

Realistic fire rendering

Garoe Dorta-Perez

CM50170 - Research project

Unit Leader: Dr Darren Cosker

University of Bath

August 2015

Signature of Author

Garoe Dorta-Perez

Contents

1	Introduction	3
2	Previous Work	6
2.1	Simulation	6
2.2	Rendering	9
2.2.1	Raster-Based	10
2.2.2	Ray-Tracing-Based	10
3	Methodology	12
3.1	Radiative Transport Equation	12
3.2	Scattering	14
3.3	Black Body Radiation	14
3.4	Absorption coefficients	14
3.4.1	Soot Absorption	15
3.4.2	Absorption From Other Chemical Species	15
3.5	Refraction	15
3.5.1	Davis' Saturation Vapour Pressure	17
3.5.2	Huang's Saturation Vapour Pressure	17
3.5.3	Ciddor's Method Summary	18
3.6	Visual Adaptation	20

4	Implementation details	21
4.1	Application Overview	21
4.2	Mental Ray [®] Shading Approach	22
4.3	Shaders Internals	23
4.3.1	Fire Volume Shader	26
4.3.2	Fire Light Shader	27
4.3.3	Voxel Dataset Shader	27
5	Results	28
6	Conclusions and Future work	29

Chapter 1

Introduction

“A case that can be made for fire being, next to the life process, the most complex of phenomena to understand.” - Hoyt Hottle



(a) Paper fire [FireImg1, 2015].



(b) Bone fire [FireImg2, 2015].

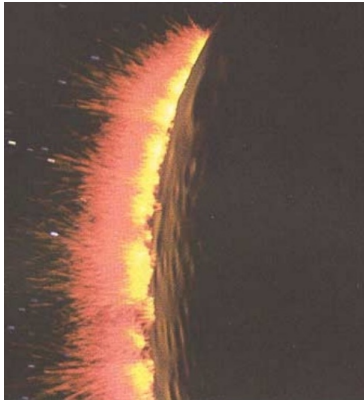
Figure 1-1: Examples of real fire.

Computer simulations are widely used to model a diverse range of natural phenomena. Their popularity lies in their inexpensive nature, predictive capabilities given good mathematical models, parameters can be changed at will to produce new results, the equations governing the system can be modified to create scenarios that are not physically attainable, etc. Computer generated graphics are used to display the results of the simulations. Rendering the data on the screen involves transforming a scene into an image that can be displayed using some light transport and illumination model.

For centuries humans have been attracted to fire due to its attractive presence and its dangerous nature, real fire examples are shown in Figure 1-1. Combustion phenomena are prevalent in daily life, candles, camp fires, explosions, car engines, cooking appliances, etc. Simulating and visualizing fire related processes has

many applications, for example it is widely used for visual effects in the film industry, simulated as part of the virtual environment in the computer games industry; or in the engineering community, where modelling engine combustion and fire safety evaluations are frequently demanded.

Computer generated examples in films include, a planet explosion in Star Trek II, as shown in Figure 1-2a, where a particle-based technique by [Reeves, 1983] was used; Shrek featured a dragon exhaling fire, as shown in Figure 1-2b, where parametric curves were used to drive the flames [Lamorlette and Foster, 2002]; or the more recent work of [Horvath and Geiger, 2009] based on 2D screen projections for the film Harry Potter and the Deathly Hallows, as shown in Figure 1-2c. In these and in many other applications, using real flames is an expensive and hazardous endeavour.



(a) Explosion in Star Trek II [Reeves, 1983].



(b) Dragon fire in Shrek [Lamorlette and Foster, 2002].



(c) Flames used in Harry Potter and the Deathly Hallows [Horvath and Geiger, 2009].

Figure 1-2: Examples of visual effects with fire in the film industry.

The computer graphics community has intensively researched the fluid behaviour of water and smoke. Fire can also be modelled as a fluid, however due to its multiphase flow, chemical reactions and radiative heat transport, the techniques used for water or smoke cannot be directly applied to flames. As a result of the aforementioned complexity and the interdisciplinary nature of the problem, fire simulation is still an open

problem in computer graphics.

A great deal of work done in the area has sacrificed complexity for interactiveness, therefore producing simplified models which hope to deceive the observer by exploiting the chaotic behaviour present in fire motion. Nevertheless, physically-based simulations incorporate the intrinsic processes that occur in a combustion scenario in order to be able to produce realistic results. In general the following effects are taken into consideration

Flame motion: under normal gravity conditions, convection, external wind forces, front propagation and gas expansion drive the shape and position of the flame.

Fuel erosion: when the fuel reaches a certain temperature, it is vaporized into a gaseous state, which rises under the influence of buoyancy.

Flame colour: the chemical species present in the fuel and the byproducts of the combustions emit energy in various wavelengths due to black body radiation.

In this report a short review of fire simulation and rendering methods is presented in Chapter 2. In Chapter 3, a state-of-the-art physically-based model to render fire, proposed by [Pegoraro and Parker, 2006], is discussed extensively. Implementation details of a Mental Ray® shader for the the given model are outlined in Chapter 2, and results are discussed in Chapter 5. Lastly, conclusions and future research directions are considered in Chapter 6.

Chapter 2

Previous Work

In order to display a realistic fire scene in a computer generated world, two differentiated stages are needed. Firstly, the fire dynamics have to be collected, this can either be done through a data capture session or simulated using fluid-based solvers. Present techniques in the area are discussed below. Secondly, the previously gathered data will be visualized on the screen using some rendering method, a brief state-of-the-art review of fire rendering procedures is presented in Section 2.2. We refer the interested readers to the more detailed survey on the topic, which has been recently presented by [Huang et al., 2014].

Fire simulation has significantly received more attention from the computer graphics community than fire rendering. As the output data of the simulation methods is used by the rendering stage, an outline of the different simulation techniques is given below. Still, the main focus of this report lies in the rendering phase of the combustion simulation pipeline.

2.1 Simulation

The techniques presented in this section are divided adopting the same criteria that was used by [Huang et al., 2014]. Since the categories are roughly chronological, the evolution of the field is easier to follow.

Particle-based methods were the first approach to simulate the visual animation of fire, as shown in Figure 2-1. A number of particles are emitted from certain locations, each particle has a set of attributes such as shape, velocity, color or lifetime. The first model with particle systems was presented by [Reeves, 1983], the particles speed and colour were perturbed with a Gaussian noise at each time step, and the colour was subject to an additional linear perturbation on its lifetime. Two particle systems were used in a hierarchy, one would control fire spread and the other a single explosion effect. An extension was proposed by [Perry and Picard, 1994], the authors modified the particle system such that each particle shape would be defined by a series of non-overlapping coplanar triangles. The transparency of would increase towards the outer vertices, thus providing an improved visual effect.

Noise-based methods focus on synthesizing the high fluctuation present in fire procedurally, as shown in Figure 2-2. The objective is to approximate the turbulence present in fire with an appropriate statistical

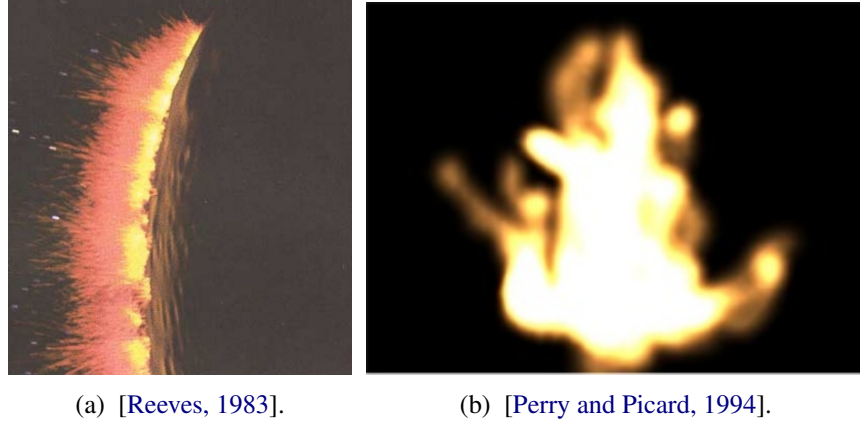


Figure 2-1: Particle-based fire examples.

model. Using a variation of Perlin noise, [Perlin, 1985] presented images of a corona of flames. However, the method is limited to 2D, where the color is a combination of non-linear arbitrary functions. This work was extended by [Perlin and Hoffert, 1989] to 3D, where they use volumetric rendering to achieve improved results.

