three-space vector pointing between points $\vec{c}_i$ and $\vec{c}_{i+1}$ onto the surface; much like the intrapoint trick for easy geodesic calculation. This alone doesn't guarantee geodesics, but relaxation techniques that allow the edge intersection points to shift can be applied to minimize the total linear distance along the projected path.

4.4 spread over textured and non-polygonal surfaces

The realism of spread over objects with surface texture maps could be enhanced by using the data in the map to alter the spreading parameters. A table with a woodgrain surface might burn preferentially along the grains and the current spread model would have to be changed to allow for access to these maps.

The implementation of the spreading functions on other geometries, like procedural or parameterized surfaces, needs to be examined if these are the modeling primitives one uses. In fact, the problem of determining geodesics might be made easier without the edge effects at polygon boundaries.

4.5 a missing burning model

One of the *interactive* goals not addressed in the fire model presented here is that there is no treatment of the deformation of objects with burning. A more complete work might have made a "tri-partite" hypothesis which included a model for these deformations.

Burning deformations were not considered in this thesis because of the desire for the fire model to exist within the paradigm of contemporary computer graphics. To accurately *burn* a log or a building, more information than just the surface information is required; to model the same in computer graphics, this is not the case.

Paper and other thermally-thin objects could perhaps remain as polygons, however to deform the surface through time one might need to replace a large polygon with many smaller ones. One idea for controlling the deformations is to represent the piece of paper as a two-dimensional finite mesh of springs and masses, and have the process of burning change the spring constants in user-defined ways. Without noise, one might expect curling of the sheet as the flame passes over.

4.6 combinatorics issues

Synthetic char helps in synthesizing a more convincing fire; some other combinatorial elements are glow, smoke, and disconnected methods of spread.

Glow can be implemented by making the color of the char particles vary with time. If the char begins as red and fades to translucent black like the old char model it would give the illusion of glow.

A smoke model (see [2] for example) can likewise be affixed to the object at the point of flame termination. As with the flame model, however, the desire for easy world-space insertion and rendering supports a particle-based model for smoke; alternatives to the flame model presented here could likely be used to produce decent smoke. Fuzzy particles colored to represent smoke, for instance, would exist in the same wind field as the flames.

In the current implementation, fires will spread only over connected objects. It would be somewhat time consuming but dramatically realistic to calculate intersections between the spline backbones of each flame and other polygonal objects in the scene, and begin a new fire on these objects accordingly. Such a form of disconnected spread would allow a fire to spread properly through a forest made of individual, polygonal trees.

4.7 conclusion

The model presented in this thesis takes a step towards synthesizing realistic fires by culling and uniting the perceptually relevant aspects of existing models from physics and graphics.

The four C's (convincing, controllable, contemporary, and computable) were identified and followed to best assure that the model would be practical for a large number of tasks. To summarize, the fire model presented in this thesis is:

Convincing: Particle system flames are inherently three-dimensional, and the spreading model is defined on the surface of three-dimensional objects. Flames exist and are effected by the external wind field $\vec{u}_\infty(\vec{p}, t)$, and they can radiate light to the environment. Fires burn on and char the surfaces of objects, and show preferential spreading in the concurrent over the opposed directions because of the physically-based spreading model. Without a quantitative measure of "visual realism," we have solicited the qualitative comments of peers on various synthesis examples and the response has been very positive.

Contemporary: Flames are rendered as translucent, Gouraud shaded triangles, and spreading occurs on polygonally-defined objects.

Controllable: The external wind field $\vec{u}_\infty(\vec{p}, t)$ allows for the variation of gravity, external large- and small-scale winds, and the movement of objects. The wind field can be arbitrarily defined to provide unique flame shapes and motions, and the colors of the flames can be chosen at will. Objects can have different burning parameters affiliated with them such as flammability and thickness of fuel. Model scalability (see Section 3.5) allows for a matchstick model to be burned like it was the size of a tree and vice versa.

Computable: The flame and the spreading model taken alone run in real time on an SGI Indigo2 Extreme. By reducing the flame model to a simple polyline, the combined fire model runs between 2 and 10 Hz. Frames rendered with the complete model reduce this to approximately 1 Hz.

Fire was chosen because of its complexity and ubiquity, and because of both, this thesis has in no way closed the book on the problem. I hope that the work presented here can be built upon in future research.

> Prometheus: I caused mortals to cease forseeing doom.
>
> Chorus: What cure did you provide them with against that sickness?
>
> Prometheus: I placed in them blind hopes ... I also gave them fire.

Bibliography

[1] J. A. Barnard and J. N. Bradley. *Flame and Combustion.* Chapman and Hall, second edition, 1985.

