

Simulating Turbulent Combustion

Daniel Barrero, Mathias Paulin, René Caubet
Institut de Recherche en Informatique de Toulouse (IRIT)
Toulouse, France

Abstract

This paper presents a system for modeling the turbulent behavior that occurs when a burning gas interacts with solid objects and the surrounding medium (often seen as a flame). More specifically we try to model the rotational motion produced by the mixture of burned and unburned gases on the flame front in addition to the reciprocal effects of the combustion and turbulence phenomena. The technique is based on the utilization of a hierarchical model of turbulence fields combined with the use of volumetric multi-grid finite differences methods and a simplified flamelet model to obtain a numerical approximation of the fluid's governing equations for turbulence and combustion phenomena.

Keywords: *Simulation, Physically Based Modeling, Animation, Fire, Combustion, Turbulence.*

1. INTRODUCTION

Fire is one of the most intriguing and attractive phenomenon present in nature. Although it is present in a day to day basis it is extremely difficult and expensive to control by the use of pyrotechnic techniques, and even then the range of effects that can be achieved remains quite limited. Naturally, as a result, the synthetic synthesis of fires which is a visible manifestation of turbulent combustion phenomena becomes an attractive goal not only for the budget conscious producer or special effects specialist, but also to the computer graphics expert, games programmer or an animator that simply wants to incorporate the simulation of realistic natural phenomena to improve his work.

One of the problems most commonly found with the creation of synthetic fires is that the particle systems [18][20], stochastic methods, and noise based techniques [9][16] most commonly used are mostly oriented to the sole purpose of the rendering of the flames and doesn't do a good work of representing the interaction of the burning gas with it's environment, and the ones that try to do it are for the most part oversimplified models of fire spread [22][18] with some exceptions only[7][8].

One of the reasons for this lack of physical accuracy in the models, is that due to the inherent complexity, interdisciplinary character, difficulty of control and wide range of manifestations of the turbulence combustion phenomenon, it is quite difficult to obtain a clear physics representation of it, even though it is a very active field of research by thermodynamics experts. On the other hand, if we limit ourselves to one subset of turbulent combustion phenomena, namely the special case of premixed turbulent flames, it is possible to take advantage of all the research work in that domain to build a suitable physics based model for a wide range of combustion effects useful in computer graphics.

In this paper we present a technique for the modeling and simulation of turbulent combustion and flames, using a physically-based flamelet (CFM)[15] model for the combustion part, in combination with a turbulence model composed of a fast volumetric finite differences multi-resolution method combined with a hierarchical model of turbulence, which is an enhanced version of our physics based multi-resolution model for turbulent gases presented in [1].

2. PREVIOUS WORK

Previous works in the area of computer graphics, have been mainly oriented to the specific rendering of flames, with some exceptions that really try to incorporate a model for fire and even then for the most part of them they only deal with the fire spreading part. The most popular techniques for flame rendering are the noise based ones and the ones that use particle systems among others.

The noise based approach initially proposed by Perlin[16], is based on the use of a fractal approximation of turbulence to perturb a color/temperature ramp in 2d. It was later improved [17] to 3d in combination with hypertextures. This approach makes quite difficult to control and model the interactions with the surrounding environment requiring a lot of trial and error to obtain the desired effect, due to the lack of a real fire model

Particle systems techniques[20] are the most popular ones for the rendering of flames since they are very efficient, simple to implement and allow for a common representation of flames, smoke and other types of gases and effects. Put aside the fact that really large number of particles are needed to hide the discrete nature of the technique[18][20], the biggest drawback of this method is, that a real gas is a continuous medium and selecting only regions of it can cause unpredictable results without a lot of hand tweaking, due to the missing model of a gas mixing with its surroundings or a real combustion model. To better reflect the rotational motion of gas turbulence some modifications have been proposed to particle systems using instead point-vortex particles[6] but they require higher calculation times or are limited to 2D.

Another approach is the utilization of stochastic methods in combination with particle systems or the utilization of a randomly perturbed vector fields in combination with deterministic fields as proposed by Stam [23] to model turbulence, who also proposed a modification to this method[22] by adding a perturbed intensity field to simulate the flame spreading over a surface, and uses a warped blob method to render the flames.

In fact, most of the approaches to create a fire model rather than a flame rendering only one, are models of fire spreading to enhance a flame rendering approach, ranging from simpler heuristic two level particle systems for the spread of fire [20],

and passing through the stochastic ones[22] to more physically based ones [7][8][18], and even then all these models assume that turbulence doesn't play any role in turbulent combustion, and/or are based on the Arrhenius approach which can be probed false from a mathematical and thermodynamical point of view. It is worth to notice that several physical based models that deal only with fluid flow or turbulence do exist [1][6][10] [21][25][26].

Now, outside of the domain of computer graphics. From the physical point of view, the phenomena of turbulent combustion has been under heavy study for a long time, and a lot of books and publications are dedicated solely to the research of this phenomena[27][14]. despite all the work done in this area the problem is still considered an open one due to the inherent complexity and interdisciplinary character of the phenomenon. Nevertheless, we can take advantage of all the existent work on this area as a basis for our model.