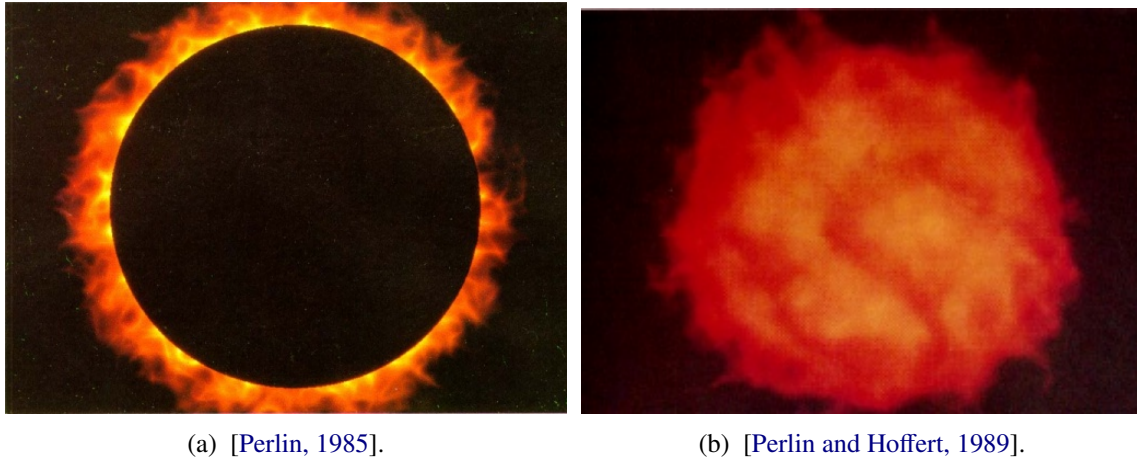


Figure 2-2: Noise-based fire examples.

Geometry skeleton are driven by rendering primitives that represent the fire main features, the high frequency details are built on top of the geometry structure, as shown in Figure 2-3. [Lee et al., 2001] presented a technique to animate fire on meshes, the flame front propagation is based on a geodesic flow field on the surface. Merging and separating of multiple fronts is achieved, however there is not control for animators over the fire evolution. [Lamorlette and Foster, 2002] proposed a B-Spline curves interpolation method, details are added from a library of images which are then cast onto the profile of the flame. The authors' technique supports a range of flame behaviour, such as spreading, flickering, and flame separation and merging. In order to achieve real time simulations, [Bridault-Louchez et al., 2006] modelled the flame shape on NURBS surfaces on which a transparent 2D texture were mapped.

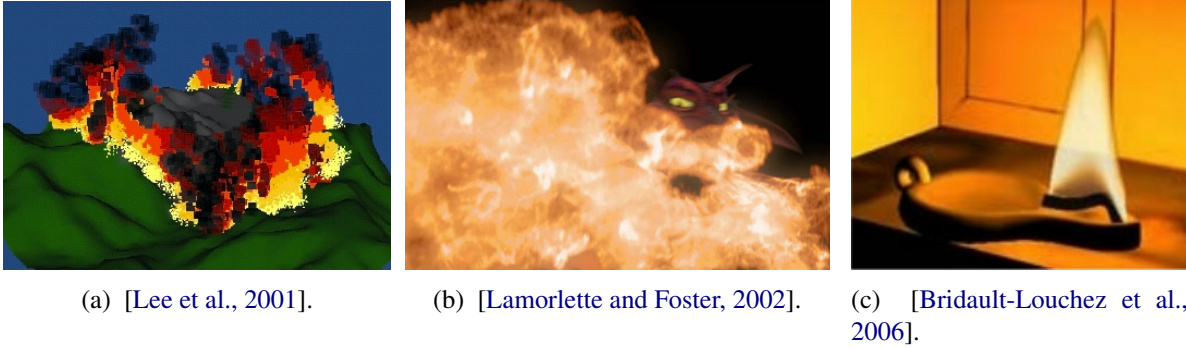


Figure 2-3: Geometry skeleton fire examples.

Data driven methods are used to simulate fire with data from real flames, as shown in Figure 2-4. The quality of the animation depends directly on the data, and by avoiding the simulation stage the computation overhead is reduced significantly. However, there are severe limitations when producing new animations and interactions with other objects. [Rushmeier et al., 1995] captured a heptane pool fire with water-cooled nitrogen-purged probes. Using two images from orographic camera views, [Hasinoff and Kutulakos, 2003] reconstructed 3D volumetric fire using tomographic techniques. [Ihrke and Magnor, 2004] extended the work by [Hasinoff and Kutulakos, 2003], the authors presented a method to reconstruct fire volumes using trilinear basis functions in scenes recorded with eight cameras. [Zhang et al., 2011] presented a method to synthesize new animation sequences based on a previously simulated data. A flow graph whose nodes connect resembling simulation states is built, similarity between nodes is measured based on flow pathline distances. New frames with smooth transitions are created by traversing the graph through different paths.

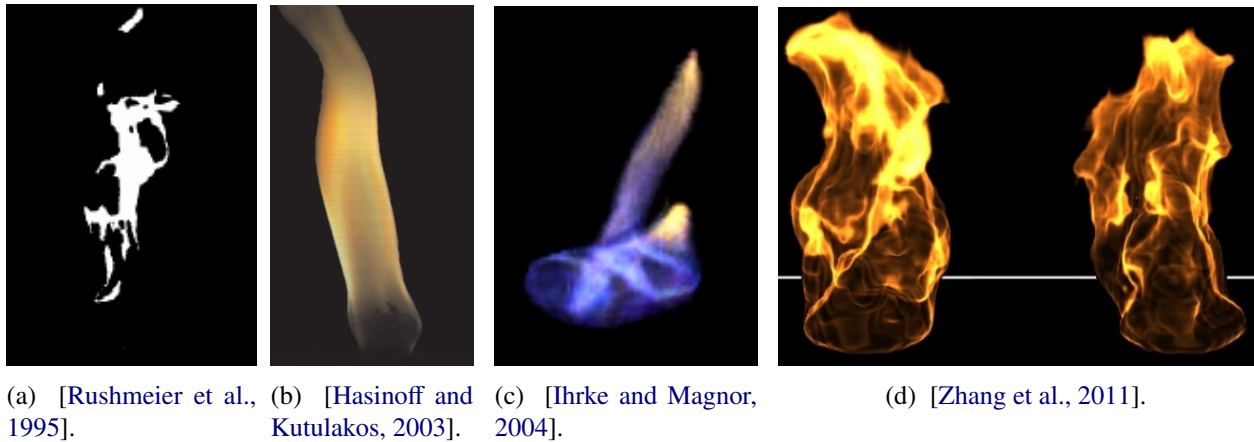


Figure 2-4: Data driven fire examples.

Physically based simulate the fire combustion processes, including flame propagation or the chemical reactions that convert fuel into gaseous products, as shown in Figure 2-5. Incompressible flow equations were used by [Stam and Fiume, 1995] to drive a fire simulation. Given initial fuel conditions, the fire spread is advected on a grid using an advection-diffusion type equation. Building on the work on a semi-Lagrangian fluid solver of [Stam, 1999], a model which includes gaseous fuel and gaseous byproducts was proposed

by [Nguyen et al., 2002]. In order to include the characteristics of the noise-based methods, [Hong et al., 2007] combined the previous model with a set of third-order equations from detonation shock dynamics presented by [Yao and Stewart, 1996]. As with the noise-based methods, this addition is visually attractive, yet it is not physically based. Capitalizing on the recent advances in GPUs parallel processing power, [Horvath and Geiger, 2009] proposed a fixed camera model. Particle properties are computed on a three-dimensional coarse grid, which are then projected into several view dependant two-dimensional slices. The authors' model is based on the assumption that fine variations, which are perpendicular to the projection plane, are not individually visible and, they do not affect significantly the overall flow.

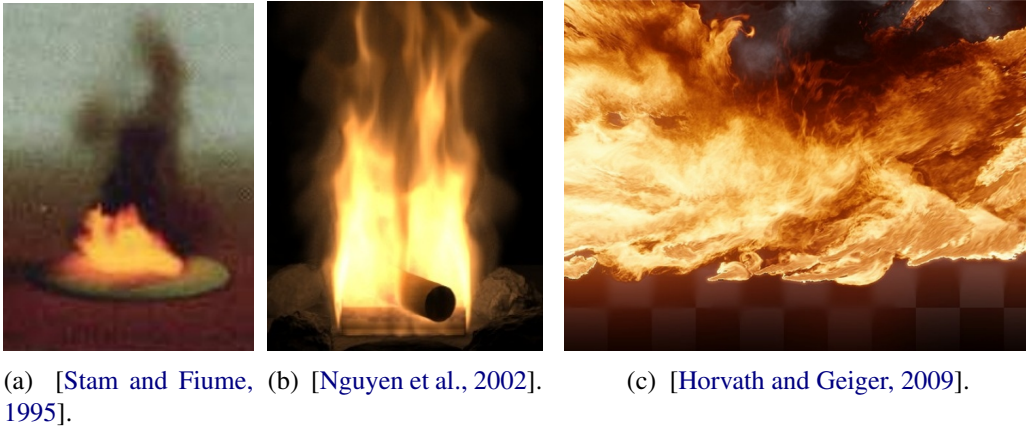


Figure 2-5: Physically-based fire examples.