[2] D. S. Ebert, F. K. Musgrave, D. Peachey, and K. Perlin. Procedural modeling and rendering techniques, 1993. Siggraph 93 Course 44 Notes.

[3] M. Faraday. The chemical history of a candle. In Charles W. Eliot, editor, *The Harvard Classics*, volume 30, pages 89–178. P. F. Collier and Son, 1910. Six lectures on the chemical history of a candle.

[4] A. C. Fernandez-Pello. Controlling mechanisms of flame spread. *Combustion Science and Technology*, 32:1–31, 1983.

[5] A. C. Fernandez-Pello. Flame spread modeling. *Combustion Science and Technology*, 39:119–134, 1984.

[6] A. G. Gaydon and H. G. Wolfhard. *Flames: Their Structure, Radiation, and Temperature.* Chapman and Hall, Limited, second edition, 1960.

[7] Andrew S. Glassner, editor. *Graphics Gems.* Academic Press Professional, 1990.

[8] Dan Hastings, 1993. e-mail communication with Rosalind Picard, forwarded.

[9] Masa Inakage. A simple model of flames. In *Proceedings of CG International*, pages 71–81. Springer-Verlag, 1990.

[10] Silicon Graphics Inc. Jello demonstration program.

[11] VCE Inc. *Pyromania! Playing with fire.* P.O. Box 921226, Sylmar, Ca. 91392-1226, 1993. Movie Effects Studio Volume 1 CD-ROM.

[12] Christos H. Papadimitriou Joseph S. B. Mitchell, David M. Mount. The discrete geodesic problem. *SIAM J. Comput.*, 16(4):647–668, August 1987.

[13] Nadia Magnenat-Thalmann and Daniel Thalmann. *Image Synthesis: Theory and Practice*, page 322. Springer-Verlag, 1987.

[14] Henri E. Mitler. Mathematical modeling of enclosure fires. In E. S. Oran and J. P. Boris, editors, *Numerical Approaches to Combustion Modeling*, chapter 23, pages 183–223. American Institute of Aeronautics and Astronautics, Inc., 1991.

[15] David M. Mount. Voronoi diagrams on the surface of a polyhedron. Technical report, Department of Computer Science, University of Maryland, College Park, MD 20742, 1985.

[16] F. Kenton Musgrave, 1994. e-mail communication, February 9.

[17] Darwyn Peachey, 1993. e-mail communication, October 22.

[18] K. H. Perlin. An image synthesizer. *Computer Graphics Proceedings, Annual Conference Series*, 19(3):287–296, 1985.

[19] K. H. Perlin and E. M. Hoffert. Hypertexture. *Computer Graphics Proceedings, Annual Conference Series*, 23(3):253–262, 1989.

[20] Rosalind Picard. Structured patterns from random fields. In *Proceedings of the 26th Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, October 1992.

[21] W. T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *ACM Transactions on Graphics*, 2(2), April 1983.

[22] W. T. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Computer Graphics Proceedings, Annual Conference Series*, 19(3):313–322, 1985.

[23] G. Sakas and R. Westermann. A functional approach to the visual simulation of gaseous turbulence. *Eurographics*, 11(3):c107–c117, 1992.

[24] K. Sims. Burning letters fire simulation, August 1990.

[25] K. Sims. Particle animation and rendering using data parallel computation. *Computer Graphics Proceedings, Annual Conference Series*, 24(4):405–413, August 1990.

[26] M. D. Smooke. Numerical modeling of laminar diffusion flames. In E. S. Oran and J. P. Boris, editors, *Numerical Approaches to Combustion Modeling*, chapter 7, pages 183–223. American Institute of Aeronautics and Astronautics, Inc., 1991.

[27] J. Stam and E. Fiume. Turbulent wind fields for gaseous phenomena. *Computer Graphics Proceedings, Annual Conference Series*, 1993.

[28] R. F. Voss. Random fractal forgeries. In R. A. Earnshaw, editor, *Fundamental Algorithms for Computer Graphics*, volume F17 of *NATO ASI Series*. Springer-Verlag, 1985.

[29] Alan Watt and Mark Watt. *Advanced Animation and Rendering Techniques: Theory and Practice*. ACM Press and Addison Wesley, 1992.

[30] F. A. Williams. Mechanisms of fire spread. In *Sixteenth Symposium (Int.) on Combustion, MIT*, pages 1281–1293. Combustion Institute, Pittsburg Penna., 1976.

[31] Andrew Witkin and Michael Kass. Reaction-diffusion textures. *Computer Graphics Proceedings, Annual Conference Series*, 25(4):299–308, 1991.