3. UNDERSTANDING TURBULENT COMBUSTION

Basically, the turbulent combustion phenomena can be divided into three big sub-problems, from a physical point of view:

- The modeling of turbulence, which by itself is a quite complex phenomena but it can be represented by means of the Navier-Stokes (N-S) equations and their extended forms which are considered a good model of fluid flow, and we can take advantage of previous works in computer graphics that solve these set of equations in reasonable times[1][10][11][21][25][26].
- The specific model of combustion without turbulence, which is also a quite complex process involving many chemical reactions in very short times. Here there isn't so much of a consensus, as a result there are models ranging from the simpler 'no-model' Arrhenius approach to more complex ones like the EBU[14],PDF[27], or flamelet models[15][19].
- The modeling of the interaction between combustion and a turbulent flow. This one can be divided into two set of problems: the effects of turbulence on the flame front (often taken into account by the more complex combustion models), and the problem of flame generated turbulence.

In fact our turbulent combustion model is based on a modified flamelet model called the Coherent Flame Model (CFM)[15] as explained in this paper. For the turbulence part we use our real time model of turbulence which uses a combination of a hierarchical decomposition of turbulence, a fast volumetric finite differences multi-resolution technique, and a wavelet model for very low scales of turbulence. This turbulence model is only going to be described briefly here, a better explanation of it can be found at [1].

3.1 Turbulence

From the point of view of fluid dynamics, a fluid's turbulent behavior can be represented by means of the Navier-Stokes set of differential equations [24], which represent the fluid in terms of its velocity (\mathbf{u}), pressure (p), density (ρ), and temperature (T), and internal energy E in the space-time continuum.

Here we are going to use a compressible representation form of the Navier-Stokes equations meaning that not only compression effects can be modeled, but also that the equations can be solved much faster [5][26]. In the vector form the equation set for the velocity of a hot turbulent gas is:

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \frac{\nu}{3}\nabla \cdot (\nabla\mathbf{u}) - \beta\mathbf{g}(T_0 - T_k) + \mathbf{F} \quad (1)$$

With \mathbf{F} =external forces applied to the fluid, ∇ =gradient operator, ν =kinematic viscosity which is directly related to the gas's Reynolds number ($Re=\mathbf{u}/\nu$, l =characteristic flow size) that is traditionally used to represent the quantity of turbulence present in a fluid, \mathbf{g} =gravity, β =thermal expansion coefficient, T_0 =reference temperature, and T_k =average temperature of a gaseous region.

This equation (1) models how the velocity of a gas changes depending on convection ($(\mathbf{u} \cdot \nabla)\mathbf{u}$), its pressure gradient (∇p), and drag ($\nu\nabla^2\mathbf{u}$). It also models the convective and rotational velocity of the gas. The fourth term ($\beta\mathbf{g}(T_0 - T_k)$) models the thermal buoyancy of the gas.

Conservation of mass is expressed by the fact that changes of density in a parcel of gas must be equal to the net flux across the parcel's boundary:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho\mathbf{u}) \quad (2)$$

The effects of convection, diffusion and turbulence of the gas produced by heat can be described by the Navier-Stokes system of equations for heat as follows:

$$\frac{\partial T}{\partial t} = \lambda\nabla \cdot (\nabla T) - \nabla \cdot T\mathbf{u} \quad (3)$$

Where λ represents turbulent and molecular diffusion, $\nabla \cdot T\mathbf{u}$ =changes due to convection, and $\lambda\nabla \cdot (\nabla T)$ =temperature changes due to diffusion and turbulent mixing.

In addition to the system of equations (1)(2)(3) we need the equations of a gas state that determine the relationship between energy, pressure, density and temperature:

$$E = c_p T, \quad p = \rho RT \quad (4)$$

c_p is the gas's specific heat, and R is the gas constant.

Also, it's very important to notice that turbulence is a multi-scale phenomenon, that means that turbulent fluctuations are associated to different scales ranging from the largest (l or global fluid flow) to the smallest turbulent one (η , Kolmogorov scale). Thus it becomes an important issue to know how the turbulence energy is distributed over the different scales present in the flow.

In fact, larger eddies transmit their energy to smaller eddies through non linear terms in what is known as the Kolmogorov energy cascade, which induces the creation of an inertial zone in spectral space where the energy E depends only on the energy flux ϵ and the wave number K , the most simple example of a Kolmogorov energy cascade spectrum is:

$$E(K) = C_k \epsilon^{2/3} K^{-5/3}, \quad C_k \cong 1.5 \quad (5)$$

It is also quite common to use the Von Karman-Pao spectra [24] for a more accurate model of the energy cascade.

We are going to take advantage of this multi-scale characteristic of the turbulence phenomena to simplify the resolution of the Navier-Stokes set of equations.

3.2 Flame Generated Turbulence

Many of the models of turbulent combustion are concerned with the effects of turbulence on the flame front. However, the flame front also modifies the turbulence, leading to the phenomenon of flame generated turbulence. This phenomena has two important characteristics:

First, when temperature changes from one side of the flame front to the other, kinematic viscosity and therefore the local Reynolds number of the gas changes accordingly, which means that a hot turbulent gas can become laminar when ignited, or the other way around, i.e. for air the kinematic viscosity for air is $T^{1.76}$, so for a flame with a temperature ratio of $T_2/T_1=8$, the Reynolds number in the burnt gases will be 40 times smaller than in the fresh gases.