Other effects directly related to fire have also been explored. [Feldman et al., 2003] presented a model to simulate suspended particles during explosions. An incompressible fluid model drives the motion of air and hot gases, and the suspended particles follow their movements. Sound is an important factor to increase the believability of a finished fire animation. [Melek and Keyser, 2005] simulated the decomposition of an object as it undergoes combustion. Heat is transferred between the different parts of the object and the air; the decomposition is modelled as a moving boundary spreading in the distance field of the object, where level set methods are used to compute the moving interfaces. [Chadwick and James, 2011] proposed a method to automatically generate plausible noise given for a given fire simulation. Low frequency sound is estimated using a physical model whose inputs are the flame front and heat release. A data driven sound synthesis approach, based on the work by [Wei and Levoy, 2000], is applied to generate the high frequency content.

2.2 Rendering

Rendering fire is more challenging than rendering other type of participating media, the main reason being that fire is a light-emitting source. The luminescence radiated by a flame is generated by black body radiation due to the high temperature of the particles during the combustion process. In order to realistically render fire, light absorption and scattering in the media, including air, has to be taken into consideration. A succinct review of the current methods used by the computer graphics community for fire rendering is presented below.

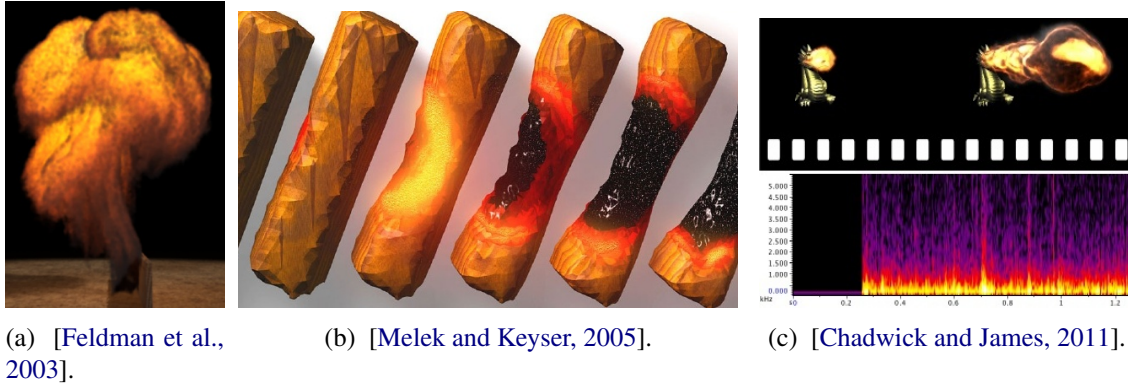


Figure 2-6: Other effects with fire.

2.2.1 Raster-Based

Raster-based techniques sacrifice quality in the interest of interactive frame rates. Some form of texture mapping is usually practised, usually only the surface of the flame is considered, and the illumination of the scene is approximated based on some parameter which can be easily computed such as a as well, e.g. the fire height.

[Reeves, 1983] applied a linear colour assignment to each particle in their simulation based on their lifetime. [Lee et al., 2001] applied the same technique to render fires on mesh surfaces, where a particle would begin with a light yellow colour, evolve to red and finish in black at the end of their lifetime. [Lamorlette and Foster, 2002] presented a technique where a based flame picture is mapped onto the two-dimensional flame profile with a base colour. For each particle an intermediate emitting value is computed, and the final colour is super-sampled profile from an approximation of the cross-sectional area of the flame, as it would appear from the camera. [Bridault-Louchez et al., 2006] used a spectrophotometer to capture photometric distributions of candles. The intensities are stored on a texture and changes in illumination over time are introduced with an attenuation factor which is computed based on the size of the flame. [Zhang et al., 2011] proposed a method where fire particles and their attributes are first projected onto a set of slicing planes, which are orthogonal to the camera direction. The planes are then blended to the screen in back-to-front order, and a one-dimensional colour texture is used as transfer function to convert flow attributes to colours and opacities.

2.2.2 Ray-Tracing-Based

Volume ray-tracing techniques offer astonishing results, however the associated computational costs are considerable. Rays are shot from the view plane and evaluated at small increments; the total radiance at the origin of the ray is computed by integrating the radiance at each step size. A further drawback for ray tracing techniques, in comparison to raster-based methods, is the lack of a standard ray-tracing pipeline.

[Rushmeier et al., 1995] presented a method to perform accurate ray casting on sparse measured data. The fire was modelled as a series of stacked cylindrical rings, where each ring has with uniform properties.

The total radiance at each point is integrated using a MonteCarlo method, summing up the measured irradiances at sample locations. [Nguyen et al., 2002] proposed a ray marching technique to solve the Radiative Transport Equation, see Chapter 3. The emitted light is computed using Planck’s formula of black body radiation, the light scattering in the media and visual adaptation to the fire spectra are modelled. [Feldman et al., 2003] also included black body radiation in their animation of fire with suspended particles, however the mapping to RGB was manually adjusted to match the images of real explosions. Direct illumination shadows were computed using deep shadow maps [Lokovic and Veach, 2000], while scattering and illumination by other objects in the scene used the technique proposed by [Jensen and Buhler, 2002]. An extension to [Nguyen et al., 2002] was presented by [Pegoraro and Parker, 2006], the authors’ model has physically-based absorption, emission and scattering properties. The spectroscopy characteristics of different fuels are achieved by modelling the electronic transitions between states in the molecules. Non-linear trajectories of light in the medium due to light refraction effects are included as well. In order to minimize the effects induced by the limitations of the RGB colour space, the visual adaptation process is presented as a post-processing effect. [Horvath and Geiger, 2009] proposed a rendering method whose main objective was user-friendliness for artists. Using the fixed camera slices described in Section 2.1, the authors’ perform a simple volume rendering to join them in a single image. Black body radiation is used for the light, the images are motion-blurred with a filter based on the velocities in the slices, and the heat distortion is added as post-processing user defined filter.

Chapter 3

Methodology

Not only explain in more detail every equation, what it means, comparison with other papers and ideally improvements or where it fails at least. Should I explain what is and how to do ray marching? What is the spectrum? How to integrate it to RGB coefficients? Add all the values for the constants here or in implementation details??

Given simulated or captured fire data, that could come from any of the methods discussed in Section 2.1, we want to generate a photo-realistic image from a camera view of the scene. From this point onwards we will assume that we have a volumetric data structure, which holds the relevant input data for the fire.

3.1 Radiative Transport Equation

The Radiative Transport Equation (RTE) [Howell and Siegel, 2002] models the variation of spectral radiance $(\omega \nabla)L(\lambda, \mathbf{x}, \omega)$ in the medium, where λ is a given wavelength in m , \mathbf{x} is the point of interest in space, and ω is a vector that points towards the viewing direction. Formally, the objective is to get a solution for the RTE for each visible point in the scene, practically, an approximate solution will be computed.

The integro-differential Radiative Transport Equation is defined as

$$\begin{aligned} (\omega \nabla)L(\lambda, \mathbf{x}, \omega) = & -\sigma_a(\lambda, \mathbf{x})L(\lambda, \mathbf{x}, \omega) + \sigma_a(\lambda, \mathbf{x})L_e(\lambda, \mathbf{x}, \omega) \\ & -\sigma_s(\lambda, \mathbf{x})L(\lambda, \mathbf{x}, \omega) + \sigma_s(\lambda, \mathbf{x})L_i(\lambda, \mathbf{x}, \omega), \end{aligned} \quad (3.1)$$

where L_i is defined by

$$L_i(\lambda, \mathbf{x}, \omega) = \int_{4\pi} L(\lambda, \mathbf{x}, \omega_i) \Phi(\lambda, \omega, \omega_i) d\omega_i, \quad (3.2)$$

where σ_a is an absorption coefficient, σ_s is a scattering coefficient, L_e is the emitted spectral radiance