The second important effect of the flame on the flow is the acceleration through the flame front. Which is caused by the fact that density changes through the flame front are directly related to the temperature ones through the flame front.

3.3 Combustion

We had chosen to use the coherent flame model (CFM) [15] based on the flamelet model[19], not only because it is one of the most accurate, but also because it integrates nicely with our turbulence model. The basic flamelet model assumes that the flame front is a continuous sheet separating fresh and burnt gases (see Figure 1): it allows us to define the behavior of a turbulent combustion phenomenon but it needs to be complemented with some evaluation method of the flame surface.

The main advantage of the CFM model over the basic flamelet model is that it is based on a simple and intuitive description of the turbulent flame and that it allows us some complex features to be taken into account without additional complexity. Instead of trying to define turbulent combustion in terms of one-point quantities like the crossing frequencies of flamelets, the CFM model considers the flame surface as the important quantity controlling the reacting flow and uses it as the parameter to model.

As a result, the turbulent flame is modeled as an ensemble of flamelets (small laminar flames). It is then a simple matter to evaluate the mean reaction rate per unit volume $\dot{\bar{w}}$ as the product of the flame surface density Σ (m^2/m^3) by the local consumption rate per unit flame surface w_L :

$$\dot{\bar{w}} = w_L \Sigma \quad (6)$$

w_L can be computed using a simple model of laminar planar stagnation point flame[13]. Σ can be obtained by using the following equation:

$$\frac{\partial \bar{\rho}\Sigma}{\partial t} = -\frac{\partial \bar{\rho}\tilde{u}_i \Sigma}{\partial x_i} + \frac{\partial}{\partial x_i} \left(\frac{\nu}{\sigma_\Sigma} \frac{\partial \Sigma}{\partial x_i} \right) + \bar{\rho} E_s \Sigma - w_L \frac{\Sigma^2}{Y} \quad (7)$$

Where: σ_Σ is a constant, E_s is the flame stretch generated by the turbulent flow ($E_s = \epsilon/k$), and Y is the non dimensional mass fraction of the reactants: $Y = Y_R/Y_R^0$, Y_R^0 is the initial mass fraction for the fresh gases, Y varies from 1 in the fresh gases to 0 in burnt gases.

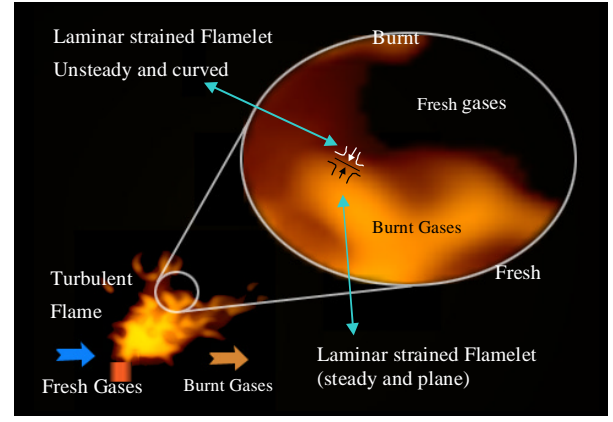


Figure 1: The Flamelet Regime.

Equation (7) is derived from an extension of the balance equation for material surfaces in turbulence[15] and it represents the turbulent dissipation of energy for turbulent reacting flows. More exactly it models the flame surface density Σ changes because of flamelet convection (first term on RHS), the flamelet turbulent diffusion (second term), the stretching of the flame front due to turbulence (third term), and the flame annihilation and shortening (last term).

As we can see, equation (7) is of similar form to equations (1) to (3), thus we can solve it by using similar methods as the ones used for the turbulence equation set.

In fact, the equations (6) and (7) are meant to be used in combination with an extended version of equations (1) to (3). In the first place we extend equation (3) to take into account the effects of the flame reaction on temperature:

$$\frac{\partial T}{\partial t} = \frac{\lambda}{\rho C_p} \nabla \cdot (\nabla T) - \nabla \cdot T\mathbf{u} - \frac{p}{\rho C_p} \nabla \mathbf{u} + \frac{Q}{\rho C_p} \dot{\bar{w}} \quad (8)$$

where Q designates the heat of reaction for per unit mass of fresh mixture $Q = -\Delta h_f^0 Y_R^0$ (Δh_f^0 is the heat of reaction per unit mass).

In the second place, we extend the system of equations with a model for the evolution of the mass fraction of the reactants Y :

$$\frac{\partial \rho Y}{\partial t} = \delta \nabla \cdot (\rho \nabla Y) - \nabla \cdot (\rho Y \mathbf{u}) - \dot{\bar{w}} \quad (9)$$

It is important to notice that the chemistry and turbulence have been treated independently here, the term w_L contains the laminar flame consumption rate but is the only chemical parameter appearing in equations (6) and (7).

4. THE SIMULATION MODEL

The full system of equations (1),(2),(4)-(9), presented in the previous section constitutes a full model for turbulent combustion. To solve this system of equations we use a modified version of our multi-resolution turbulence model[1], which basically uses a hierarchy of turbulence fields in combination with a special multi-grid finite volume method to obtain a numerical solution to the equation system.

The method consists basically of three main parts (Figure 2): a turbulence basis set, a turbulence hierarchy and a multi-resolution voxel grid which is used to model the simulation environment and where the numerical resolution of the equations actually takes place.

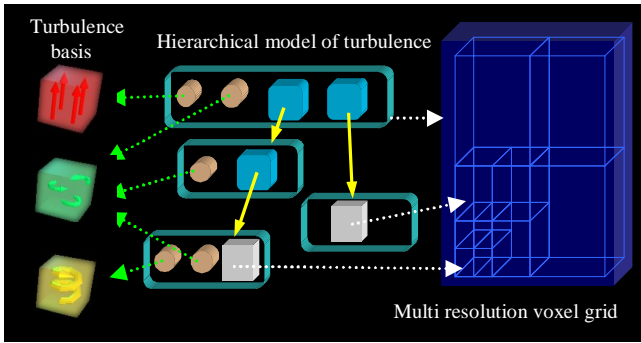


Figure 2: Relationship between the different parts of the hierarchical turbulence combustion model

The turbulence basis and turbulence hierarchy are based on the idea that turbulence phenomena can be expressed as a multi-scale phenomenon, and that some identifiable kinds of turbulence have a tendency to appear in a repetitive way at all scales (i.e. vortexes). As a result it is possible to express complex turbulence at a given scale as a composition of the simpler turbulence forms at lower scale and a set of lower level complex ones and so on in a recursive way, the relationship between the scales is expressed by the Kolmogorov energy cascade as explained before.

The turbulence basis keeps a list of known turbulence behaviors (i.e. vortexes, sinks, or more complex physical or wavelet models) evaluated in parametric space, and allows for an animator to obtain a specific special effect by using these to control the fluid flow.

The turbulence hierarchy in reality can be seen as a model of the Kolmogorov energy cascade, each of the level represents a turbulence scale, and each node corresponds to: A set of large scale fields in a level which are a set of deterministic fields from the turbulence basis at that given scale, and the corresponding level of the multi-resolution grid. This is possible by the consideration of each of the voxels of the grid corresponding to this level as a special large scale turbulence field, meaning that it can contain lower scales of turbulence in it, following the hierarchical definition of turbulence. Furthermore, these lower levels are not necessarily of the same type. This allows us to combine the finite elements and wavelet methods in a seamless manner. All this can be represented more clearly in a graphical way as shown by the Figure 2.

To model turbulent combustion the turbulence basis and turbulence hierarchy models are virtually left untouched, only the multi-grid part of the model has been improved to take into account the special characteristics of turbulent combustion as explained in the next section.

4.1 Solving the Equation Set

The actual resolution of the system of equations is done by using the multi resolution voxel grid, in combination with the turbulence hierarchy to account for the energy exchange between the different scales of turbulence.

In the first place to model the simulation environment we use an octree like approach (Figure 3), this allows for a straight forward integration of the interactions between the gas and the solid objects present on the scene as showed by Foster[10][11][12] since we can use these representation as boundary conditions for our simulation space.

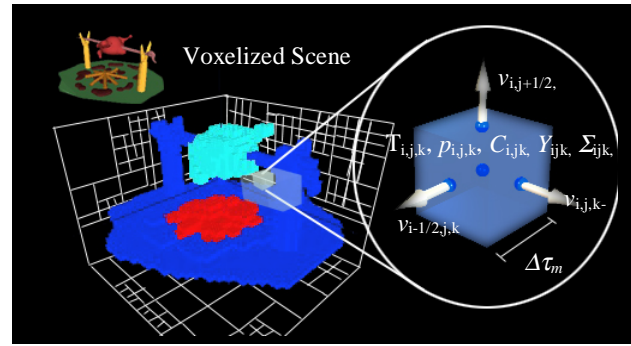


Figure 3: Scene represented by a multi-resolution grid

This representation of the simulation scene as an octree is also used to define the initial discretization size for the gas space and as an initial minimum height for the energy cascade.

Now for each cell on the grid, we associate the different gas properties like temperature, temperature $T_{i,j,k}$, pressure p , density ρ , viscosity ν , mass fraction of reactants Y , fuel quantity C (which can be defined for each object via a fuel map), etc... Likewise in the center of each face of a cell, we define the gas velocity perpendicular to that face: this leads to the velocities shown in the Figure 3. To solve the system of equations (1),(2),(7)-(9), we have to rewrite them in an appropriate form that is applicable to the voxel grid, using a numerical method known as finite centered differences[5][10]. For example: a differential term such as $\partial T/\partial y$ at voxel $[i,j,k]$ can be approximated using finite centered Taylor series as:

$$\frac{\partial T}{\partial y} = \frac{T_{i,j+1,k} - T_{i,j-1,k}}{2\Delta y} - O(\Delta y^2) \quad (10)$$

Where $O(\Delta y^2)$ is the error due to high order terms. A good discussion of the method to rewrite the equations can be found on [5] [10][11][12] and is not going to be repeated here.

Once all the differential equations (1)(2)(7)-(9) are expressed in terms of variables using finite differences they can be used directly as the update rules for an explicit integration scheme like fourth order Runge-Kutta.

However, rapid changes in pressure created by steep pressure and temperature gradients could cause the system to diverge rapidly, which is quite often the case for turbulent combustion. Also given that the rewritten set of equations is for an uniform one level grid, the lowest level of turbulence that can be modeled are the ones that are bigger than the grid size, requiring very dense grids to be able to represent high turbulent flows.