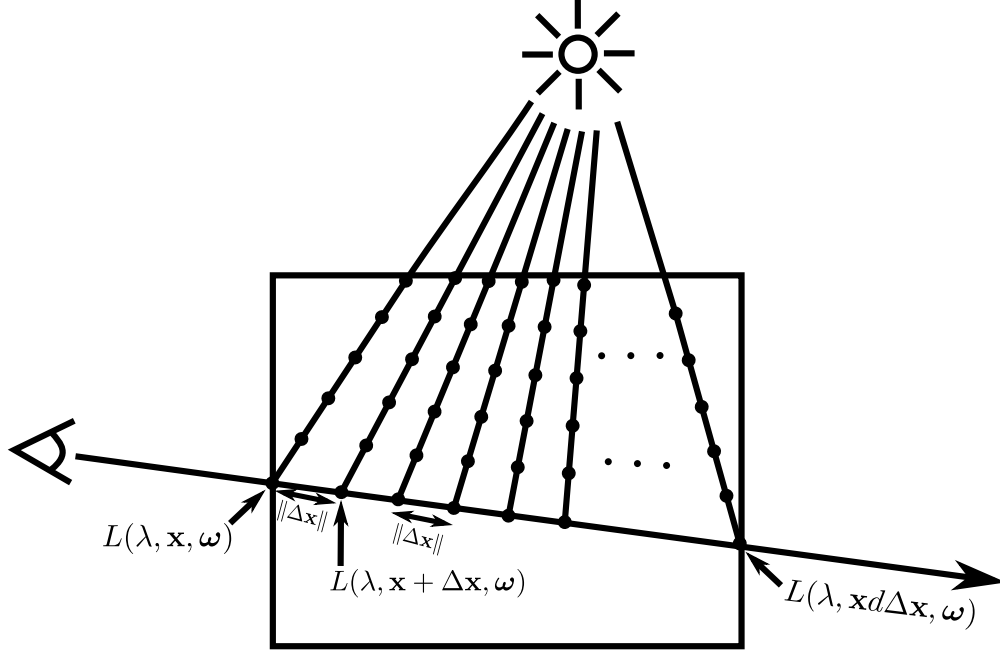


Figure 3-1: Calculating the illumination values for samples along a ray.

at the point, L_i is the in-scattering radiance, Φ is a scattering phase function and ω_i is a scattering sampling direction. In order to get an analytical solution to the aforementioned equation, the properties of the medium are assumed to be homogeneous over a small segment $\|\Delta \mathbf{x}\|$ in space,

$$L(\lambda, \mathbf{x} + \Delta \mathbf{x}, \omega) = e^{-\sigma_t(\lambda, \mathbf{x})\|\Delta \mathbf{x}\|} L(\lambda, \mathbf{x}, \omega) + (1 - e^{-\sigma_t(\lambda, \mathbf{x})\|\Delta \mathbf{x}\|}) \frac{\sigma_a(\lambda, \mathbf{x}) L_e(\lambda, \mathbf{x}, \omega) + \sigma_s(\lambda, \mathbf{x}) L_i(\lambda, \mathbf{x}, \omega)}{\sigma_t(\lambda, \mathbf{x})}, \quad (3.3)$$

where $\sigma_t = \sigma_a + \sigma_s$ is the extinction coefficient, a visual depiction on this solution is shown in Figure 3-1. Examining each individual term in Equations 3.2 and 3.3, the following factors will demand a more detailed definition

- Light **scattering** in the media $L_i(\lambda, \mathbf{x}, \omega)$ with a phase function $\Phi(\lambda, \omega, \omega_i)$
- Emitted light $L_e(\lambda, \mathbf{x}, \omega)$ by **black body radiation** effects
- Electromagnetic **absorption**, σ_a , of certain wavelengths
- Photons may follow non linear paths $\Delta \mathbf{x}$, due to spatially varying **refractive** indices
- The eyes response to the final radiance $L(\lambda, \mathbf{x}, \omega)$ is modified through **visual adaptation** processes

3.2 Scattering

Some photons travelling along arbitrary directions ω_i , might be scattered in the direction of interest ω . The amount of photons which are subjected to this process is controlled by a coefficient σ_s , the radiance carried by the photons is denoted by $L_i(\lambda, \mathbf{x}, \omega)$ and follows a distribution defined by the phase function

$$\Phi(\lambda, \omega, \omega_i) = \frac{1 - g(\lambda)^2}{4\pi(1 + g(\lambda)^2 - 2g(\lambda)\omega\omega_i)^{\frac{3}{2}}}, \quad (3.4)$$

where ω is the outgoing direction, g can be a function of the wavelength, although in most cases is chosen to be constant. The value of g must be in the range $(-1, 1)$, where $g < 0$ corresponds to backwards scattering, $g = 0$ to isotropic scattering, and $g > 0$ to forwards scattering. Monte Carlo techniques can be used to sample the light scattering directions ω_i .

3.3 Black Body Radiation

Emitted light $L_e(\lambda, \mathbf{x}, \omega)$ from an idealized physical body that absorbs all incident electromagnetic radiation can be modelled with Planck's equation for black body radiation. This equation characterizes the electromagnetic radiation $B(T, \lambda, \eta)$ emitted by a black body in thermal equilibrium at a given temperature T .

$$L_e(\lambda, \mathbf{x}, \omega) = B(T, \lambda, \eta) = \frac{2hc^2}{\lambda^5(e^{\frac{hc}{\lambda kT}} - 1)}, \quad (3.5)$$

where $h = 6.62606957 \times 10^{-34} \text{ J/s}$ is Planck's constant, $k = 1.3806488 \times 10^{-23} \text{ J/K}$ is Boltzmann constant, η is the refraction index of the medium and $c = c_0/\eta$ is the speed of light in the current medium, where $c_0 = 299792458 \text{ m/s}$ is the speed of light in a vacuum. The refraction index η varies across the medium, the procedure to compute it will be described in Section 3.5.

3.4 Absorption coefficients

As fire is not an idealized black body radiator, certain wavelengths are not present in the emitted radiance. This behaviour can be modelled using absorption coefficients for each chemical that would mask the undesired wavelengths, such that

$$\sigma_a L_e(\lambda, \mathbf{x}, \omega) = \sum_{i=0}^N \sigma_{ai} B_i(T, \lambda, \eta), \quad (3.6)$$

where N is the number of different chemical species in the mix, σ_{ai} and $B_i(T, \lambda, \eta)$ are the absorption

coefficient and light emission of the i^{th} chemical. In the interest of simplicity, the absorption and radiation effects for **soot** will be treated separately from those caused by **other chemical species**.

3.4.1 Soot Absorption

The spectral absorption coefficient of soot is defined as

$$\sigma_a(\lambda, \mathbf{x}) = \frac{48N(\mathbf{x})\pi R^3 nm}{\lambda^{\alpha(\lambda)}((n^2 - m^2 + 2)^2 + 4n^2 m^2)}, \quad (3.7)$$

where $N(\mathbf{x})$ is the number density, density per unit volume, R is the radius of a soot particle, n , m and $\alpha(\lambda) = 1.39$ are optical constants for different types of soot. In Table 3.1, values for the optical constants n and m are provided for several materials, the data was obtained from [Dalzell and Sarofim, 1969]. The radius of soot particles was determined in the range $R \in \{50 \text{ \AA} \dots 800 \text{ \AA}\}$ by [Dalzell and Sarofim, 1969]. Since our data is defined with soot densities, we have chosen the radius to be the mean value $R = 425 \times 10^{-10}$ metres.

Table 3.1: Absorption constants for propane and acetylene, wavelengths are in nanometres.

		Wavelengths			
		435.8	450	550	650
Propane	n	1.57	1.56	1.57	1.56
	nm	0.46	0.5	0.53	0.52
Acetylene	n	1.56	1.56	1.56	1.57
	nm	0.46	0.48	0.46	0.44

3.4.2 Absorption From Other Chemical Species

The absorption coefficients associated with a given spectral frequency can be computed as

$$\sigma_a(\lambda, \mathbf{x}) = \frac{\phi(\lambda)N_2 A_{21} \lambda^4 (e^{\frac{hc}{\lambda kT}} - 1)}{8\pi c} \quad (3.8)$$

where $\phi(\lambda)$ is the normalized spectral line, N_2 is the number density of elements, A_{21} is an Einstein coefficient measuring the transition probabilities of an spontaneous emission.

3.5 Refraction

Assuming a negligible reflection index, the refraction angles for the rays can be easily computed using Snell's law

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{\eta_2}{\eta_1}, \quad (3.9)$$

where θ_1 is the incident angle, θ_2 is the refracted angle, η_1 is the index of refraction of the media which the ray is coming from and η_2 is the index of refraction of the media which the ray is going to. The values for all the constants in this section are defined in Table 3.2.

[Ciddor, 1996] proposed a method to compute the refractive indices of air

$$\eta_{air} = 1 + \frac{\rho_a \eta_{axs}}{\rho_{axs}} + \frac{\rho_w \eta_{ws}}{\rho_{ws}}, \quad (3.10)$$

where ρ_{axs} is the density of dry air, ρ_{ws} is the density of pure water vapour, ρ_a and ρ_w are the equivalent quantities for dry air, η_{axs} is the refractive index for CO_2 and η_{ws} is the refractive index for water vapour. The refractive index for CO_2 is estimated as

$$\eta_{axs} = \eta_{as} \left(1 + 0.534 \times 10^{-6} (x_c - 450) \right), \quad (3.11)$$

$$\eta_{as} = 10^{-8} \left(\frac{k_1}{k_0 - \lambda} + \frac{k_3}{k_2 - \lambda} \right), \quad (3.12)$$

where $x_c \in \{300 \dots 450\}$ is the proportion in ppm of CO_2 , k_0 , k_1 , k_2 and k_3 are constants, and λ is the wave number (reciprocal of the vacuum wavelength). The refractive index for water vapour is computed as

$$\eta_{ws} = 1.022 \times 10^{-8} (w_0 + w_1 \lambda + w_2 \lambda^2 + w_3 \lambda^3), \quad (3.13)$$

where w_0 , w_1 , w_2 and w_3 are constants.