We overcome these problems in part by adapting the finite differences method to the multi-resolution grid. This can be easily accomplished by applying the method for regular grids to each one of the sublevels of the grid (which are themselves regular grids) from the biggest level to the lower resolution ones, using the obtained intermediate results and the Kolmogorov energy cascade to set the initial boundary conditions and energy transfer between the turbulence scales. This propagation can be easily accomplished by using the turbulence tree. From the grid point of view this is like using a donor-acceptor method[5].

To know if it is necessary to evaluate for lower scales of turbulence and as a result create new grids, we analyze the state of a grid/voxel on a given level for energy peaks, and steep

changes, and if it doesn't have lower scale grids associated with it and it's at a level higher than η (Kolmogorov scale) a new lower scale grid is created.

For the case of sharp gradients on fluid density, we needed to use a relaxation adjustment if the divergence field error on lower scales was too large to account for changes of mass, fortunately this is not necessary anymore for turbulent combustion since the mass changes are being modeled by the evolution of the mass fraction for the reactants.

4.1.1 Autosimilar form of the Navier-Stokes equations

We use the autosimilar form of the Navier-Stokes set of equations in a wavelet space [1][4] to approximate the scales of turbulence lower than the smallest cell on the grid hierarchy instead of using simple linear interpolation.

Although this formulation of the Navier-Stokes equations requires the initial value of vorticity $w_0(x)$ to be small in a measure space of type Morrey-Campanato[1][4] and some homogeneity conditions, it can be probed that using wavelet analysis it is possible to obtain a solution without having a completely adapted space[1][4] which is the case of the smallest cells on the grid that fulfill the homogeneity conditions for the most part, but this is out of the scope of this article and a good discussion of it can be found on [1] and [4]. With this model, the equation (1) for the velocity becomes:

$$\gamma_u(r) = E(\mathbf{u}(r_0 + r)\mathbf{u}(r_0)) \sim \varepsilon^{-2/3} r^{2/3} \quad (11)$$

which is the dual expression of the spectral density for the power of the velocity field dependence in $\varepsilon^{-2/3} r^{2/3}$. With this formulation the time-space variables are dynamically linked, as a result the velocity field presents the following autosimilar relationship:

$$\mathbf{u}(\lambda r, \lambda^{1-H_v}) \longrightarrow \lambda^H \mathbf{u}(r, t) \quad (12)$$

which is pretty similar to the one for the fBm, but the analogy with the fBm motion is not complete because the velocity field is not a gaussian process.

One of the main advantages of using this approach is that we are able to obtain an approximation for the turbulence scales lower than the minimum cell size. Also it allows for the utilization of coarser grids and smaller hierarchies, speeding up the simulation process as a result.

4.2 Following the Flame Front

To model the flame generated turbulence is only a matter of modifying the viscosity and density of each one of cells according to the temperature changes across the flame front, the problem is that we need to know the flame surface. Also we need it to be able to cope with the CFM model and it simplifies the evaluation of the local consumption rate per unit flame surface w_L .

Fortunately this problem is very similar to the tracking of fluid's surface. In fact, so similar that, we use an approach based on the one proposed by Foster[11][12] for tracking a fluid surface with minimal modifications:

The approach consists in identifying each cell as a gas, free, or static cell (solid object), and in keeping series of tracking particles at each of the cells containing a gas or in the surface. Each of these particles will be updated according to its relative position to the flame front.

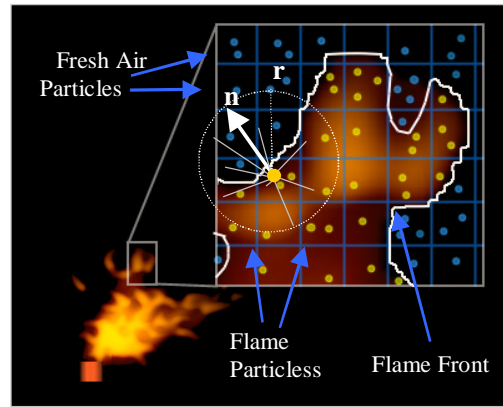


Figure 4: Following the flame front.

Initially, all particles are going to be marked as fresh air particles. Then, following the resolution of the combustion model for that particle position in a cell they get to be marked as flame particles or not (based on the dimensional mass fraction of the reactants Y).

To find the flame front, we must analyze each of the particles on a gas cell adjacent to one or more free cells in comparison to the particles on the free cells (Figure 4), to obtain an implicit representation of the flame front's surface.

This approach not only will simplify the application of the flamelet model but also suggests a direct method for visualizing the flame surface.

4.3 Interaction With Objects

The interaction with objects on the scene is taken into account automatically since we use the octree-representation of the scene as boundary conditions for the simulation. As explained before for each cell we define the fuel quantity and the mass of reactants, this is done for each cell of the simulation. For the specific case of the cells belonging to a solid object we give the values corresponding to the object's burning properties like specific heat, fuel quantity and reactants mass based on the objects inflammability. It is possible to define a fuel map on the surface of an object in a similar way as applying a texture and use this fuel map in combination with the objects properties to initialize the cell values. Note that we don't model the changes of geometry caused by the burning process.