The ρ densities are computed as follows

$$\rho = \frac{pm_a}{zrT} \left(1 - x_w \left(1 - \frac{m_w}{m_a} \right) \right), \quad (3.14)$$

where p is the pressure in pascals, r and m_w are constants,

$$m_a = 0.0289635 + 12.011 \times 10^{-8} (x_c - 400), \quad (3.15)$$

$$z = 1 - \frac{p}{T} \left(a_0 + a_1 t + a_2 t^2 + (b_0 + b_1 t) x_w + (c_0 + c_1 t) x_w^2 \right) + \left(\frac{p}{T} \right)^2 (d + e x_w^2), \quad (3.16)$$

where a_0 , a_1 , a_2 , b_0 , b_1 , c_0 , c_1 , d and e are constants.

$$T = t + 273.15, \quad (3.17)$$

$$x_w = \frac{fhv}{p}, \quad (3.18)$$

$$f = \alpha + \beta p + \gamma t^2, \quad (3.19)$$

where α , β and γ , and $h \in \{0 \dots 1\}$ is the air relative fractional humidity, 0 for dry air and given by the user for moist air.

The pressure p can also be computed using the ideal gas law

$$pV = \frac{NRT}{V} = nT, \quad (3.20)$$

where N is the number of molecules, V is the total volume of the gas, T is the temperature of the gas, n is the number density, $R = kN_a$, where k is the Boltzmann constant and N_a is Avogadro constant.

Depending on the situation, there are two acceptable methods to compute the vapour pressure v , both are described below.

3.5.1 Davis' Saturation Vapour Pressure

In [Ciddor, 1996] paper, the author used the method proposed by [Davis, 1992] to compute the saturation of vapour pressure

$$v = e^{AT^2+BT+C+D/T}, \quad (3.21)$$

where A , B , C and D are constants. This equation is relatively easy to compute, however incorrect results will be obtained for temperatures below 0 °C. As we are concerned with fire rendering, which entails high temperatures, this method was chosen for its simplicity.

3.5.2 Huang's Saturation Vapour Pressure

The International Association for the Properties of Water and Steam (IAPWS) adopted an alternative technique to [Davis, 1992], which was proposed by [Huang, 1998]. The more recent method will address the drawbacks of the previous one, giving reasonable results for temperatures below 0 °C, nevertheless, the generality comes with increased complexity in the equations.

$$v = 10^6 \left(\frac{2C}{X} \right)^4, \quad (3.22)$$

$$X = -B + (B^2 - 4AC)^{\frac{1}{2}}, \quad (3.23)$$

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} 1 & K_1 & K_2 \\ K_3 & K_4 & K_5 \\ K_6 & K_7 & K_8 \end{bmatrix} \begin{bmatrix} \Omega^2 \\ \Omega \\ 1 \end{bmatrix}, \quad (3.24)$$

$$\Omega = T + \frac{K_9}{T - K_{10}}, \quad (3.25)$$

where $K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8, K_9$, and K_{10} are constants.

3.5.3 Ciddor's Method Summary

The inputs of Ciddor's technique are a wavelength λ_0 , a temperature t , a pressure p , a relative humidity value h and a CO_2 concentration x_c . The values for all the constants needed for the method are defined in Table 3.2. A refraction index is computed following this method for each voxel in the volumetric data structure. The steps to compute the index of refraction are listed below

1. Precompute $z_a = 0.9995922115$, from Equation 3.16 with $t = 20$ °C, $p = 101.325$ Pa and $x_w = 0$.
2. Find $\lambda = 1/\lambda_0^2$, and T with Equation 3.17.
3. Find v with the method described in Section 3.5.1 or in Section 3.5.2.
4. Find x_w using Equations 3.19 and 3.18.
5. Find η_{as} using Equation 3.12.
6. Find η_{ws} using Equation 3.13.
7. Find m_a using Equation 3.15.
8. Find η_{axs} using Equation 3.11.
9. Find z_m using Equation 3.16.
10. Find $\rho_{axs} = (p_0 m_a)/(z_a r T_0)$, where $p_0 = 101325$, and $T_0 = 288.15$, using Equation 3.14.
11. Find $\rho_w = (x_w p m_w)/(z_m r T)$ using Equation 3.14.
12. Find $\rho_a = ((1 - x_w) p m_a)/(z_m r T)$ using Equation 3.14.
13. Find the index of refraction η_{air} using Equation 3.10.

Table 3.2: Values for the constants in the equations used to compute the index of refraction.

Name	Value	Name	Value
ρ_{ws}	0.00985938	a_0	$1.58123 \times 10^{-6} KPa^{-1}$
x_c	450 ppm	a_1	$-2.9331 \times 10^{-8} Pa^{-1}$
k_0	$238.0185 \mu m^{-2}$	a_2	$1.1043 \times 10^{-10} Pa^{-1}$
k_1	$5792105 \mu m^{-2}$	b_0	$5.707 \times 10^{-6} KPa^{-1}$
k_2	$57.362 \mu m^{-2}$	b_1	$-2.051 \times 10^{-8} Pa^{-1}$
k_3	$167917 \mu m^{-2}$	c_0	$1.9898 \times 10^{-4} KPa^{-1}$
w_0	$295.235 \mu m^{-2}$	c_1	$-2.376 \times 10^{-6} KPa^{-1}$
w_1	$2.6422 \mu m^{-2}$	d	$1.83 \times 10^{-11} K^2 Pa^{-2}$
w_2	$-0.03238 \mu m^{-2}$	e	$-0.765 \times 10^{-8} K^2 Pa^{-2}$
w_3	$0.004028 \mu m^{-2}$	α	1.00062
p	1 atm = 101325 Pa	β	$3.14 \times 10^{-8} Pa^{-1}$
r	$8.31451 Jmol^{-1} K^{-1}$	γ	$5.6 \times 10^{-7} ^\circ C^{-2}$
m_w	0.018015 kg/mol	k	$1.3806488 \times 10^{-23} J/K$
N_a	$6.02214129 \times 10^{23} mol^{-1}$	K_4	$1.20208247025 \times 10^4$
A	$1.2378847 \times 10^{-5} K^{-2}$	K_5	$-3.23255503223 \times 10^6$
B	$-1.9121316 \times 10^{-2} K^{-1}$	K_6	1.49151086135×10
C	33.93711047	K_7	$-4.82326573616 \times 10^3$
D	$-6.3431645 \times 10^3 K$	K_8	$4.05113405421 \times 10^5$
K_1	$1.16705214528 \times 10^3$	K_9	$-2.38555575678 \times 10^{-1}$
K_2	$-7.24213167032 \times 10^5$	K_{10}	$6.50175348448 \times 10^2$
K_3	-1.70738469401×10		

3.6 Visual Adaptation

The human eye presents a non-linear response to incident radiance L , [Naka and Rushton, 1966] proposed a simple equation to model this phenomena

$$R(L, \tau) = \frac{L}{L + \tau}, \quad (3.26)$$

where τ is a non-linear adaptation state. This state is determined by the visual system to maximize the perception of features for a given scene.

TALK SOMEWHERE ABOUT THE OTHER VISUAL ADAPTATION MODELS PRESENTED IN FAIRCHILD:2005

Chapter 4

Implementation details

4.1 Application Overview

Explain how to use the shaders and the commands, add images of the scenes in Maya

The model chosen for our data is that of a three-dimensional voxel dataset, a cube in space is be uniformly divided and we have a data value for each voxel. This data is either a soot density estimate for the *fire volume shader* or a temperature estimate for the *fire light shader*. Two file formats are supported, dense float voxel data is ASCII or sparse binary RGBA values, as shown in Figure 4-1. In the ASCII format, the file first line are three integers, separated by spaces, declaring the width, height and depth of the data, and *whd* lines with a single floating point value. The binary sparse format starts with a 4 byte integer declaring how many data points are in the file, each voxel has its coordinates x, y, z as 4 bytes integers and a rgb alpha colour, where each channel is an 8 bytes floating point double value.

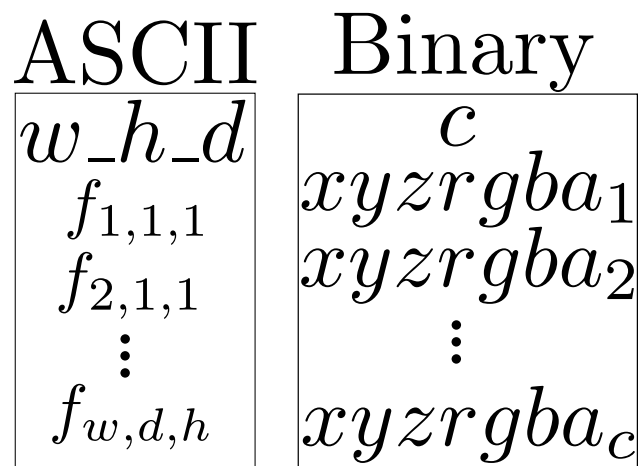


Figure 4-1: Suported file formats.