As an example in the Figure 3 the voxels corresponding to the chicken are possible smoke sources, and the ones belonging to the logs and coal are identified as flammable.

4.4 Flame Generated Smoke

One of the most identifiable secondary effects of turbulent combustion is the generation of smoke as some burnt gas particles of the flame start to cool down. Unfortunately to our knowledge no definite analytical models of this phenomena exists.

To model the creation of smoke we use the set of particles used to track the flame front as the sources for the smoke density using a method similar to the one proposed by Stam in [22]:

If for a given particle the mass fraction of reactants Y is between a given threshold and is below a given temperature T_s , this particle can be used to define a smoke source for the time it continues to fill in the given conditions, the source term for the

smoke density is then a function of the Particle Temperature and density of that flame particle ρ_{flame} (which is directly related to the mass of the reactants per unit of volume):

$$S_{\rho, \text{smoke}} = v_s \exp\left(-\frac{T_{\text{flame}}}{T_s}\right) \rho_{\text{flame}} \quad (13)$$

4.5 The Turbulent Combustion Algorithm

The algorithm for calculating the behavior of hot turbulent gases and combustion is composed of two stages. Given the physics-based nature of the method, it resembles the first preparation steps in a fluid dynamics program, but simpler in many ways.

4.5.1 Preparation Stage

The preparation work that an animator must do can be divided in seven steps. Some of them are made first in an automatic way, and can be fine hand tuned in a later step:

1. Set the subdivision range to decompose the environment in a coarse voxel grid, with a biggest size length between $\Delta\tau$ and $\Delta\tau_m$, which will be used to subdivide the solid objects.
2. Set the different fluids space of influence within the simulation environment.
3. Set the subdivision parameters of the environment into a coarse voxel grid to better suit the animator's requirements. An initial automatic subdivision is done based on the objects and fluids positions using the size length limitations.
4. Set boundary conditions for velocity and temperature in the grid, that is, place all the heat sources and sinks, combustion sources, fuel cells, and assign all the fuel maps to the objects
5. Create all the deterministic fields to control the fluid flow.
6. For each different fluid in the simulation:
 - 6.1. Consider the different fluids physical properties: viscosity, thermal expansion, molecular diffusion, Reynolds number, etc.
 - 6.2. Set the different fluids maximum space of influence and place them in the environment
7. Determine Δt from the minimum 1/30 of a second and the largest stable time step.

4.5.2 Automatic Simulation Stage

Once the physical properties of the different objects and gases composing the scene are set the user simply strikes a match by indicating the origin(s) of the combustion and then the automatic part of the simulation starts. The simulation algorithm can be summarized as described by the pseudo-code shown in Figure 5.

5. RENDERING

Many approaches for the rendering of gaseous phenomena have been presented in recent years and are not going to be explained here. Basically, we use two methods:

To help the animator to set up the simulation environment and see the output in real time. We use a simple system of mass-less particles to render the smoke, and for the flame we use the evaluated flame front surface in combination with the system of particles used to track it. This is to give it a more realistic fuzzy look since the surface of a flame is in reality an implicit surface of varying thickness and not a thin layer like the simple flame front polygonal surface could imply.

```

For each time frame do
  Apply boundary conditions to the objects
  Update Fuel map/cells
  Actualize boundary conditions
  Generate flame front tracking particles
  Push Energy: Top down energy redistribution
  For each level of the hierarchy
    For each field in the level
      If is deterministic
        Update field
        Transfer energy/heat between fields
      Elself is voxel cell
        Update cell variables using finite differences
        If not subdivided
          Update Wavelet conditions
      Fi
        Find divergence field for the gas to conserve
        mass if demanded
        Apply relaxation adjustment if necessary
    End
  End
  Pull Energy: Down to top free energy redistribution
  Update flame front tracking particles (move and
  advect)
End
  
```

Figure 5: Pseudo-algorithm for the automatic simulation stage.

To produce a high quality output we use hierarchical version of Stam's model of fuzzy blobs highly coupled with our turbulence model for the smoke, and use the set of tracking particles to obtain an implicit representation of the flame.

On either case the contribution of energy from the flame into the environment is equal to the sum of contributions from each of the flamelets composing the flame, which is entirely determined by it's local temperature through blackbody radiation.

6. RESULTS

We have developed an interactive implementation of the above models, and even though we haven't applied any kind of optimization to the algorithms besides the most obvious ones, we are able to simulate turbulent combustion at interactive rates.

To use the system the user specifies the physical properties of each of the objects on the scene and then lights up the fire and the results of the simulation are immediately visualized using OpenGL for the interactive rendering.

All the images were rendered on a PentiumII 450 MHZ PC with a NVIDIA TNT 3D card. The time step for all the simulations is $\Delta t=0.033\text{sec.}$, for all cases the activation temperature T_a is 2300°K and the smoke generation temperature T_s is 1200°K , with an ambient temperature of 290°K . One animation frame is rendered at every 10 simulation steps.

The Image 1 shows two frames at times $t=1$ and $t=6$ for a simulation of turbulent fire over a plane made of a flammable material on a highly turbulent environment ($Re=10000$). It is possible to see the fast spreading of fire over the plane. It takes an average time of 0.1sec. per simulation step.