4.2 Mental Ray® Shading Approach

How mental ray shoots rays around and what is its paradigm.

Mental Ray® approach to solving the rendering equation is based on path tracing, as shown in Figure 4-2, for each pixel in the camera view, an eye ray will be shot in the scene. On an intersection with an object in the scene, its material shader will be called, this shader will shoot a light ray for each light in the scene, which in effects calls the light shader of the given light. In order to compute the irradiance at the intersection point, the light shader will probably trace a shadow ray from the light to the intersection point. Eventually, the material shader will compute the final colour with the information received from the light shader.

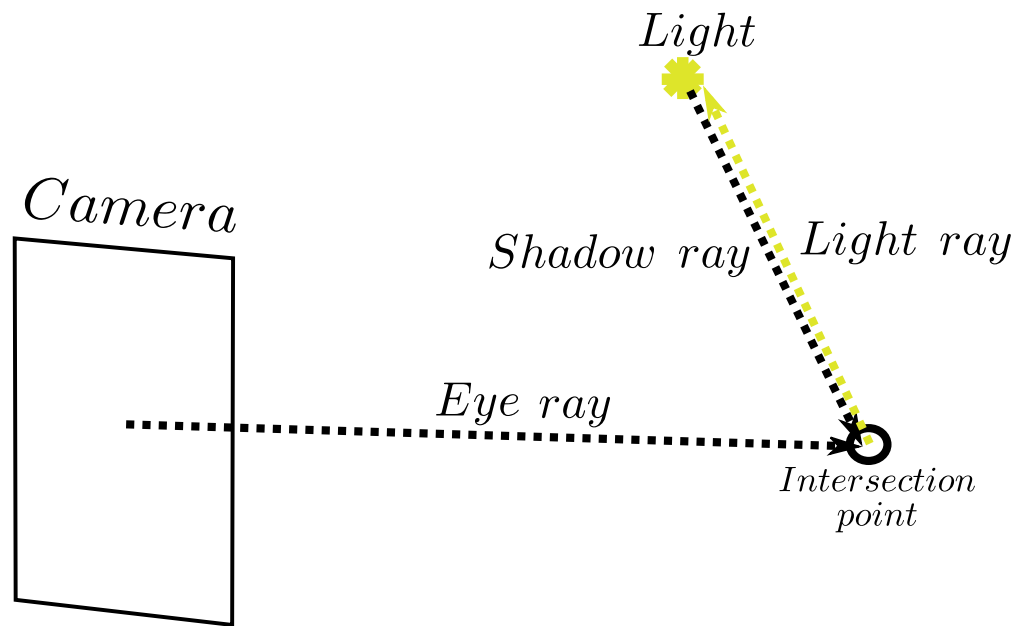


Figure 4-2: Mental Ray® simple ray casting example.

Under this assumptions, the steps required to solve the RTE, Equation 3.3, are:

1. Shoot a ray from the eye into the scene
2. If the ray intersects the fire volume, perform ray marching in the volume
 - (a) Compute direct illumination at the current point
 - i. Get initial radiance using black body radiation
 - ii. Attenuate with absorption coefficient
 - (b) Compute indirect illumination at the current point
 - i. Use phase function to get the scattered light distribution
 - (c) Advance to next point in ray marching
 - i. Change ray direction using refraction index

3. Compute eye visual adaptation to fire colour

The previous operations are done

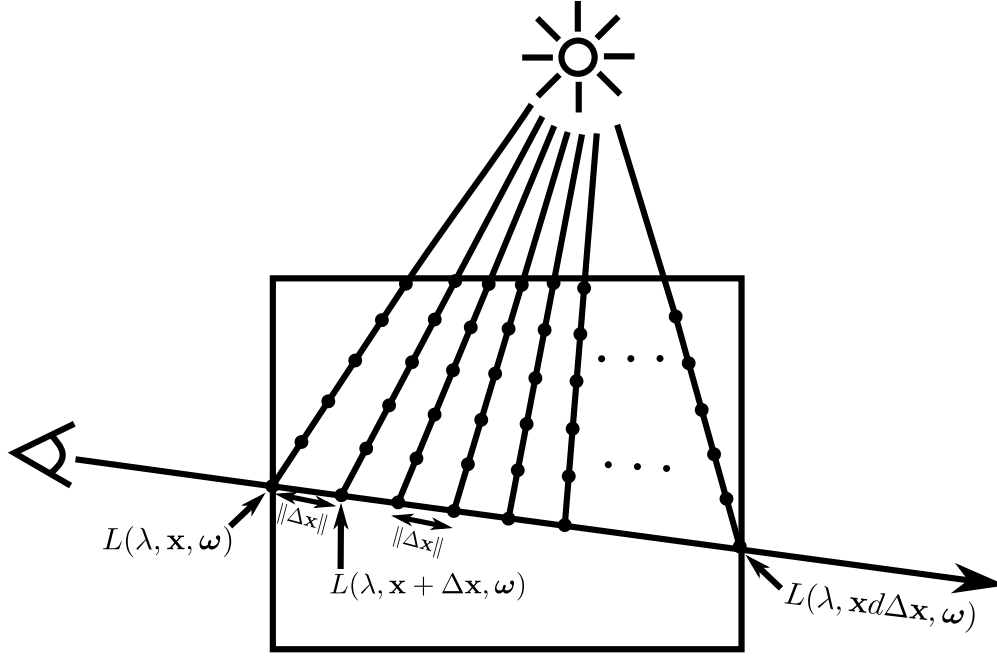


Figure 4-3: Calculating the illumination values for samples along a ray.

Equation 3.3 provides a radiance value for the next march increment, however we want to compute the value at the ray intersection with the volume, i.e. at the ray origin. So we need to rewrite the equation as

$$L(\lambda, \mathbf{x}, \omega) = e^{\sigma_t(\lambda, \mathbf{x}) \|\Delta \mathbf{x}\|} L(\lambda, \mathbf{x} + \Delta \mathbf{x}, \omega) + (1 - e^{\sigma_t(\lambda, \mathbf{x}) \|\Delta \mathbf{x}\|}) \frac{\sigma_a(\lambda, \mathbf{x}) L_e(\lambda, \mathbf{x}, \omega) + \sigma_s(\lambda, \mathbf{x}) L_i(\lambda, \mathbf{x}, \omega)}{\sigma_t(\lambda, \mathbf{x})}. \quad (4.1)$$

In fire phenomena the effect the scattered light in the final image is practically imperceptible [Pegoraro and Parker, 2006]. Exploiting the previous prior knowledge, the scattering contributions can be safely ignored by setting the coefficient σ_s equal to zero, which simplifies the previous equation to

$$L(\lambda, \mathbf{x}, \omega) = e^{\sigma_a(\lambda, \mathbf{x}) \|\Delta \mathbf{x}\|} L(\lambda, \mathbf{x} + \Delta \mathbf{x}, \omega) + (1 - e^{\sigma_a(\lambda, \mathbf{x}) \|\Delta \mathbf{x}\|}) L_e(\lambda, \mathbf{x}, \omega). \quad (4.2)$$

4.3 Shaders Internals

Talk about things like which parts are written in parallel code, instance support, shader internal memory, sparse data, how the software escalates, memory consumption, Maya integration, spectrum to RGB inte-

gration, maybe more details about units for black body radiation and everything that was not explained before

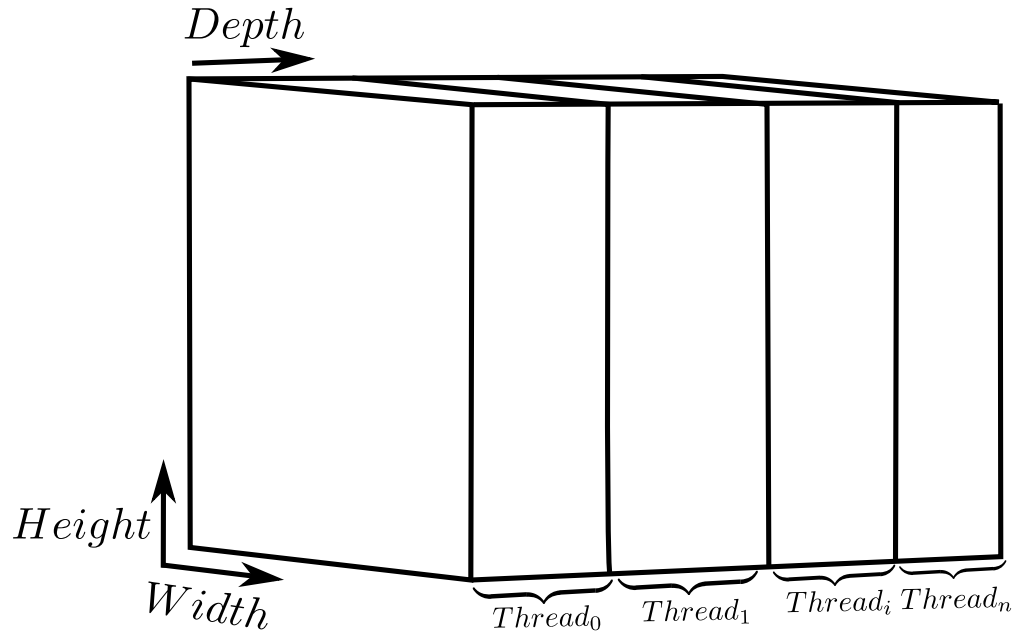
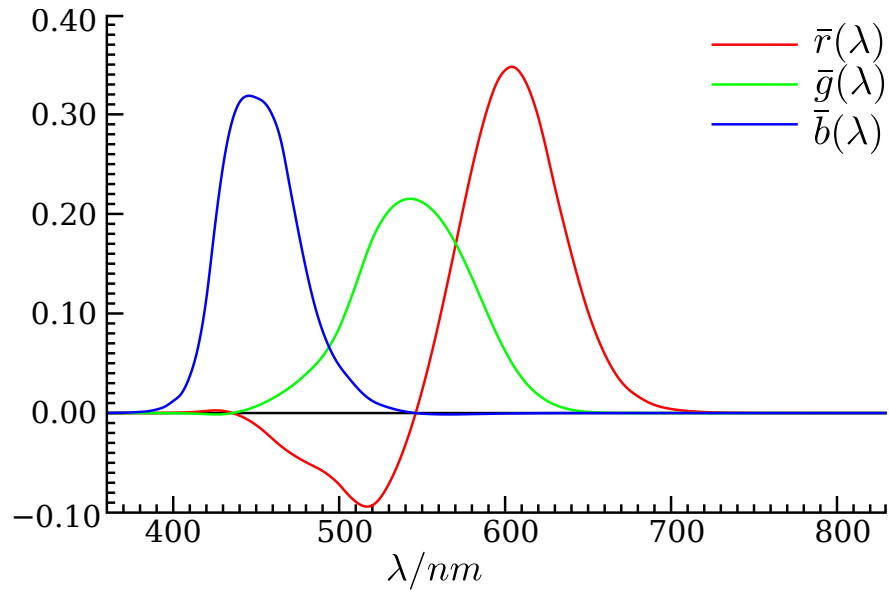


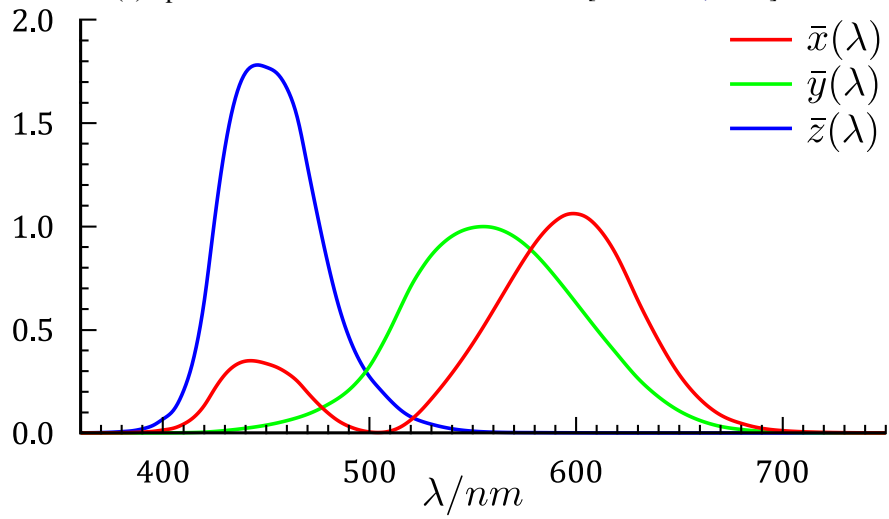
Figure 4-4: Voxel Dataset division of coefficients computation in threads.

In order to speed up the render time, all spectrum related computations are integrated to RGB coefficients. Our spectrum class has a number of samples that can be set a compile time, in general it is set to 30 samples. However, in certain cases a lower number is used, e.g. when computing the soot absorption coefficients, since we have values for the constants for only four wavelengths, using more than four samples in the spectrum would not improve the results. RGB coefficients are the standard colour representation used in computer screens, however not every colour visible by the human eye can be reproduced in the RGB space. A further disadvantage, is that for certain colours negative values for the $\bar{r}(\lambda)$ coefficient are needed, as shown in Figure 4-5a. The XYZ colour space is built using three sets of imaginary primaries, which have a series of favourable properties. Any colour which can be seen by the human eye can be represented with some $x(\lambda)$, $y(\lambda)$, $z(\lambda)$ coefficients, the Y channel is equal to the photopic luminous efficiency function $V(\lambda)$, which models the variation of perceived brightness, and as shown in Figure 4-5b, the coefficients are always positive.

Given a Spectral Power Distribution (SPD) $S(\lambda)$, the x , y , z coefficients for $S(\lambda)$ in the XYZ space are defined with respect to spectral matching curves $X(\lambda)$, $Y(\lambda)$ and $Z(\lambda)$, such that



(a) Spectral tristimulus values for CIE RGB [CIE.RGB, 2015].



(b) Spectral tristimulus values for CIE XYZ [CIE.XYZ, 2015].

Figure 4-5: Colour matching curves for RGB and XYZ spaces.

$$\begin{aligned}
x &= \frac{1}{\int Y(\lambda)d\lambda} \int_{\lambda} S(\lambda)X(\lambda)d\lambda, \\
y &= \frac{1}{\int Y(\lambda)d\lambda} \int_{\lambda} S(\lambda)Y(\lambda)d\lambda, \\
z &= \frac{1}{\int Y(\lambda)d\lambda} \int_{\lambda} S(\lambda)Z(\lambda)d\lambda.
\end{aligned} \tag{4.3}$$

Note that the term $1/\int Y(\lambda)d\lambda$ is added in each equation to act as a normalization factor for the colour brightness. As we are concerned with sampled values on a discrete domain, the integration for XYZ coefficients is approximated by a Riemann sum

$$\begin{aligned}
x &\approx \frac{1}{\int Y(\lambda)d\lambda} \sum_i X_i c_i, \\
y &\approx \frac{1}{\int Y(\lambda)d\lambda} \sum_i Y_i c_i, \\
z &\approx \frac{1}{\int Y(\lambda)d\lambda} \sum_i Z_i c_i,
\end{aligned} \tag{4.4}$$

where c_i is the i^{th} coefficient in the SPD, X_i is area for the the corresponding sample range the $X(\lambda)$ curve and equivalently for Y_i and Z_i . Note that $1/(\int Y(\lambda)d\lambda)$, X_i , Y_i and Z_i are constants and they can be precomputed for efficiency. For the Riemann sum, the area under the samples is approximated using piecewise linear interpolation. The conversion from XYZ to RGB space is performed using

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} \int R(\lambda)X(\lambda)d\lambda & \int R(\lambda)Y(\lambda)d\lambda & \int R(\lambda)Z(\lambda)d\lambda \\ \int G(\lambda)X(\lambda)d\lambda & \int G(\lambda)Y(\lambda)d\lambda & \int G(\lambda)Z(\lambda)d\lambda \\ \int B(\lambda)X(\lambda)d\lambda & \int B(\lambda)Y(\lambda)d\lambda & \int B(\lambda)Z(\lambda)d\lambda \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{4.5}$$

where $R(\lambda)$, $G(\lambda)$, $B(\lambda)$ are the spectral curves for the red, green and blue colours respectively. Note that all the factors in the transformation matrix are constants and they can be precomputed for efficiency.

4.3.1 Fire Volume Shader

In order to achieve smoother rendering results, a trilinear interpolation is computed in each step in the ray-marching algorithm. Computing the interpolation smoothed the shape of the flames and reduced the effects of outliers in the input data. Tricubic interpolation was also considered, however it was discarded due to the significant increase in computational overhead.