The Image 2 illustrates the spreading of fire over a complex object. The fire is started at the foots of the wooden figure. The fire spreads faster in the vertical direction with flames lying closely to the object's surface (legs and body), due to the fact that flames spreading on the vertical direction correspond to a concurrent fire spread regime. This simulation takes an average time of 0.2sec. per simulation step.

The image 3 shows a complex scene with fire and smoke interacting with their environment, the octree representation used for this simulation is the one shown on Figure 3, fire is started at the center of the fireplace and it spreads over the logs and coal. Fire is allowed to produce smoke. This simulation takes an average time of 1 sec. per simulation step.

7. CONCLUSION

The main advantage of this turbulence combustion model is the seamless combination of different modeling methods. Which not only allows for the simulation of turbulent combustion at interactive rates (although there's still a lot of room for improvement on the algorithms used). But also simplifies the controllability of the simulation by the combination of physical parameters and the superposition of deterministic fields for extended animator control, without sacrificing the accuracy of the simulation.

Also as the heat transfer between gases and solid objects is done mainly in the grid space, it is possible to simulate the spread of fire on complex objects without additional overhead. And it opens the possibility of a great deal of effects to be developed in the future like changes of the object's geometry caused by the burning process using CSG techniques on the voxel representation of the objects.

A problem with this method is that the grid cell orientation can affect the results, although it is partly solved by the way we evaluate the lower scales of turbulence within each cell. This can be improved by using non-structured unaligned grids.

Even though our goal was to create a physically realistic model of turbulent combustion, we made a lot of assumptions and oversimplifications. Like the model for the generation of smoke or the one for flame generated turbulence, and left out a lot of other characteristic behaviors of turbulent combustion that didn't contribute notably on the final images; but, that could affect the behavior significantly on certain situations and would require further research.

8. REFERENCES

[1] Abry P., "Ondelettes et Turbulences", Collection Nouveaux Essais, Diderot Editeur, Arts & Sciences, Paris, France, 1997.

[2] Barrero D., Paulin M., Caubert R., "A Physics Based Multi-resolution Model for the Simulation of Turbulent gases and Combustion", Eurographics Workshop on Computer Animation and Simulation (EGCAS'99), Springer-Verlag, Milan, September 1999.

[3] Candel S., Poinot T., "Flame Stretch and the balance equation for the flame area", Combustion Science and Technology, 1990

[4] Cannone M., "Ondelettes, Paraproducts et Navier-Stokes", Collection Nouveaux Essais, Diderot Editeur, Arts & Sciences, Paris, France, 1995.

[5] Collela P., Puckett E.G., "Modern Numerical Methods for Fluid Flow", University of California, 1998

[6] Chen J.X. et al. "Real-Time Fluid Simulation in a Dynamic Virtual Environment", IEEE Computer Graphics and Applications, pp. 52-61, May-June 1997.

[7] Chiba N., Ohkawa S., Muraoka K., Miura M., "Two-dimensional Simulation of Flames, Smoke and the Spread of Fire", J. of Vis. And Comp. Animation, 5, pp.37-54, 1994.

[8] Inakake M., "A Simple Model of Flames", Proceedings of CG International 89, pp 71-81, Springer-Verlag, 1989.

[9] Ebert D.S., Musgrave F.K., Peachey D., Perlin K., Worley S., "Texturing and Modeling a procedural approach", Academic Press Inc., Cambridge MA, 1994.

[10] Foster N., Metaxas D., "Modeling the Motion of a Hot, Turbulent Gas", ACM Computer Graphics, SIGGRAPH 97, Addison Wesley, August 1997.

[11] Foster N., Metaxas D., "Realistic Animation of Liquids", Graphics Models and Image Proc., 58(5), pp 471-483, 1996.

[12] Foster N., Metaxas D., "Controlling Fluid Animation", Proceedings of CG'97, 1997.

[13] Giovangigli V., Smooke M., "Extinction of Strained premixed laminar flames with complex chemistry", Combustion Science and Technology 53, 1987

[14] Kuo K.C. , "Principles of Combustion", John Wiley Intersci., 1986

[15] Marble F.E, Broadwell J., "The coherent flame model for turbulent chemical reactions", Project SQUID, Report TRW-9-PU, 1977.

[16] Perlin K.H. "An Image Synthesizer", Computer Graphics Proceedings, Annual Conference Series, 19(3) pp287-296, 1985.

[17] Perlin K.H., Hoffert E.M. "Hypertexture", Computer Graphics Proceedings, Annual Conference Series, 23(3) pp287-296, 1989.

[18] Perry. C.H., Picard R.W., "Synthesizing Flames and Their Spreading ", Fifth Eurographics Workshop on Animation and Simulation, Oslo, Norway, September 1994.

[19] Pope S, Cheng W, "The stochastic flamelet model of turbulent premixed combustion", Twenty Second Symposium on Combustion, p. 781, The Combustion Institute, 1988.

[20] Reeves et al. "Approximate and Probabilistic Algorithms for Shading and Rendering Particle Systems", ACM Computer Graphics, SIGGRAPH 85, pp.313-322, July 1985.

[21] Stam J., "Stable Fluids", ACM Computer Graphics, SIGGRAPH 99 , Addison Wesley, August 1999.

[22] Stam J., Fiume E., "Depicting Fire and Other Gaseous Phenomena Using Diffusion Processes", ACM Computer Graphics, SIGGRAPH 95, pp.129-136, Addison Wesley, August 1995.

[23] Stam J., Fiume E., "Turbulent Wind Fields for Gaseous Phenomena", ACM Computer Graphics, SIGGRAPH 93, pp 369-373, Addison Wesley, August 1993.

[24] Streeter V.L., "Fluid Mechanics", McGraw Hill, 1995.

[25] Weimer H., Warren J., "Subdivision Schemes for Fluid Flow", ACM Computer Graphics, SIGGRAPH 99, Addison Wesley, August 1999.

[26] Witting P., "Computational Fluid Dynamics in a Traditional Animation Environment", ACM Computer Graphics, SIGGRAPH 99, pp 129-136, Addison Wesley, August 1999.

[27] Williams F.A., "Combustion Theory", 2nd ed., Benjamin Cummings, Menlo Park, 1985.

About the authors

Daniel Barrero is a PhD student at the Image Synthesis Team of the Institut de Recherche en Informatique de Toulouse (IRIT), France

E-mail: barrero@irit.fr

Mathias Paulin (PhD) is a permanent member of the research staff of the Image Synthesis Team of the Institut de Recherche en Informatique de Toulouse (IRIT), France

E-mail: paulin@irit.fr

Professor René Caubet is the director of the Image Synthesis Team of the Institut de Recherche en Informatique de Toulouse (IRIT), France

E-mail: caubet@irit.fr

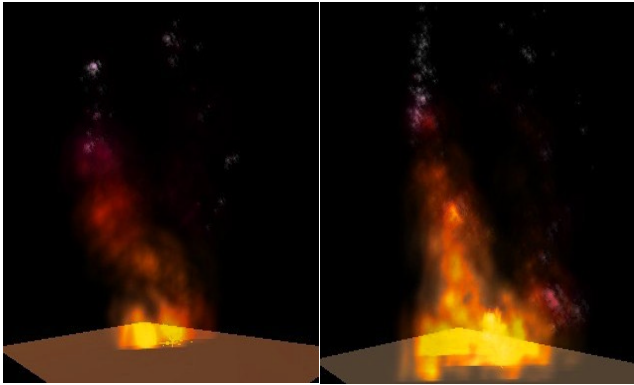


Image 1: Sample of a highly turbulent flame.

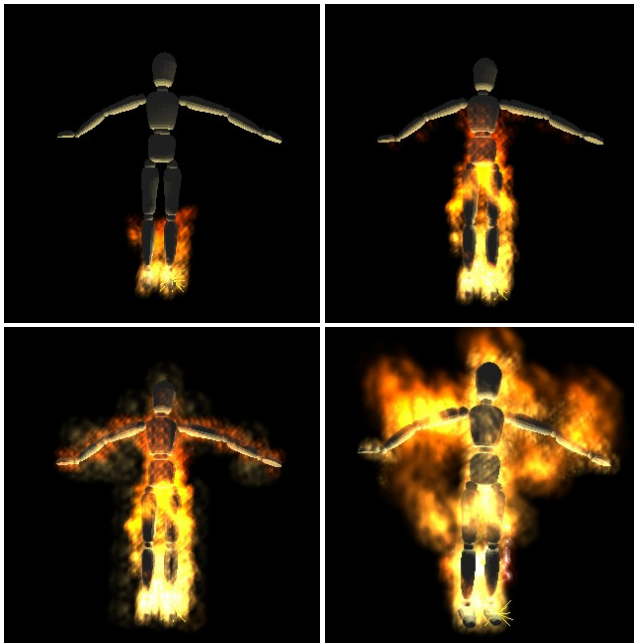


Image 2: Spread of fire over a complex object, at $t=5, 10, 25, 50$ sec. of simulation time.

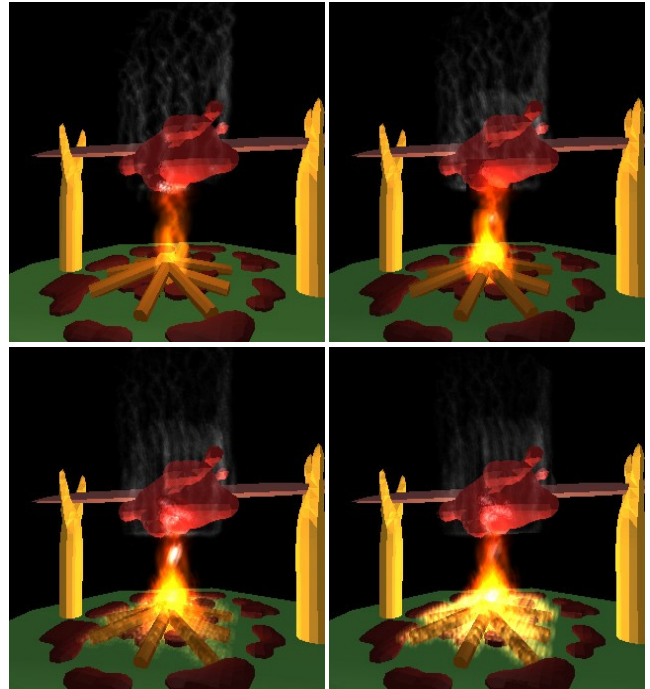


Image 3: Example of flame spreading over complex objects and fire generated smoke interacting with the environment.