The implementation of the eye visual adaptation to the colours in fire, described in Equation 3.26 is performed as described in [Nguyen et al., 2002], which uses Von Kries model [Fairchild, 2005]

$$\begin{aligned}
\mathbf{x}_a &= \mathbf{M}^{-1} \mathbf{T}_w \mathbf{M} \mathbf{x}_i, \\
\mathbf{l}_{max} &= \mathbf{M} \mathbf{x}_{max}, \\
\mathbf{T}_w &= \begin{bmatrix} 1/l_{max} & 0 & 0 \\ 0 & 1/m_{max} & 0 \\ 0 & 0 & 1/s_{max} \end{bmatrix}, \\
\mathbf{M} &= \begin{bmatrix} 0.4002 & 0.7076 & -0.0808 \\ -0.2263 & 1.1653 & 0.0457 \\ 0 & 0 & 0.9182 \end{bmatrix},
\end{aligned} \tag{4.6}$$

where $\mathbf{x}_a = [x_a, y_a, z_a]^T$ is a column vector with the XYZ adapted coefficients, \mathbf{T}_w is the Von Kries transformation, x_{max} are the coefficients for the maximum temperature present in the fire, $\mathbf{l}_{max} = [l_{max}, m_{max}, s_{max}]^T$ are the LMS coefficients of x_{max} , \mathbf{M} is a XYZ to LMS transformation matrix and, $\mathbf{x}_i = [x_i, y_i, z_i]^T$ is a column vector with the XYZ input coefficients. The XYZ to LMS \mathbf{M} matrix is a Hunt-Pointer-Estevéz transformation, proposed by [Hunt and Pointer, 1985], which has been normalized to the CIE Standard Illuminant D65. Since M is a small 3×3 matrix, the inversion can be performed with any of the standard methods, in our case we used Matlab[®] default's algorithm and the values were hard-coded in the shader. Solving the aforementioned equation in the XYZ space, instead of the RGB, will improve the quality of the final rendered image.

4.3.2 Fire Light Shader

4.3.3 Voxel Dataset Shader

Chapter 5

Results

Images everywhere and proper analysis of what the user is seeing, what is failing and why. Include default Maya fire effect, pictures of fire from the papers mentioned in the previous work section and a historical overview on the improvements in the shaders.

Chapter 6

Conclusions and Future work

Bibliography

- [Bridault-Louchez et al., 2006] Bridault-Louchez, F., Leblond, M., and Rousselle, F. (2006). Enhanced illumination of reconstructed dynamic environments using a real-time flame model. In *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '06, pages 31–40, New York, NY, USA. ACM.
- [Chadwick and James, 2011] Chadwick, J. N. and James, D. L. (2011). Animating fire with sound. *ACM Trans. Graph.*, 30(4):84:1–84:8.
- [Ciddor, 1996] Ciddor, P. E. (1996). Refractive index of air: new equations for the visible and near infrared. *Appl. Opt.*, 35(9):1566–1573.
- [CIE.RGB, 2015] CIE.RGB (2015). CIE RGB Spectral Values. https://commons.wikimedia.org/wiki/File:CIE1931_RGBCMF.svg#/media/File:CIE1931_RGBCMF.svg. Accessed: 21-07-2015.
- [CIE.XYZ, 2015] CIE.XYZ (2015). CIE XYZ Spectral Values. https://commons.wikimedia.org/wiki/File:CIE_1931_XYZ_Color_Matching_Functions.svg#/media/File:CIE_1931_XYZ_Color_Matching_Functions.svg. Accessed: 21-07-2015.
- [Dalzell and Sarofim, 1969] Dalzell, W. H. and Sarofim, A. F. (1969). Optical constants of soot and their application to heat-flux calculations. *ASME Journal of Heat Transfer*, 91(1):100–104.
- [Davis, 1992] Davis, R. S. (1992). Equation for the determination of the density of moist air (1981/91). *Metrologia*, 29(1):67.
- [Fairchild, 2005] Fairchild, M. D. (2005). *Color appearance models*. John Wiley & Sons.
- [Feldman et al., 2003] Feldman, B. E., O’Brien, J. F., and Arikan, O. (2003). Animating suspended particle explosions. *ACM Trans. Graph.*, 22(3):708–715.
- [FireImg1, 2015] FireImg1 (2015). Fire image 1. <https://en.wikipedia.org/wiki/File:Fire.JPG>. Accessed: 21-07-2015.
- [FireImg2, 2015] FireImg2 (2015). Fire image 2. https://en.wikipedia.org/wiki/File:Large_bonfire.jpg. Accessed: 21-07-2015.
- [Hasinoff and Kutulakos, 2003] Hasinoff, S. and Kutulakos, K. (2003). Photo-consistent 3d fire by flame-sheet decomposition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1184–1191 vol.2.

- [Hong et al., 2007] Hong, J.-M., Shinar, T., and Fedkiw, R. (2007). Wrinkled flames and cellular patterns. *ACM Trans. Graph.*, 26(3).
- [Horvath and Geiger, 2009] Horvath, C. and Geiger, W. (2009). Directable, high-resolution simulation of fire on the gpu. *ACM Trans. Graph.*, 28(3):41:1–41:8.
- [Howell and Siegel, 2002] Howell, J. and Siegel, R. (2002). *Thermal radiation heat transfer*. CRC press.
- [Huang, 1998] Huang, P. H. (1998). New equations for water vapor pressure in the temperature range-100 c to 100 c for use with the 1997 nist/asme steam tables. In *Proc. of the 3rd International Symposium on Humidity and Moisture (Teddington: NPL)*, volume 1, pages 69–76.
- [Huang et al., 2014] Huang, Z., Gong, G., and Han, L. (2014). Physically-based modeling, simulation and rendering of fire for computer animation. *Multimedia Tools and Applications*, 71(3):1283–1309.
- [Hunt and Pointer, 1985] Hunt, R. W. G. and Pointer, M. R. (1985). A colour-appearance transform for the cie 1931 standard colorimetric observer. *Color Research & Application*, 10(3):165–179.
- [Ihrke and Magnor, 2004] Ihrke, I. and Magnor, M. (2004). Image-based tomographic reconstruction of flames. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '04, pages 365–373, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.
- [Jensen and Buhler, 2002] Jensen, H. W. and Buhler, J. (2002). A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.*, 21(3):576–581.
- [Lamorlette and Foster, 2002] Lamorlette, A. and Foster, N. (2002). Structural modeling of flames for a production environment. *ACM Trans. Graph.*, 21(3):729–735.
- [Lee et al., 2001] Lee, H., Kim, L., Meyer, M., and Desbrun, M. (2001). Meshes on fire. In Magnenat-Thalmann, N. and Thalmann, D., editors, *Computer Animation and Simulation 2001*, Eurographics, pages 75–84. Springer Vienna.
- [Lokovic and Veach, 2000] Lokovic, T. and Veach, E. (2000). Deep shadow maps. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 385–392, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [Melek and Keyser, 2005] Melek, Z. and Keyser, J. (2005). Multi-representation interaction for physically based modeling. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, SPM '05, pages 187–196, New York, NY, USA. ACM.
- [Naka and Rushton, 1966] Naka, K. I. and Rushton, W. A. H. (1966). S-potentials from luminosity units in the retina of fish (cyprinidae). *The Journal of Physiology*, 185(3):587–599.
- [Nguyen et al., 2002] Nguyen, D. Q., Fedkiw, R., and Jensen, H. W. (2002). Physically based modeling and animation of fire. *ACM Trans. Graph.*, 21(3):721–728.
- [Pegoraro and Parker, 2006] Pegoraro, V. and Parker, S. G. (2006). Physically-based realistic fire rendering. *Natural Phenomena*, pages 51–59.
- [Perlin, 1985] Perlin, K. (1985). An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296.

- [Perlin and Hoffert, 1989] Perlin, K. and Hoffert, E. M. (1989). Hypertexture. *SIGGRAPH Comput. Graph.*, 23(3):253–262.
- [Perry and Picard, 1994] Perry, C. H. and Picard, R. W. (1994). Synthesizing flames and their spreading. In *Proceedings of 5th Eurographics workshop on animation and simulation*, pages 105–117.
- [Reeves, 1983] Reeves, W. T. (1983). Particle Systems - a Technique for Modeling a Class of Fuzzy Objects. *ACM Trans. Graph.*, 2(2):91–108.
- [Rushmeier et al., 1995] Rushmeier, H., Hamins, A., and Choi, M. Y. (1995). Volume rendering of pool fire data. *Computer Graphics and Applications, IEEE*, 15(4):62–67.
- [Stam, 1999] Stam, J. (1999). Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 121–128, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [Stam and Fiume, 1995] Stam, J. and Fiume, E. (1995). Depicting fire and other gaseous phenomena using diffusion processes. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 129–136, New York, NY, USA. ACM.
- [Wei and Levoy, 2000] Wei, L.-Y. and Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 479–488, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [Yao and Stewart, 1996] Yao, J. and Stewart, D. S. (1996). On the dynamics of multi-dimensional detonation. *Journal of Fluid Mechanics*, 309:225–275.
- [Zhang et al., 2011] Zhang, Y., Correa, C. D., and Ma, K.-L. (2011). Graph-based fire synthesis. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '11, pages 187–194, New York, NY, USA. ACM.